המתרגל: עמרי אבנד omriabnd@gmail.com

תרגילים - מי' מטלון. עד הלילה לאחרונה מהם 2-.
אם יש בעיות למייל - ניצור קשר ופתרו.

מבט כשאנחנו מדברים על אלגו', אנחנו מדברים על סדרה של פעולות שמורות לאלגוריתם ג כי יש דברים שאפשר לחשב שלא אפשר לחשב אותם מה שחשוב הוא שהתוצאה היתה סופי. יהיה לה חשיבה להבדיל מה לה לאלגוריתם מסוים. אפשר גם לעלות כל הבעיה חסום ויש לנו מספר הקלטים שאלגוריתם מקבל. ואנחנו מעוניינים על אלגו, אנחנו מעוניינים על סדרה של פעולות שמורות

ואנחנו אוהבים את הכוח שלה יחד עם יכולת סדר הגודל של הקלט שנ"ן כלומר יש אשכים איך הפונקציה מתנהגת לקלטים קטנים, אלא מעניין אותנו איך היא מתנהגת כשהקלט גדל.

נגדיר כאוסף נוסחית: תהי $g: \mathbb{N} \to \mathbb{N}$

$$\Theta(g) = \left\{ f: \mathbb{N} \to \mathbb{N} \;:\; \begin{array}{l} \exists c_1, c_2 > 0,\ n_0 \in \mathbb{N} : \\ \forall n \geq n_0 \quad 0 < c_1 g(n) \leq f(n) \leq c_2 g(n) \end{array} \right\}$$

אלה פונקציות ש"גדלות באותו סדר גודל כמו ש g.

לכל $k>0$ הפונקציה n^k שייכת ל- $O(2^n)$.

דוגמה: לכל $\varepsilon>0$ מתקיים $\log n \in O(n^\varepsilon)$.

הוכיחו את הטענות הבאות, אשר מראות כי הסימון O אכן מייצג חסם אסימפטוטי:

הוכחה: (יש שיעדיפו לומר אלגברי, אך הכוונה היא לאותו הדבר) נראה כי $f \in O(g)$ אם ורק אם $\frac{f(n)}{g(n)}$ חסומה מלעיל, כלומר קיים $c>0$ כך ש- $\frac{f(n)}{g(n)} \le c$ החל ממקום מסוים.

או במילים אחרות $\frac{f(n)}{g(n)} \le c$ עבור $n \ge n_0$ אם ורק אם $\frac{f(n)}{g(n)} \to \infty$ כאשר $n \to \infty$.

$$O(g) = \{f : \mathbb{N} \to \mathbb{N} : \exists c>0\ \exists n_0\ \forall n \ge n_0 \quad f(n) \le cg(n)\}$$

או במילים אחרות $\frac{f(n)}{g(n)} \le c$ עבור $n \ge n_0$ אם ורק אם $\frac{f(n)}{g(n)} \to 0$ כאשר $n \to \infty$.

$$O(g) = \{f : \mathbb{N} \to \mathbb{N} : \forall c>0\ \exists n_0\ \forall n \ge n_0 \quad f(n) \le cg(n)\}$$

הגדרה שקולה: $f : \mathbb{N} \to \mathbb{N}$ נאמר ש- $f \in O(g)$ אם קיים $c>0$ כך ש- $T(n) = O(n^2)$ לכל n החל ממקום מסוים.

דוגמה: נניח ש- $T(n) = n \log n$ לכל n

$$T(n) = \Omega(n)$$
$$T(n) = O(n^2)$$

אבל לא קיים פולינום d כך ש- $T(n) = \Theta(n^d)$ - לכן זו טענה נכונה.

הגדרה נוספת: יהיו $g : \mathbb{N} \to \mathbb{N}$.

$$O(g) = \{f : \mathbb{N} \to \mathbb{N} : \exists c>0\ \exists n_0\ \forall n \ge n_0 \quad f(n) \le cg(n)\}$$

או במילים אחרות $\frac{g(n)}{f(n)} \ge c$ עבור $n \ge n_0$ אם ורק אם $\frac{g(n)}{f(n)} \to \infty$ כאשר $n \to \infty$.

$$\Omega(g) = \{f : \mathbb{N} \to \mathbb{N} : \forall c>0\ \exists n_0\ \forall n \ge n_0 \quad f(n) \ge cg(n)\}$$

או במילים אחרות $\frac{g(n)}{f(n)} \to 0$ כאשר $n \to \infty$.

הוכחה: (יש שיעדיפו לומר אלגברי, אך הכוונה היא לאותו הדבר) נראה כי $f \in \Theta(g)$ אם ורק אם $f \in O(g)$ וגם $f \in \Omega(g)$.

הוכחה: $\log n, an, 1 \in O(n)$

$$an \in \Theta(n)$$
$$\log n \in O(n^\varepsilon)$$

הוכחה: נראה כי $f \in O(g)$ אם ורק אם $g \in \Omega(f)$. (יש שיעדיפו לומר אלגברי, אך הכוונה היא לאותו הדבר) נראה כי $f \in \Theta(g)$ אם ורק אם $f \in O(g)$ וגם $g \in O(f)$.

טענה לא מוכחת: $f \in \Theta(g)$ אזי $f \in O(g)$ וגם $f \in \Omega(g)$

סדרת פיבונאצ'י

סדרת פיבונאצ'י, האפורטאת אוסדרת a_n, נוסחת הנסיגה

$$\begin{cases} a_0 = 1 \\ a_1 = 1 \\ a_{n+2} = a_n + a_{n+1} \end{cases}$$

הקבוצה שלה: 1, 1, 2, 3, 5, 8, 13, ...

איך נחשב אותה?

<u>דרך I</u>:

```
Fibonacci(n)
if (n=0 or n=1)
    return 1
else
    return Fibonacci(n-1) + Fibonacci(n-2)
```

כל אחת מהקריאות, מלבד שתיים שבסוף, מבצעת אותן רק שלוש פעולות. לכן זמן הריצה שלו הוא אקספוננציאלי:

$$T(n) = T(n-1) + T(n-2) + O(1)$$

↩ נשפר למטה את הקונסטרוקציה

<u>טענה</u>: $a_n \geq 2^{\lfloor \frac{n}{2} \rfloor}$

<u>הוכחה</u>: באינדוקציה על n.

בסיס הדורה עבור $n=0, n=1$: $a_n = 1 \geq 2^{\left(\frac{n}{2}\right)} = 1$

נניח עבור n ונוכיח עבור $n+2$:

$$a_{n+2} = a_{n+1} + a_n \geq 2a_n \geq 2 \cdot 2^{\lfloor \frac{n}{2} \rfloor} = 2^{\lfloor \frac{n}{2} \rfloor + 1} \geq 2^{\lfloor \frac{n+2}{2} \rfloor}$$ ☺

מכאן, $a_n = \Omega(2^{\frac{n}{2}})$, ולכן $T(n) = \Omega(2^{\frac{n}{2}})$.

זה שקול ל- $T(n) = \Omega(a_n)$ שהוא אחד מסדרת פיבונאצ'י.

כי בנוסף ל $T(n-2)+T(n-1)$ נוסף עוד $O(1)$ שהוא לא דומיננטי

פתרון II:

```
Fibonacci (n)
F[0] ← 1
F[1] ← 1
for i = 2 ... n
    F[i] = F[i-1] + F[i-2]
return F[n]
```

זה נותן לנו $\Theta(n)$ (ליניארי)

פתרון III: אסתרה על הקונה כמו של אלגוריתם

נגיד $A = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$ (כאן)

$$\begin{pmatrix} a_n \\ a_{n-1} \end{pmatrix} = A \begin{pmatrix} a_{n-1} \\ a_{n-2} \end{pmatrix} = AA \begin{pmatrix} a_{n-2} \\ a_{n-3} \end{pmatrix} = \dots = A^{n-1} \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

עכשיו הבעיה שלנו היא לחשב אלגוריתם זה אפשר בזמן:

```
Power (A, n)
if n=1
    return A
B ← Power (A, ⌊n/2⌋)
if (n is even) return
    return B·B
else
    return B·B·A
```

```
Fibonacci (n)
return
[Power(((1 1)(1 0)), n-1) · (1)(1)]_1
```

← איבר ראשון

זמן $A^{13} = (A^6)^2 A = ((A^3)^2)^2 A = ((AAA)^2)^2 A$

מה מן הדברים שם זה מאוזן הכי טוב בחלקה?

$$T(n) = T(\lfloor \tfrac{n}{2} \rfloor) + O(1)$$

כאן נוסחה כללית של $O(\log n)$

טענה $T(n) = O(\log n)$

הוכחה: בדרך כלל יהיה ש: $T(n) \le c \log n$

נניח שזה $n = 2^k$.

$T(2) \le c\log 2 = c$ עבור $k=2$

נניח נכונות עבור $k-1$ ונוכיח עבור k: $T(2^{k-1}) \le c\log 2^{k-1}$ (1)

$$T(2^k) = T(2^{k-1}) + d \overset{(1)}{\le} c\log 2^{k-1} + d = c\log 2^k - c + d \overset{?}{\le} c\log 2^k$$

כלומר נדרוש ש- $T(2) \le c$ וכן $d - c \le 0$

$\Rightarrow$ נוסיף $T(2) = c$ ונדע שהוא טוב גם ל- (1)

לא הבנו עדיין כאן שמדובר כאן, אבל תבלע נסתר שאפשר לי שפרנץ שום דבר.

אם יש לנו מספר שהוא $O(n)$ אבל לוקח לנו $O(\log n)$ ביטים בשביל לייצג אותו בבסיס בינארי.

- פעולה חיבור של שני מספרים שאורך הייצוג שלהם הוא $O(n)$ לוקחת $O(n)$ פעולות.

- פעולת הכפל של שני מספרים שאורך הייצוג שלהם הוא $O(n)$ לוקחת $O(n^2)$ פעולות (אפשר לחשב את זה כמו בטור)

מדגלה II הפעולה היקרה ביותר היא $F[i] \leftarrow F[i-1] + F[i-2]$ עבור $i = 2 \dots n$

ניכר ש- $a_n = \frac{\varphi^n}{\sqrt{5}}$ ולכן לוקח לנו $O(n)$ ביטים לייצג אותו.

אבל אם נשתמש בנוסחת הנסיגה עלולה נקבל

$$T_B(n) = T_B(n-1) + O(n)$$

מספר הביטים שדרוש לייצג n

ומנוסחת הפתרון אנקבל $T_B(n) = O(n^2)$.

מדגלה III הפעולה היקרה ביותר אפשרית מרובעת. אז לא הפסדנו מרובעת לוקח $O(n^2)$.. אחרי שהפסדנו $O(n^2)$ כפל מרובעת ביטים, ולכן הם כפלים תיבורים.

היה לנו $B = A^{\frac{n}{2}}$. נשאל מה אפשר לעשות עם $A^{n/2}$

$$\begin{pmatrix} a_n \\ a_{n-1} \end{pmatrix} = \underbrace{A^{n-1}}_{\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} a_{11} + a_{12} \\ a_{21} + a_{22} \end{pmatrix}$$

⇐ כל A^n יש מספרים שאותם ניתן לחשב ב $O(n)$ אם נאכסן אותה בזיכרון המחשב בצורה שתאפשר גישה ב- $\frac{n}{2}$ אותם ניתן מתקבל $O(n)$. נוסף שני מספרים ראלה לשלה לנו $O(2^n)$. ואנו מאפשרים לה:

$$T_B(n) = T_B(\lceil \tfrac{n}{2} \rceil) + O(2^n)$$

שלה $T_B(n) = O(2^n)$

נשאלה עם מה: תשכנו את סיבוכ' ב-2 דרכים מתקבלות לנו הדגמת ב- III ראוה הניגו שיפור עצום. אבל השיכנים מאוד יקרים של המחשב והשתכנו על פעולות ביטוי, קיבלנו את אותה תוצאה.

⇐ אה שאקבלים תלוי מאד שמורים יפור.

מחקה שהקלט n אפשר לייצוג לה לא נתנין להניח שחיבור מחיק בהצמן לוקח $O(1)$.

אלה catch הוא שלנים לעשות את מה שהאת מעשה

מר' אלעד
06.11.1 ⑭

קידוד האסמן

משפט שכתוב מיד. קוד כן ניתן להפיק מיידית אחרת. לפי טיב למשפט את כי הקוד ולאו לא סופי קוד כן ניתן להפיק מיידית אחרת. לפי טיב למשפט

קוד הוא מיידי = כמה ניתן הסמלים אחרים X, קוד כזה הוא $c: X \to \{0,1\}^*$
($\{0,1\}^*$ היא קבוצת סדרות של אפסים ואחדים).

נקבע לכל סדרה אחרים: קידוד של סדרה $(a_1,\dots,a_n)$ יהיה השרשור $c(a_1),\dots,c(a_n))$.

למשל, אם $X=\{a,b,c\}$ ו- $a\to 0$, $b\to 00$ אז $abc\to 0000$
$c(a)=1$, $c(b)=01$, $c(c)=10$.

הבעיה - קוד c כזה יתן אפשרות יותר אחת כ-$\{0,1\}^*$ ולכן לא יהיה מיידי.

למשל: $a\to 0$
$b\to 10$
$c\to 01$
$d\to 00$

קידוד אחר.

אלגוריתם כיצד לקודד רשימת סמלים X כך שהקוד יהיה מיידי.

עיקרון: $a\to 0$
$b\to 10$
$c\to 110$
$d\to 111$

אלגוריתם: קוד $c: X\to\{0,1\}^*$ נקרא קוד הופמן אם עבור כל $n\geq 2$ מתקיים $\sum_{i=1}^{n} c(a_i)\cdot 2^{-|c(a_i)|} = 1$.

למשל קיים $c:\{a,b,c,d\}\to\{0,1\}^*$ כך ש- $n=4$.

עבור $n\neq 4$ לא קיים קוד הופמן.

$$\underbrace{0}_{a}\underbrace{10}_{b}\underbrace{10}_{b}\underbrace{0}_{a} \longleftarrow abba$$

קוד הופמן לא יתכן שיהיה מילה של מילה אחרת בקוד.

דוגמה: $a\to 0$, $b\to 10$, $c\to 11$ הוא קוד הופמן.

$a\to 0$
$b\to 01$
$c\to 11$

דוגמה: $a\to 0$
$b\to 00$
$c\to 11$ לא יתכן קוד הופמן.

מסתיבה עבורת הקוד הרשע.

מתקבה β רו. אם היה המשך כלשהו הכלו הייתה רישא של מילה אחרת בקוד. הנחה שלילה שהרי מילה מתקבה שכן מילה רישא קוד רישא לעצמה תמיד.

$p_T(z)$ = הסתברות של z מפלט T או לחילופין אורך הקידוד של z.

$B(T) = \sum_{x \in Z} F(z) \cdot p_T(z)$ אורך הקידוד האחרון

$F = (f_1, \ldots, f_n)$

$X = \{a_1, \ldots, a_n\}$

אחרון: $Z \in X$ כך ש $F(z)$ היא התדירות של z, לדוגמה אורך קידוד

מהיפוך קוד רישא $\{0,1\}^* \to X$ המוגדר ע"י פלט T, ומהיפוך תדירות z

אורשהו כדי לחשב את הפלט.

ASCII שלפי אורך מסוים לתדירות האותיות ואותנו רואים לקידוד אותו מחלקה של קודי האסמן הקוד משמש לדחיסה. ניתן שיש קופץ יפרע עם תווי

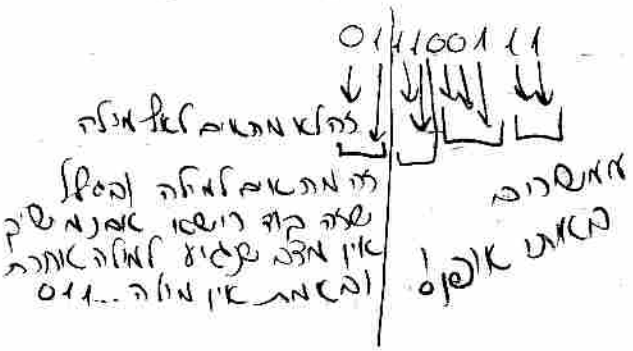
ושלגים שמאלה מוסיפים 0.
ושלגים ימינה מוסיפים 1

מאלה קוד, היה יוצא שאין קוד הוא רישא של אולי קוד אחרת.

⑤

1001, 11, 01 — קידוד של אותיות הקלט

פענוח של הקידוד 1001110



נניח שקיבלנו את X, F המייצגים קוד רישא המינימלי B – אך אולי לא אופטימלי.

הוכיחו שקיים קוד רישא אופטימלי T המייצג את X.

פתרון:

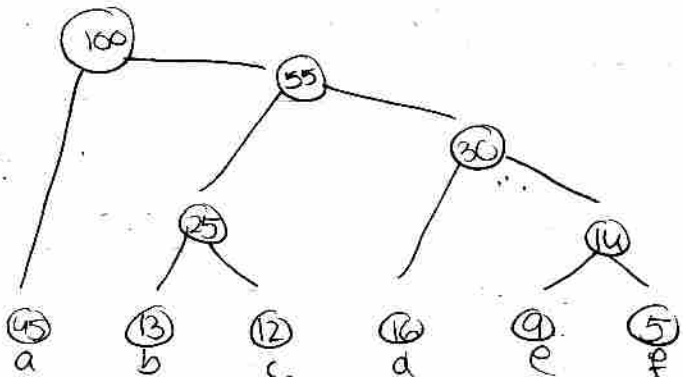
X	a	b	c	d	e	f
F	45	13	12	16	9	5

$X = (a_1, \ldots, a_n)$ קלט: קוד הא"ב

$F = (f_1, \ldots, f_n)$

אלגוריתם:

```
n ← |X|
Q ← build-min-heap(X, F) // min-priority-queue
for i = 1 to n-1 {
    create a new node w;
    left[w] ← x ← extract-min(Q)
    right[w] ← y ← extract-min(Q)
    f(w) ← f(x) + f(y)
    insert(Q, w)
}
return extract-min(Q) // return the root of the tree
```



קוד האופטימלי מינימלי ע"י אלגוריתם חמדני (ולכן הקשר להרצאות) אמנם אם
הניחנו כי מין הסרטים אונתני רוצים שאלגוריתם החמדני הכי יעילה
יהיה הקידוד הכי קצר.
למרות לה אלגוריתם החמדני לא נתן תמיד לה אחד שגורם לאופן לקבלי את
מה שהכי טוב בהרצאה למטה מה יקרה שלה היה הכי טוב באופן אלגוריתמי.
ההוכחה של האלגוריתם נעשית באופן מאוד אופייני לאלגוריתמים
חמדניים.
הרעיון של שאנחנו לקוד אופטימלי הוא זה מינימי אפשר (לעומת עם
קידוקוד יש 0 או 2 בנים). למה? כי לאחרת:

למעשה יהי X אוסף תווים כך שלכל $x \in X$ שתדירות $f(z)$
יהיו x,y שכרים בשני תדירות מינימלית. אז קיים קוד חישוב
אופטימלי המוגדר ע"י עץ T כך ש- $d_T(x) = d_T(y)$ ו- x,y שונים
כך בבינו האחרון.
הוכחה: ניקח קוד אופטימלי TOPT. נניח לפי הם שלים בשלי
שונה מקסימלי ב- TOPT ונית שהם אחים (אפשר להניח את זה
כי העץ מלא). כעת נבנה עץ חדש T: נית מהעץ $f(x) \geq f(y)$
ו $f(a) \geq f(b)$ ונקח נתלוי את גובה הקוד של a בעץ של x -
העץ שמקבלים נסמנו T. למשל:

נשלח תשובה לכאורה שאינו מאופטימלי? 8

$$B(T_{OPT}) - B(T) = \sum_{z\in X} f(z) d_{T_{OPT}}(z) - \sum_{z\in X} f(z) d_T(z) =$$

$$= f(x) d_{T_{OPT}}(x) + f(a) d_{T_{OPT}}(a) - f(x) d_T(x) - f(a) d_T(a) =$$

$$(x) d_{T_{OPT}} \quad \| \quad (a) d_{T_{OPT}} \quad \| \quad d_T(x) \quad \| \quad d_T(a)$$

$$= \left(f(a) - f(x)\right)\left(d_{T_{OPT}}(a) - d_{T_{OPT}}(x)\right) \leq 0$$

↳ לפי ההנחה על x ו-a.

$$B(T_{OPT}) \leq B(T)$$

$\Leftarrow$ אם T אופטימלי,

כלומר נתון אז a ו-y מעלי T ונקבל עץ T'

ומאותו אופן $B(T) \geq B(T') \Leftarrow T'$ אופטימלי.

! T אוזלת קול כנדרש. (11)

<u>טענה</u>: יהיו X, F כמו מקודם! x, y אחרים בעלי תדירות מינימלית, אז קיים $X' = (X \setminus \{x,y\}) \cup \{z\}$ כך ש- $f(z) = f(x) + f(y)$ שאם T'_{n-1} אופטימלי עבור X' אז עץ T_n המתקבל ע"י הוספת שני בנים ל-z ב-T'_{n-1} הוא אופטימלי עבור X.

<u>הוכחה</u>:

$$\forall z \in X \setminus \{x,y\} \quad d_{T_n}(z) = d_{T'_{n-1}}(z)$$

$$d_{T_n}(x) = d_{T_n}(y) = d_{T'_{n-1}}(z) + 1$$

$$f(x)d_{T_n}(x) + f(y)d_{T_n}(y) = \left(f(x)+f(y)\right)\left(1 + d_{T'_{n-1}}(z)\right) =$$

$$= f(z) + f(z) d_{T'_{n-1}}(z) =$$

$$\sum_{x,y,z \in Z} d_{T_n}(z) f(z) \quad \text{זה אותו דבר.} \quad \left\{ \begin{array}{l} f(z) d_{T'_{n-1}}(z) \\ \sum_{x,y,z \in Z} d_{T'_{n-1}}(z) f(z) \end{array} \right.$$

שכולל יותר ניסוף

(*) $B(T_n) = B(T'_{n-1}) + f(x) + f(y)$

נניח בשלילה ש-T_n לא אופטימלי, כלומר קיים עץ T_n^{OPT} אופטימלי עבור X שמתקיים $B(T_n) > B(T_{n-1}^{OPT})$

שנת ש- T_n^{OPT} y,x צמתים (מהם מה הקודמת).
נבנה T_{n-1} מ- T_n^{OPT} ע"י החלפת תת-העץ ב- ש
ונזכור ש- $f(x)+f(y)=f(w)$. אנו מאותו חישוב כמו (*)

$$B(T_n^{OPT}) = B(T_{n-1}) + f(x) + f(y)$$

$$\Rightarrow B(T_{n-1}) = B(T_n^{OPT}) - f(x) - f(y) <$$

$$< B(T_n) - f(x) - f(y) = B(T_{n-1}^{OPT})$$

צ"ל סתירה כי לא יכול להיות שתרון יותר טוב מאופטימלי!

מה קרה פה בעצם? הנחנו קיום פתרון אופטימלי ולבנינו דומה T_{n-1}^{OPT} ומאנו בנינו פתרון T_n. הנחנו שאילה שקיים פתרון יותר טוב T_n^{OPT} ומאנו הגלתנו לעובדה פתרון יותר טוב מאושר T_n^{OPT} לבעיה הקטנה. ולכן סתירה כי היתה לנו מלכ ש T_{n-1}^{OPT} הינו הכי טוב!

הוכחת נכונות קלה האופן <u>משפר באינדוקציה</u> ושימוש מלמעלה!

- שפוך משימות מומשקלות -

נזכיר את בעיית שפוך המשימות? יש n משימות (t_i, v_i) וצריך לשפוך אותם מקסימום $(v, 0)$ כך שלא תהיה הפסקה וגם מספר מקסימלי של נוסעים.

סיכמנו אז כי f_i שאינו את המשימות בסדר עולה לפי t_i ומצד שני מתנהג את המשימות שלנו מתנהגות עם הקיבולת שלה v.

שרשרנו נוסחת את הלינים - ניתן למשימות אשכלות.

בפרט $f_{n-1}(t_i, v_i)$ ומשקלות v_i לכל i כך ש $t_i \geq t_n - v$ הוחזרה $\max_{i=1}^{n-1} f_{n-1}(t_i, v_i)$ אחרת $1 + \max_{i=1}^{n-1} f_{n-1}(t_i, v_i)$.

מדוע שאלגוריתם הקבלה הבא פותר את בעיית האריזה הכללי שהוא בעיית המקסימום:

נמיין, הקבלה הבא פותר את הלינים:

האלגוריתם ימדר את הקטעים $\{1, 2, 3, 4, 5, 6, 7\}$ לפי האשכול שהם נמצאים בו, כלומר שהקטעים 1, 5, 7 נמצאים באשכול 1, הקטעים 2, 4 באשכול 2, והקטע 3 באשכול 3.

1
2
3
4
5
6
7
0
1
2
3
4
5
6
7
1
2
3
4
5
6
7
0
1
2
2
3
4
5
6
7
0

טענה: 1) אם $O \in n$ אז $\{1,\dots,n\}$ אופטימלי שווה ערך ל $\{1,\dots,p(n)\}$

2) אם $O \notin n$ אז O אופטימלי שווה ערך ל $\{1,\dots,n-1\}$

הוכחה:

1) מכיוון ש $\{1,\dots,p(n)\} \subseteq \{1,\dots,n\}$ ו O אופטימלי קיים $O(p(n))$ כך ש- $\sum_{i \in O} v_i < \sum_{i \in \{1,\dots,n\}} v_i$ נוסף לזאת, הדפים אשר n וקטע $\sum_{i \in O} v_i < \sum_{i \in \{1,\dots,p(n)\}} v_i$ אז $\{1,\dots,p(n)\}$ מתנגד לכך ש O אופטימלי.

2) $\{1,\dots,n-1\} \subseteq O(*)$ כי אחרת אפשרי שווה ערך ל $\{1,\dots,n\}$ והוא אופטימלי שווה ערך ל $\{1,\dots,n\}$. O נותן אקסיאם (*) כי הפתרונות האפשריים של $\{1,\dots,n\}$ יכלו מכיוון שהוא פתרון ל $\{1,\dots,n-1\}$ (מכלל (*)) אז הוא לא נותן אקסיאם על הפתרונות של $\{1,\dots,n-1\}$. (!!) $\{1,\dots,n\}$

$$OPT(n) \in \{OPT(n-1),\ v_n + OPT(p(n))\}$$

שתיהן אופטימלי
מעדיפים את זו שבה n או לא.

מסקנה: $OPT(n) = \max\{OPT(n-1),\ v_n + OPT(p(n))\}$

וכך נראה לנו אלגוריתם רקורסיבי. רק שנראה שלפני שנשמור את התשובה נבדוק אם היא כבר נמצאת שם כי אחרת נחזור אקספוננציאלי.

והאלגוריתם הוא:

Compute-OPT(j)

M[0] ← 0

if M[j] is empty

M[j] ← max{compute-OPT(j-1), v_j + compute-OPT(p(j))}

return M[j]

הפונקציה הבאה מקבלת את המערך M ואת האינדקס j.
היא מחזירה את הקבוצה S עבור תת-הבעיה הראשונית j.

Find Solution (j)

if $j = 0$ return $\emptyset$

if $(v_j + M[p(j)] \geq M[j-1])$

return {Find-Solution($p(j)$)} $\cup \{j\}$

return Find-Solution$(j-1)$

דוגמה:

j	$p(j)$
1	0
2	0
3	2
4	2

בכיתה הוצג קוד מפורט יותר של הפונקציה הנ"ל.

רגרסיה לינארית בשיטת הריבועים הפחותים

נתונה קבוצת נקודות $\{(x_i, y_i)\}_{i=1}^{n}$. רגרסיה לינארית מוצאת ישר $y = ax + b$ שממזער את סכום ריבועי המרחקים האנכיים (בכיוון ציר y) של הנקודות מהישר:

$$Err(y = ax+b, \{(x_i,y_i)\}_{i=1}^{n}) = \sum_{i=1}^{n}(ax_i + b - y_i)^2$$

ניתן להוכיח שהפתרון הוא:

$$a = \frac{n\sum x_i y_i - (\sum x_i)(\sum y_i)}{n\sum x_i^2 - (\sum x_i)^2} \qquad b = \frac{\sum y_i - a\cdot\sum x_i}{n}$$

אפילו אם יש הנקודות שלנו למעלה מקו ישר לגמרי.
אנו צריכים לבחון קו שיעבור כמה שיותר נקודות ומעט מאוד
נקודות יהיו מתחת לקו הישר:

כמו, אפילו מדובר שלפי דעתי בלבד כי אפשר לקחת שני
נקודות מאוכלוסיה לדוגמה יש ישפ אחד ויחיד שמתאר.
אנו מדובר שאנחנו רוצים למצוא איזה trade off מין
מספר הנקודות לעומת הקירוב.

ניסוח פורמלי לבעיה: נתונים $P=\{(x_i,y_i)\}_{i=1}^n \subseteq \mathbb{R}^2$
(נשים לב $P=\{p_i=(x_i,y_i)\}$) (מפה תוקה של P מסומנים,
מעוניינים בקו ישר ℓ שמחובר P ומקיים את הקריטריונים
הבאים ככל האפשר:

- מספר הנקודות מעל הקו הוא מספר הנקודות מתחת לקו.
- מספר הנקודות הכי קרוב לקו ℓ הוא $C>0$.

דוגמה

הקו: מתקבל ע"י $\ell_1 + \ell_2 + C$
שני מסומנים
הנמוך
השני
השלישי

נסמן e_{ij} לטעות הריבועית הזעירה ביותר עבור $p_i, \ldots, p_j$
כאינטרפולציה ע"י פונקציה לינארית.

טענה: אם נתונה חלוקה אופטימלית של סיומים אחרון $p_n, \ldots, p_j$
אז לפחות הפתרון האופטימלי הוא
$$OPT(n) = OPT(j-1) + e_{jn} + C$$

הוכחה: כרגיל, אם קיימת חלוקה טובה יותר ל $p_1, \ldots, p_n$
מאשר החלוקה $s_k, \ldots, s_1$ אז ניקח אותה ונוסיף לה את
$p_j \ldots p_n$ ונקבל חלוקה ל $p_n, \ldots, p_1$ טובה יותר
מאשר $s_k, \ldots, s_1$ המתקבלה לאופטימלית של $s_{k-1}, \ldots, s_1$.

$\Leftarrow$ $s_{k-1}, \ldots, s_1$ אופטימלית עבור $p_{j-1} \ldots p_1$

$\Leftarrow$ $OPT(n) = OPT(j-1) + e_{jn} + C$

- $OPT(n)$: קלט עבור $s_k \ldots s_1$
- $OPT(j-1)$: קלט עבור $s_{k-1} \ldots s_1$
- e_{jn}: שגיאה של סיומים אחרון
- C: הוספנו קטע

לכן מצאנו את נוסחת הנסיגה הבאה:

$$OPT(n) = \min_{1 \le j \le n} \{ e_{jn} + C + OPT(j-1) \}$$

וכך מצאנו אלגוריתם - לשאיר טבלה שיש בה את
ה חישוב המינימום.

Segmented-Least-Squares (n)

```
M[0] ← 0
for ij s.t. j ≤ i
    calculate eij // using the formula
for j = 1, ..., n
    M[j] = min_{1≤i≤j} {eij + C + M[i-1]}
return M[n]
```

ברור כי סיבוכיות האלגוריתם היא $O(n^2)$ עם n המשתנים
המופיעים. יש לציין שם n נקודות. ניקח סיומים

אופטימלי. לסיבוכיות יש משמעות אחרת.
למשל הטענה שאנחנו מקבלים את האופטימום עבור כל
קלט יתכן ואנו אנחנו הרי יודעים שלגוריתם בעל
זמן ריצה הינדיקציה.

פתרון בעיה ברשימה מתיאלץ 1:

$A = [x_1, \ldots, x_n]$ $\forall i \in [1..n]$ $1 \leq x_i \leq n$

אפשר לחשוב על המערך הזה כפונקציה:

$A: \{1, \ldots, n\} \longrightarrow \{1, \ldots, n\}$

ונרשום את שם על כל ריצף מכוון - כל קודקוד מתאים
ל - A של הקודקוד למשל

i	1	2	3	4	5
A(i)	4	1	4	2	3

מתאימים קודקוד
שממנו הכניסה
שווה היא 2.

בגלל כלל חיבת להיות מסלול ואפשר שנכנסנו למסלול
כי הקודקוד עם דרגת כניסה 2.
אבל אנחנו פשוט מתניקים 2 שונטרים ומתפשים מסלול
מ"רשימה" הבאה.

זיווג מקסימלי: משפט הול-כאב

(ת"ל) גרף דו-צדדי $G=(V,E)$, $V=L\cup R$

(כלומר) מהו גודל הזיווג M ב-G המקבל מקסימלי.

זיווג = זיווג M מגרף G הוא תת-קבוצה של E כך שכל קודקוד ב-V הוא קצה של היותר קשת אחת ב-M.

דוגמה:

הדגמה כמו שהוסיפו את הנקודה s ו-t, והוסיפו קשתות מ-s אל כל קודקוד ב-L, והוסיפו קשתות מכל קודקוד ב-R אל t.
הגדירו את קשתות הגרף מ-L ל-R. וכמו הקשתות שיש מהן לפחות קיבולת 1.

למה: יש ב G זרימה שלמה (שלכל הקיבולים שלו) בגודל שווה לגודל הזיווג המקסימלי ב-G.

כלומר הלכיוון - $f(s,V)=|f|=\sum_{v\in V} f(s,v)=f(\{s\},V)$

שלכל זרימה שלמה (שלכל הקיבולים שלו) בגודל שווה לגודל הזיווג המקסימלי ב-G.

(סימון: אם $A,B\subseteq V$ אז $f(A,B)=\sum_{u\in A}\sum_{v\in B} f(u,v)$ $c(A,B)=\sum_{u\in A}\sum_{v\in B} c(u,v)$)

לכל משפט שהוכחנו בדיוק הלכיוון הקל תקף בכל שווה ושווה לגודל הלכיוון.

הוכחה: עלינה לכיוון ה f ואפיין $f(u,v)=1$: $M=\{(u,v)\in L\times R : f(u,v)=1\}$

ואז בדיוק שמתקיים $|M|=\sum_{v\in R}\sum_{u\in L} f(u,v)=f(L,R)=f(\{s\}\cup L,R\cup\{t\})=|f|$

והקיבועים הן $f(n,\lambda)-c(n,\lambda)=f(n,\lambda)$.

$$E' = \{(\lambda,\eta) \in \Lambda\times\Lambda \mid f(\lambda,\eta) \neq c(\lambda,\eta)\}$$

ונראה $f_G \circ g = id_{E'}$ כאשר $G = (V, E')$

- נבנה את $f \to f+g$ ונבדוק f_G מתאפס.
- הוא לדוגמה g שאלגוריתם f ומחסיר קומבינציה ומשאיר 0
- נמצא את הקומבינציה המינימלית g ו"מחסיר"
- G את קודקודי f_G מחסיר כך g-ב
- $f_G \leftarrow G$
- נאתחל את f לפונקציה האפס.

אלגוריתם FF

ומשנותן לה 0 אין אור?

המסקנה היא שלגוריתם אקספוננציאלי בגודל הקלט. אך הבעיה בדיסציפלינה אלגוריתם פולינומי.

(ii) שמשתמש ב-M ל-τ_0. ובאותה הדרך כמעט.

ב-τ שמשתמש ב-M וזה נותן 1 אם קודקוד ב-R ל-R ($1-1$- מכוון ההפוך) נותנים 1 ו-0 עם קודקוד הוכחה: שפיר לדוגמה f ל"הלאה 1 עם קצה מלילים מן τ עם כלילים M.

(ii) אפס $1-c(\lambda',\zeta) \geq f(\lambda',\zeta)$. מדוע?

$$\sum_{\eta:(\eta,\lambda)\in O\tau(M)} f(\lambda,\eta) = \sum_{\eta:(\eta,\lambda)\in O\tau(M)} c(\lambda,\eta) \geq f(\lambda',\zeta)$$

ζ הוא הקודקוד היחיד שאינו יוצא קצה שכניסת η אפס

$$\sum_{\eta':(\eta',\lambda)\in O\tau(M)} f(\eta',\lambda) \geq c$$ לכן בהכרח

M ניתן כי אם לאפס τ (ואז בשתי קלילות M- אינו שיבאר אינו (כמאר סביב הלילות החיובית מאנו) אל הלילה שיבאר אינו (כמאר סביב הלילות החיובית מאנו)

Basketball Elimination

- נתונה קבוצת $S = \{1, \ldots, n\}$ וקבוצה z.
x קבוצה y נקראת מכרעת עבור z אם w_x משתקים. לקבוצות x, y יש בדיוק g_{xy} משחקים שטרם שוחקו ביניהן.
- נשאלת השאלה z יכולה לסיים את האליפות (אולי בתיקו) וקבוצה z יכולה לסיים עם טיבו מאזן ניצחונות מכרעת עבור z.

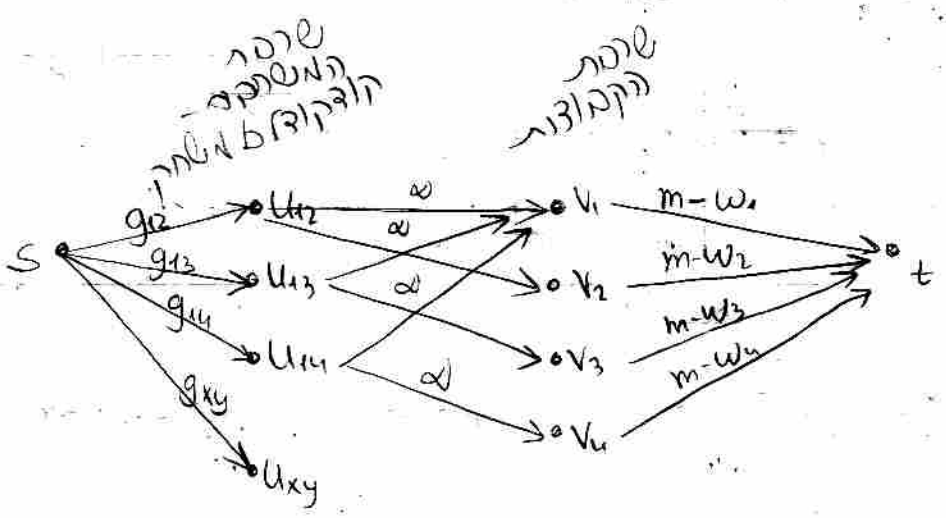
לדוגמה:

x	w_x
1	20
2	18
3	17
4	9

g_{xy}	1	2	3	4
1	0	1	6	4
2	1	0	1	4
3	6	1	0	4
4	4	4	4	0

נסתכל על קבוצה 1. יש לה 9 משחקים בינהן. ואם ניצחו את כולם יגיעו ל-37 משחקים. אבל יש קבוצה בינהן שתגיע ל-$\frac{37+9}{2} = 23$ משחקים. לכן 4 יכולה להשיג לכל היותר $m = 20 + 9 = 29$ ניצחונות. לכן 4 לא יכולה להיות אלופה.

אבל מה עושים? נשתמש בהפיכה של אלגוריתם מקסימום זרימה.



הקבוצה z יכולה להיות אלופה אם $\{v_x, v_y\} \cap \{u_{xy}\}_{\text{קשתות שנכנסות}} \cap \{t\}_{\text{קשתות שיוצאות}} = \emptyset$

ההקבלה שלנו z היא לכל תלת אדרשת. היש מינימום מכל
המשתקים שנשארו לה ובכולם יש נפחונות.
ההקבלות מוגדרות להיות $g_{xy} = (h_{xy}, s)c$

$$c(u_{xy}, v_x) = c(u_{xy}, v_y) = \infty$$

$$c(v_x, t) = m - w_x$$

זה לא מיידה מהשת כלאת כי איזו תוקפה של המשתקים שנשארו
מן ההקבלות S כפי שאלף הסוכנים x מוגדרה יותר
$m - w_x$ משתקים.

נשים $g^* = \sum_{x \in S} g_{xy}$. אנו אם קיבלנו מינימום g^*
אז ההקבלה שלנו z עשויה לעבוד.

g^* ונראה שנפחונות ב- g^* המשתקים שנשארו מן שלף הסוכנים
x לאו דווקא מיותר $m - w_x$ משתקים מתוך הסוכנים שנשארו,
אחרת אפשר להגדיר מינימום מדשת. לכן הגדרה יהיה אף g^*.
מסקנה: ההקבלה שלנו עשויה לעבוד אמ"מ קיבלנו מינימום מדשת לפי
g^*.

שוק-פלקטון מופיע הן שהגיע בו היא שהוא לא אומר איך
להגדיר אפשרי אותן ואת ובנוסף לקבוע שמרים ארוכים לאב.
לעומת

אם נמחר ב פשם את האפשרי הזרוע נדרוש לעשות $2 \cdot 10^6$ אינדיקציה

ותהי $G=(V,E)$ רשת זרימה ופונקציות משקל $\omega_1: V \to \mathbb{R}^+$ ו- $\omega_2: E \to \mathbb{R}^+$. ותהי קבוצה $S \subseteq V$.

נגדיר לכל $A \subseteq V$ המכילה את S את $val(A) = \sum_{v \in A} \omega_1(v) + \sum_{\substack{u \notin A \\ v \in A}} \omega_2(u,v)$

(כאשר $\omega_2(u,v)=0$ אם $(u,v) \notin E$)

אינטואיציה: ניתן לחשוב על הרשת כעל מערכת של ערים שחלקן מחוברות בכבישים חד-סטריים. $\omega_1(v)$ מייצג את כמות האנשים שנמצאים בעיר v בתחילת היום, ו- $\omega_2(u,v)$ מייצג את כמות האנשים שיכולים לנסוע בכביש (u,v) בכל יום. S מייצגת את קבוצת הערים שבהן נמצאים מרכזי הביטחון. בכל יום, אנשים שנמצאים בעיר שאינה במרכז הביטחון יכולים לנסוע לעיר אחרת דרך כביש. אנשים שנמצאים במרכז הביטחון נשארים שם.

<u>משפט</u> - Max-flow - Min cut

יהי $(A, V\setminus A)$ חתך (חתך = חלוקה שקודקוד המקור שייך לשתי הקבוצות בחלוקה, וקודקוד היעד שייך לשנייה) כך ש- $s \in A$ ו- $t \notin A$. נסמן ב- f_{max} זרימה מקסימלית ב- G. נסמן $|f_{max}|$ את גודל הזרימה. אזי קיבולת החתך $(A, V\setminus A)$ היא $|f_{max}|$. בנוסף, $|f_{max}|$ הוא המינימלי שבין כל החתכים ב- $G_{f_{max}}$ (גרף השאריות של f_{max}) שבהם s ו- t נמצאים בקבוצות שונות.

<u>שאלה</u>: איך למצוא זרימה?

- נאתחל את f לזרימת האפס.

$f(S,T) = \sum_{\substack{u \in S \\ v \in T}} f(u,v)$

$|f| = f(S,T) \leq C(S,T)$

אלגוריתם שלפיו נתחיל עם $|f| = C(S,T)$ של הדוגמה. בסוף נקבל $|f_{max}| \leq \min_{(S,T)} C(S,T)$. אפשר להראות ש- $|f_{max}| = C(A, V\setminus A)$:

אז הצלחנו להראות שקיים $c(S,T)$ שמספר המסלולים הזרים בקשתות הוא לכל היותר $c(S,T)$. אבל אנחנו רוצים שמספר המסלולים הזרים בקשתות יהיה בדיוק $|f_{max}|$. נראה שמספר המסלולים הזרים בקשתות הוא $c(A,V\setminus A)=|f_{max}|$.

$$|f_{max}| = f_{max}(A,V\setminus A) = \sum_{\substack{u\in A\\ v\in V\setminus A}} f_{max}(u,v) = \sum_{\substack{u\in A\\ v\in V\setminus A}} c(u,v) = c(A,V\setminus A)$$

כי $f_{max}(u,v)=c(u,v)$ לכל קשת f_{max} שעוברת בחתך (u,v)

☺

עכשיו נוכיח את המשפט הזה בכללותו.

$$val(A) = \sum_{\substack{u\in A\\ v\notin A}} \omega_1(u) + \sum_{\substack{u\notin A\\ v\in A}} \omega_2(u,v) \quad (u,v)$$

נתונה רשת: נוסף קודקוד מזוג t, הקודקוד היחיד ש t יהיה היעד. נוסף קשתות (t,u) ל- E.

לכל $u\in V$ נגדיר קיבולת $c(u,t)=\omega_1(u)$: (u,t)

$c(u,v)=\omega_2(u,v)$: $(u,v)\in E$

הקודקודים ברשת - V'

הקשתות - E'

מה ניתן לומר על חתכים $(A,V'\setminus A)$ אחרים:

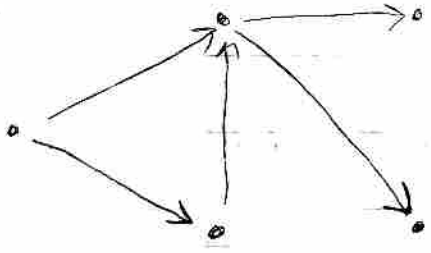
$$c(A,V'\setminus A) = \sum_{\substack{u\in A\\ v\in V\setminus A}} c(u,v) = \sum_{u\in A} c(u,t) + \sum_{\substack{u\in A\\ v\in V\setminus A}} c(u,v) =$$

$$= \sum_{u\in A} \omega_1(u) + \sum_{\substack{u\notin A\\ v\in A}} \omega_2(u,v)$$

שניתן בהחלט לחשב בזמן פולינומי. לכן נחפש חתך A כך שהחתך $(A,V'\setminus A)$ הוא קטן אינימלי. וזו בעיה שאותנו יודעים לפתור.

Project - Selection

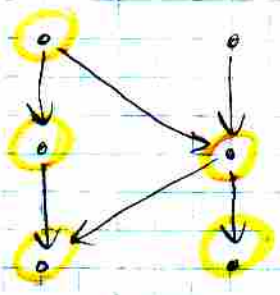
יש קהילה של פרויקטים $P=\{1,\dots,k\}$ ועבור פרויקט i יש
רשימה של שדות i - הפרויקטים שרשומים עבורו תלויים אחד
בשני (למשל אם שני פרויקטים כאלו רוצים לבנות
קווה שלישי במקום קווה שניה ...)
פרויקט i מקבל תגבור מפרויקט j אם הוא תלוי בתחום
אחד j מהדברים החיוניים עבור i.
בהינתן מבנה $G=(V,E)$ כאשר $V=P$ ו- E קיימת בדיוק (j,i) אחד
מקדים ל- i.
קהילת פרויקטים $A\subseteq P$ נתונת ביקורת אם לכל $i\in A$
$\exists j\in A$ כך ש- $(j,i)\in E$. למשל הנה קהילה נתונת ביקורת.



הרווח שנפיק מקהילת פרויקטים $A\subseteq P$ הוא $\text{profit}(A)=\sum_{i\in A} p_i$
נבנה מפתרון קהילת פרויקטים הנתונת לביקורת המקסימה את $\text{profit}(A)$
(בהינתן מפתרון) נבנה רשת $G'=(V',E',c)$ כאשר
$V'=V\cup\{s,t\}$ כך ש-
$E'=E\cup\{(s,i):p_i>0\}\cup\{(i,t):p_i<0\}$
ופונקציית הקיבול c היא
$(u,v)\in E \Rightarrow c(u,v)=\infty$
$(s,i) \Rightarrow c(s,i)=p_i$
$(i,t) \Rightarrow c(i,t)=-p_i$

נשים לב שחתך $(A'\cup V\setminus A', A')$ הוא בעל קיבולת סופית אמ"מ $A'\setminus\{s\}$
הוא קהילת פרויקטים הנתונת לביקורת.

קבוצה
ניתנת לבידוד.



קבוצות סינגלטון ניתנות לבידוד אפילו אין בהן שום צלע אחת
ויוצאת מהן.

$$c(A', V\setminus A') = \sum_{\substack{u\in A' \\ v\notin A'}} c(u,v) = \left(\begin{array}{c}\text{מיוחסים} \\ \text{סופיים}\end{array}\right) + \sum_{\substack{(u,v)\in E \\ u\in A' \\ v\notin A'}} c(u,v)$$

* $A' = V\setminus A$ ולכן $V\setminus A' = A$

טענה: אם A קבוצה שמקיימת הניתנת לבידוד אז קיימת
ויתכן ומתקיים שוויון $(A', V\setminus A')$ הוא $\sum_{i\in A} p_i = C$
כך ש $\sum_{i:p_i>0} p_i = C$.

הוכחה: יש ל-s 3 סוגי קשתות:

(i) קשתות E-ין (אם ואלה אין בהחלט)

(ii) קשתות מהסוג (s,i) עבור $0<p_i$

(iii) קשתות מהסוג (i,t) עבור $0>p_i$

$$c(A', V\setminus A') = \sum_{\substack{i\in V\setminus A \\ p_i>0}} c(s,i) + \sum_{\substack{i\in A \\ p_i<0}} c(i,t) =$$

$$= \sum_{p_i>0} c(s,i) - \sum_{\substack{i\in A \\ p_i>0}} c(s,i) + \sum_{\substack{i\in A \\ p_i<0}} c(i,t) =$$

$$= \sum_{p_i>0} p_i - \sum_{\substack{i\in A \\ p_i>0}} p_i + \sum_{\substack{i\in A \\ p_i<0}} (-p_i) =$$

$$= C - \sum_{\substack{i\in A \\ p_i>0}} p_i - \sum_{\substack{i\in A \\ p_i<0}} p_i = C - \sum_{i\in A} p_i$$

☺

- אותנו יזמים שקיים חתך מקבוצה סופית מהשת $(V\setminus A, A)$) אשר חתך המינימום כשל קבוצת סופית
אשר את כל הקבוצות שניתנות לבידוד.
- C ואז תלוי ב-A אשר מצמצם קיבולת החתך שקול למקסום $\sum_{i\in A} p_i$

(14)

לפחות אם (A', v'|A') כך שלכל A'

$$\sum_{i \in A} p_i = profit(A)$$

מתקיים ש-

(15) 29.11.06
ת. אלעזר

טרנספורם פורייה

- **מטרה:** לכפול פולינומים.

פולינום ניתן ליצג בשתי צורות - $f(x)=\sum_{i=0}^{n-1} a_i x^i$ - אפשר לייצג אותו ע"י n יחידות תאוצות שהן $(a_0,\dots,a_{n-1})$.

- **טענה:** לכל $\{(x_i,y_i)\}_{i=0}^{n-1}$ כך ש x_i שונים, קיים פולינום יחיד f כך ש $f(x_i)=y_i$ לכל i. f מדרגה לכל היותר n.
- **מסקנה.** אפשר לייצג פולינום ממעלה n ע"י n נקודות שונות. נתון $(x_0,\dots,x_{n-1})$ ונייצג את f ע"י $(f(x_0),\dots,f(x_{n-1}))$.

אנשים רגילים מייצגים פולינומים במקדמים. אנחנו נשתמש בייצוג השני. אנחנו יודעים להכפיל פולינומים בייצוג הראשון בזמן $O(n^2)$. לעומת זאת בייצוג השני אפשר להכפיל בזמן $O(n)$. אבל אם היינו יודעים בדיוק איך לעבור מייצוג אחד לשני, היינו יכולים להכפיל בזמן יעיל.

ייצוג 1	ייצוג 2	כפל
$O(n^2)$	$O(n)$	
f	f	$O(n\log n)$
g	g	$O(n\log n)$
$f\cdot g$	$f\cdot g$	$O(n\log n)$

אלגוריתם המבצע המרה בין הייצוגים נקרא - FFT:

נקודות ההתחלה הן $(1,\omega_n,\dots,\omega_n^{n-1})$ כאשר ω_n הוא שורש יחידה פרימיטיבי מסדר n. אנחנו נעבוד עם n שהוא חזקה של 2.

$\omega_n = e^{\frac{2\pi i}{n}}$ כלומר $\frac{n}{2}$

ארשום לעצמי הפרד ומשול:

$$f(x_0)=\sum_{i=0}^{n-1} a_i x_0^i = \sum_{i=0}^{n-2} a_{2i} x_0^{2i} + \sum_{i=0}^{\frac{n}{2}-1} a_{2i+1} x_0^{2i+1} = \sum_{i=0}^{\frac{n}{2}-1} a_{2i}(x_0^2)^i + x_0 \sum_{i=0}^{\frac{n}{2}-1} a_{2i+1}(x^2)^i$$
$$= f_E(x_0^2) + x_0 f_O(x_0^2)$$

והרי, $f_E(x)$ ו- $f_O(x)$ פולינומים מדרגה $\frac{n}{2}$.
ואם נציב את $(1, \omega_n, \ldots, \omega_n^{2n-1})$ נקבל שני מחזורים של $(1, \omega_{\frac{n}{2}}, \ldots, \omega_{\frac{n}{2}}^{n-1})$.

Recursive-FFT($\vec{a}$):

```
n ← length(a)
if n = 1 return a
ω_n ← e^{2πi/n}
ω ← 1
a_E ← (a_0, a_2, ..., a_{n-2})
a_O ← (a_1, a_3, ..., a_{n-1})
y_E ← Recursive-FFT(a_E)
y_O ← Recursive-FFT(a_O)
for k ← 0 to n/2 - 1
    y_k ← y_E(k) + ω y_O(k)
    y_{k+n/2} ← y_E(k) - ω y_O(k)
    ω ← ω · ω_n
return y
```

$$T(n) = \underbrace{2T\left(\frac{n}{2}\right)}_{\text{רקורסיה}} + \underbrace{O(n)}_{\text{לולאה}}$$

$$\Rightarrow T(n) = O(n \log n)$$

המטרה היא הפוך, למצוא מקדמים כנקודות $(1, \ldots, \omega_n^{n-1})$ למקדמים שמתבלבלת ע"י, אינו הבסיסי שנמצא מתלוסס את מצב ב- ω, ומטרת למקדמים ב- n. (כמו הולכת בכיתה)

(16)

$f(x) = x + 5$ $\quad g(x) = 3x + 2$

coefficients: $(5,1,0,0)$ $\quad (2,3,0,0)$

נקודות:

2. נקודות שיהיה חזקה של

$$FFT\begin{pmatrix}5\\1\\0\\0\end{pmatrix} = \begin{pmatrix}FFT\begin{pmatrix}5\\0\end{pmatrix}\\FFT\begin{pmatrix}5\\0\end{pmatrix}\end{pmatrix} + \begin{pmatrix}1\\i\\-1\\-i\end{pmatrix} \otimes \begin{pmatrix}FFT\begin{pmatrix}1\\0\end{pmatrix}\\FFT\begin{pmatrix}1\\0\end{pmatrix}\end{pmatrix} =$$

אכפלה איבר איבר

$$= \begin{pmatrix}FFT(5) + \begin{pmatrix}1\\-1\end{pmatrix} \otimes \begin{pmatrix}FFT(0)\\FFT(0)\end{pmatrix}\\FFT(5) + \begin{pmatrix}1\\-1\end{pmatrix} \otimes \begin{pmatrix}FFT(0)\\FFT(0)\end{pmatrix}\end{pmatrix} + \begin{pmatrix}1\\i\\-1\\-i\end{pmatrix} \otimes \begin{pmatrix}FFT(1) + \begin{pmatrix}1\\-1\end{pmatrix} \otimes \begin{pmatrix}FFT(0)\\FFT(0)\end{pmatrix}\\FFT(1) + \begin{pmatrix}1\\-1\end{pmatrix} \otimes \begin{pmatrix}FFT(0)\\FFT(0)\end{pmatrix}\end{pmatrix} =$$

$$= \begin{pmatrix}5\\5\\5\\5\end{pmatrix} + \begin{pmatrix}1\\i\\-1\\-i\end{pmatrix} \otimes \begin{pmatrix}1\\1\\1\\1\end{pmatrix} = \begin{pmatrix}6\\5+i\\4\\5-i\end{pmatrix}$$

באותה דרך

$$FFT\begin{pmatrix}2\\3\\0\\0\end{pmatrix} = \begin{pmatrix}5\\2+3i\\-1\\2-3i\end{pmatrix}$$

$\Rightarrow$ המכפלה של $f \cdot g$ היא

$$\begin{pmatrix}6\\5+i\\4\\5-i\end{pmatrix} \otimes \begin{pmatrix}5\\2+3i\\-1\\2-3i\end{pmatrix} = \begin{pmatrix}30\\7+17i\\-4\\7-17i\end{pmatrix}$$

ועכשיו נחזיר את המכפלה ע"י אלגוריתם ה- IFFT:

$$FFT^{-1}\begin{pmatrix}30\\7+17i\\-4\\7-17i\end{pmatrix} = \begin{pmatrix}FFT^{-1}\begin{pmatrix}30\\-4\end{pmatrix}\\FFT^{-1}\begin{pmatrix}30\\-4\end{pmatrix}\end{pmatrix} + \begin{pmatrix}1\\-i\\-1\\i\end{pmatrix} \otimes \begin{pmatrix}FFT^{-1}\begin{pmatrix}7+17i\\7-17i\end{pmatrix}\\FFT^{-1}\begin{pmatrix}7+17i\\7-17i\end{pmatrix}\end{pmatrix} =$$

$$= \begin{pmatrix}26\\34\\26\\34\end{pmatrix} + \begin{pmatrix}1\\-i\\-1\\i\end{pmatrix} \otimes \begin{pmatrix}14\\34i\\14\\34i\end{pmatrix} = \begin{pmatrix}40\\68\\12\\0\end{pmatrix} \xrightarrow{\cdot \frac{1}{4}} \begin{pmatrix}10\\17\\3\\0\end{pmatrix}$$

$$\hookrightarrow FFT^{-1}\begin{pmatrix}a\\b\end{pmatrix} = \begin{pmatrix}a\\a\end{pmatrix} + \begin{pmatrix}1\\-1\end{pmatrix} \otimes \begin{pmatrix}b\\b\end{pmatrix} = \begin{pmatrix}a+b\\a-b\end{pmatrix}$$

הכיוון והפוליטרים הקיבוץ - במקרה הקלאסי הקונוטיבים הכיוון
שמריץ עשאור .

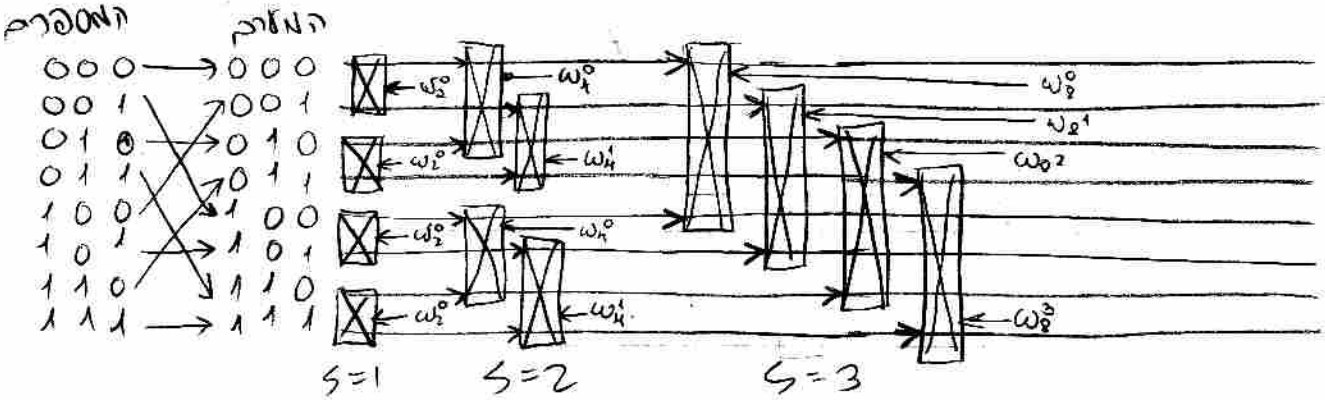
זה שקורה באלגוריתם נקרא פעולת פרפר

ב- FFT עושים הרבה פעולות פרפר רצף ומדמים אותן יחד.

איך נוכל לשפר? נסתכל איפה הבזבזים שעשאים ביחד את הסך.
נוכל להסתכל על הסתכלות בינארי. לאשר

אנחנו רואים שהבזבזים הראשונים שוקלים ביחד - ספרתם הראשונה להיה
אחר השפרה השנייה אפשר להיה וכן הלאה.
אז נעיין את המספרים ב- bit reversal

ניתן לראות שהסדר של האיברים $a_0, \ldots, a_7$ בתחילת האלגוריתם הוא סדר ה-bit-reversal של המספרים המתאימים לאיברים בסדר הרגיל:



כפי שאפשר לראות האלגוריתם הזה טוב כי אפשר לחלק את הקלט ל-2 בצורה שווה ואז לחשב את התוצאה מהקלט הזה, אם יש 8 איברים אז אפשר לחלק אותם מהקלט ל-2 ואחד שזה $\log$ פעולות ול-2 שיש איברים אחד שם השני.

Iterative-FFT (a)

```
Bit-reverse-copy (ā)
n ← length a
for s ← 1 to log n          // כל פעם הרמה
    m ← 2^s
    ω_m ← e^(2πi/m)
    for k ← 0 : m : (n-1)    // מקפיצות של m (כמו ב-MATLAB)
        ω ← 1
        for j ← 0 to m/2 - 1
            t ← ω · a(k+j + m/2)
            u ← a(k+j)
            a(k+j) ← u + t
            a(k+j + m/2) ← u - t
            ω ← ω · ω_m
```

את נשאר הבעיה שלנו?

$$\sum_{s=1}^{\log n} \left(\frac{n-1}{2^s}\right) \frac{n}{2} \cdot O(1) = O(1) \underbrace{\sum_{s=1}^{\log n} \frac{n-1}{2}}_{n \log \frac{n-1}{2}} = O(n \log n)$$

טוב, לא מספיק שיצא אותו דבר כי עשינו את אותו הדבר שהרקורסיה עושה רק בצורה אוטומטית.

נשארו נחשד: f, g פולינומים מדרגה n.
נשמור לרפול אותם בדיוק לה שהיו בקלט.
אולי אם התעקלנו והשבנו רק ב-n נקודות אנחנו
לא יכולים לשחזר פולינום. זה פולינום בכלל מלבד זה $f \cdot g$
של הנקודות שכן נתנו!

נניח שקיבלנו $h(x)$ אותו שהשבנו על $1, \omega_n, \dots, \omega_n^{n-1}$

$$h(x) = f(x)g(x) \quad x = 1, \omega_n, \dots, \omega_n^{n-1}$$

$$\Rightarrow h(x) - f(x)g(x) = \underbrace{(x-1)(x-\omega_n)\dots(x-\omega_n^{n-1})}_{x^n - 1} \cdot r(x)$$

$$(x^n-1)\sum_{i=0}^{l} r_i x^i = \sum_{i=0}^{l} r_i x^{i+n} - \sum_{i=0}^{l} r_i x^i =$$

אנחנו יכולים לשחזר שאלו בדיוק לחלקה. h ו- fg הם
שווי מקדמים עבור המקדמים ה-j רשר $n > j > l$.

שקילות מודולרית

$x \bmod y$ = השארית של x בחלוקה ב-y

הגדרה: $a,b \in \mathbb{Z}$ יקראו שקולים מודולו $m \in \mathbb{N}, m>0$ אם $a = b + km$ עבור $k \in \mathbb{Z}$.

סימון: $a \equiv_m b$ או $a \equiv b \pmod m$

ניתן להגדיר פעולות חיבור וכפל מודולו m ע"י:

$$a +_m b = (a+b) \bmod m$$

$$a \cdot_m b = (a \cdot b) \bmod m$$

נשים ♡ שאם $a \equiv_m c$! $b \equiv_m d$ אז $a+_m b = c +_m d$, $a \cdot_m b = c \cdot_m d$

gcd - האלגוריתם היעיל ביותר.

הגדרה: אם $a,b \in \mathbb{N}$, לא שניהם אפס, אז

$$gcd(a,b) = \max\{d \geq 1 : d|a \wedge d|b\}$$

מדוע שקבלנו כי יש מקסימום כי 1 מקיים זה והקבוצה סופית. לעומת זאת קבוצת כל המספרים השלמים לא תכיל מקסימום.

טענה: אם $a,b \in \mathbb{N}$, לא שניהם אפס אז

$$gcd(a,b) = \min\{x a + y b > 0 : x,y \in \mathbb{Z}\}$$

דוגמה: $gcd(7,5) = 1 = 3 \cdot 7 + (-4) \cdot 5$

הוכחה: נניח $sx + ty = 5$ ניצוג את הביטויים הלינאריים השלמים ע"י נקודות במישור.

ראשית, נסמן ב-s את המרחק המשותף של a,b.

$a \bmod s = a - \lfloor \frac{a}{s} \rfloor s = a - \lfloor \frac{a}{b} \rfloor b = (sx+ty) - \lfloor \frac{a}{b} \rfloor (qx+ty) = (1 - \lfloor \frac{a}{b} \rfloor q)x$

$\Rightarrow a \bmod s$ בעל ערכי משמרים של a,b, אבל $a \bmod s < s$ בעוד ש-s הוא המינימום של הקבוצה). $\Leftarrow a \bmod s = 0$

לכן $s|a$ וגם $s|b$.

נשאלת שאלה כיצד נמצא את s ביעילות? נשים לב שאם $a \bmod s = 0$ אז s הוא המחלק המשותף הגדול ביותר של a,b.

נשים לב ש- $a \bmod b = a - \lfloor \frac{a}{b} \rfloor b$ ולכן $a \bmod b$ הוא צירוף לינארי שלם של a,b.

נשים לב ש- $a \bmod b < b$ ולכן $a \bmod b$ הוא מספר קטן יותר מ-a ומ-b.

נשים לב ש- $gcd(a,b) = gcd(b, a \bmod b)$ כי כל מחלק משותף של a,b הוא מחלק משותף של $b, a \bmod b$ וכל מחלק משותף של $b, a \bmod b$ הוא מחלק משותף של a,b.

אלגוריתם אוקלידס:

```
Euclid(a,b):
    if b = 0 then return a
    else return Euclid(b, a mod b)
```

משפט: אלגוריתם אוקלידס מחשב את $gcd(a,b)$ בזמן $O(\log a + \log b)$.

הוכחה: נוכיח באינדוקציה על $a+b$ שמספר הקריאות הרקורסיביות הוא $O(\log a + \log b)$.

בסיס: $a+b \leq 1$ ברור.

צעד: נניח נכונות עבור $a+b < n$ ונוכיח עבור $a+b = n$.

אם $b \geq \frac{a}{2}$ אז $a \bmod b = a - b < \frac{a}{2}$.

אם $b < \frac{a}{2}$ אז $a \bmod b < b < \frac{a}{2}$.

בכל מקרה, $a \bmod b < \frac{a}{2}$.

לכן $(a \bmod b) + b < \frac{a}{2} + b \leq a$.

לכן $(a \bmod b) + b < a$.

לכן $(a \bmod b) + b < n$.

לכן $(a \bmod b) + b$ מקיים את הנחת האינדוקציה.

לכן מספר הקריאות הרקורסיביות הוא $O(\log a + \log b)$.

משפט: אלגוריתם אוקלידס מחשב את $gcd(a,b)$ בזמן $O(\log a + \log b)$.

הוכחה: נוכיח באינדוקציה על $a+b$ שמספר הקריאות הרקורסיביות הוא $O(\log a + \log b)$.

בסיס: $a+b \leq 1$ ברור.

צעד: נניח נכונות עבור $a+b < n$ ונוכיח עבור $a+b = n$.

אם $b \geq \frac{a}{2}$ אז $a \bmod b = a - b < \frac{a}{2}$.

אם $b < \frac{a}{2}$ אז $a \bmod b < b < \frac{a}{2}$.

בכל מקרה, $a \bmod b < \frac{a}{2}$.

לכן $(a \bmod b) + b < \frac{a}{2} + b \leq a$.

לכן $(a \bmod b) + b < a$.

לכן $(a \bmod b) + b < n$.

לכן $(a \bmod b) + b$ מקיים את הנחת האינדוקציה.

לכן מספר הקריאות הרקורסיביות הוא $O(\log a + \log b)$.

כלומר המספרים הם אותו שארית מחלוקה. ■

עתה אפשר להוכיח את נכונות האלגוריתם (שמחזיר את המחלק המשותף הגדול ביותר) באינדוקציה על הארגומנט השני. ⇐ הנה אלגוריתם אוקלידס הרקורסיבי:

```
Euclid (a,b)
if b=0
    return a
else
    return Euclid (b, a mod b)
```

* הערה: הנכונות של האלגוריתם נובעת מהטענה - שם הוכח שאם הפעלת האלגוריתם על (a,b) מחזירה d, אזי d הוא המחלק המשותף הגדול ביותר של a ו-b.

$\gcd(a,b) = \gcd(b, a \bmod b)$. ☺

שוב נשים לב שהמספרים אכן הולכים וקטנים, כיון ש $a \bmod b < b$. לכן לפי הטענה הקודמת $\gcd(a \bmod b, b) = \gcd(a,b)$ ⇐ הקבוצה של המחלקים המשותפים של a ו-b זהה לקבוצה של המחלקים המשותפים של $a \bmod b$ ו-b.

אם a מתחלק ב-b אזי b הוא המחלק המשותף הגדול ביותר של a ו-b, כלומר $\gcd(a,b) = b$.

אחרת, $a = qb + r$ כאשר $0 < r < b$. ⇐ $r = a \bmod b$.

$\gcd(a,b) = \gcd(b, a \bmod b)$ ⇐ $\gcd(a,b) = \gcd(b,r)$

הוכחה: יהיו $x, y \in \mathbb{Z}$ כך ש- $ax + by$ הוא הצירוף הלינארי הקטן ביותר החיובי של a ו-b. נראה ש- $a \bmod b$ הוא צירוף לינארי של a ו-b עם אותם מקדמים x ו-y.

$$a \bmod b = a - \left\lfloor \frac{a}{b} \right\rfloor b = a - \left\lfloor \frac{a}{b} \right\rfloor (ax + by) =$$

$$= a(1 - \left\lfloor \frac{a}{b} \right\rfloor x) + b(-\left\lfloor \frac{a}{b} \right\rfloor y)$$

כלומר $a \bmod b$ הוא צירוף לינארי של a ו-b עם אותם מקדמים x ו-y כמו $ax + by$.

⇐ הקבוצה של הצירופים הלינאריים של a ו-b זהה לקבוצה של הצירופים הלינאריים של b ו-$a \bmod b$.

שוב, מכיון שהמספרים אכן הולכים וקטנים, נגיע בסופו של דבר לזוג שבו השני מחלק את הראשון. ⇐ הנה האלגוריתם הרקורסיבי הראשון של אוקלידס (שמחזיר את המחלק המשותף הגדול ביותר) כפי שהוא מופיע בספרים הרבה יותר מאוחר משהוא נכתב.

אלגוריתם של אוקלידס

טענה: יהי $a, b \in \mathbb{N}$ כך ש- $a > b$. אזי

$$\gcd(a,b) = \gcd(b, a \bmod b)$$

הוכחה: יהיו $x, y \in \mathbb{Z}$ כך ש- $ax + by$ הוא הצירוף הלינארי החיובי הקטן ביותר של a ו-b. נראה ש- $a \bmod b$ הוא צירוף לינארי של a ו-b עם אותם מקדמים x ו-y.

$$a \bmod b = a - \left\lfloor \frac{a}{b} \right\rfloor b = a - \left\lfloor \frac{a}{b} \right\rfloor (ax + by) =$$

$$= a(1 - \left\lfloor \frac{a}{b} \right\rfloor x) + b(-\left\lfloor \frac{a}{b} \right\rfloor y)$$

כלומר $a \bmod b$ הוא צירוף לינארי של a ו-b עם אותם מקדמים x ו-y כמו $ax + by$.

⇐ הקבוצה של הצירופים הלינאריים של a ו-b זהה לקבוצה של הצירופים הלינאריים של b ו-$a \bmod b$.

שוב, מכיון שהמספרים אכן הולכים וקטנים, נגיע בסופו של דבר לזוג שבו השני מחלק את הראשון. ⇐ הנה האלגוריתם הרקורסיבי הראשון של אוקלידס (שמחזיר את המחלק המשותף הגדול ביותר) כפי שהוא מופיע בספרים הרבה יותר מאוחר משהוא נכתב.

```
Euclid (a,b)
if b=0
    return a
else
    return Euclid (b, a mod b)
```

* הערה: הנכונות של האלגוריתם נובעת מהטענה - שם הוכח שאם הפעלת האלגוריתם על (a,b) מחזירה d, אזי d הוא המחלק המשותף הגדול ביותר של a ו-b.

$\gcd(a,b) = \gcd(b, a \bmod b)$. ☺

שוב נשים לב שהמספרים אכן הולכים וקטנים, כיון ש $a \bmod b < b$. לכן לפי הטענה הקודמת $\gcd(a \bmod b, b) = \gcd(a,b)$ ⇐ הקבוצה של המחלקים המשותפים של a ו-b זהה לקבוצה של המחלקים המשותפים של $a \bmod b$ ו-b.

אם a מתחלק ב-b אזי b הוא המחלק המשותף הגדול ביותר של a ו-b, כלומר $\gcd(a,b) = b$.

אחרת, $a = qb + r$ כאשר $0 < r < b$. ⇐ $r = a \bmod b$.

$\gcd(a,b) = \gcd(b, a \bmod b)$ ⇐ $\gcd(a,b) = \gcd(b,r)$

הוכחה: יהיו $x, y \in \mathbb{Z}$ כך ש- $ax + by$ הוא הצירוף הלינארי החיובי הקטן ביותר של a ו-b. נראה ש- $a \bmod b$ הוא צירוף לינארי של a ו-b עם אותם מקדמים x ו-y.

$$a \bmod b = a - \left\lfloor \frac{a}{b} \right\rfloor b = a - \left\lfloor \frac{a}{b} \right\rfloor (ax + by) =$$

$$= a(1 - \left\lfloor \frac{a}{b} \right\rfloor x) + b(-\left\lfloor \frac{a}{b} \right\rfloor y)$$

כלומר $a \bmod b$ הוא צירוף לינארי של a ו-b עם אותם מקדמים x ו-y כמו $ax + by$.

⇐ הקבוצה של הצירופים הלינאריים של a ו-b זהה לקבוצה של הצירופים הלינאריים של b ו-$a \bmod b$.

שוב, מכיון שהמספרים אכן הולכים וקטנים, נגיע בסופו של דבר לזוג שבו השני מחלק את הראשון. ⇐ הנה האלגוריתם הרקורסיבי הראשון של אוקלידס (שמחזיר את המחלק המשותף הגדול ביותר) כפי שהוא מופיע בספרים הרבה יותר מאוחר משהוא נכתב.

```
Euclid (a,b)
if b=0
    return a
else
    return Euclid (b, a mod b)
```

* הערה: הנכונות של האלגוריתם נובעת מהטענה - שם הוכח שאם הפעלת האלגוריתם על (a,b) מחזירה d, אזי d הוא המחלק המשותף הגדול ביותר של a ו-b.

$\gcd(a,b) = \gcd(b, a \bmod b)$. ☺

שוב נשים לב שהמספרים אכן הולכים וקטנים, כיון ש $a \bmod b < b$. לכן לפי הטענה הקודמת $\gcd(a \bmod b, b) = \gcd(a,b)$ ⇐ הקבוצה של המחלקים המשותפים של a ו-b זהה לקבוצה של המחלקים המשותפים של $a \bmod b$ ו-b.

אם a מתחלק ב-b אזי b הוא המחלק המשותף הגדול ביותר של a ו-b, כלומר $\gcd(a,b) = b$.

אחרת, $a = qb + r$ כאשר $0 < r < b$. ⇐ $r = a \bmod b$.

$\gcd(a,b) = \gcd(b, a \bmod b)$ ⇐ $\gcd(a,b) = \gcd(b,r)$

הוכחה: יהיו $x, y \in \mathbb{Z}$ כך ש- $ax + by$ הוא הצירוף הלינארי החיובי הקטן ביותר של a ו-b. נראה ש- $a \bmod b$ הוא צירוף לינארי של a ו-b עם אותם מקדמים x ו-y.

$$a \bmod b = a - \left\lfloor \frac{a}{b} \right\rfloor b = a - \left\lfloor \frac{a}{b} \right\rfloor (ax + by) =$$

$$= a(1 - \left\lfloor \frac{a}{b} \right\rfloor x) + b(-\left\lfloor \frac{a}{b} \right\rfloor y)$$

כלומר $a \bmod b$ הוא צירוף לינארי של a ו-b עם אותם מקדמים x ו-y כמו $ax + by$.

⇐ הקבוצה של הצירופים הלינאריים של a ו-b זהה לקבוצה של הצירופים הלינאריים של b ו-$a \bmod b$.

שוב, מכיון שהמספרים אכן הולכים וקטנים, נגיע בסופו של דבר לזוג שבו השני מחלק את הראשון. ⇐ הנה האלגוריתם הרקורסיבי הראשון של אוקלידס (שמחזיר את המחלק המשותף הגדול ביותר) כפי שהוא מופיע בספרים הרבה יותר מאוחר משהוא נכתב.

```
Euclid (a,b)
if b=0
    return a
else
    return Euclid (b, a mod b)
```

* הערה: הנכונות של האלגוריתם נובעת מהטענה - שם הוכח שאם הפעלת האלגוריתם על (a,b) מחזירה d, אזי d הוא המחלק המשותף הגדול ביותר של a ו-b.

טענה: אם $a > b \geq 1$ עבור אלגוריתם Euclid מתקיים $a \geq F_{k+2}$, $b \geq F_{k+1}$ לכל $k \geq 1$ (כאשר F_k הוא איבר k בסדרת פיבונאצ'י).

מסקנה: עבור $a > b \geq 1$ כלשהם, Euclid מבצע פחות מ- k קריאות רקורסיביות.

מכיוון ש $F_k \sim c\left(\frac{\sqrt{5}+1}{2}\right)^k$ עבור c כלשהו אזי סיבוכיות הזמן היא $O(\log b)$

הוכחת הטענה: באינדוקציה על k:

$k = 1$ $\quad b \geq 1 = F_2$, $\quad a \geq 2 = F_3$

עבור k כלשהו נניח עבור $k-1$.

נניח כי Euclid(a,b) מבצע k קריאות. אזי Euclid$(b, a \bmod b)$ מבצע $k-1$ קריאות. מהנחת האינדוקציה $b \geq F_{k+1}$,

$$a \geq b + (a - \lfloor \tfrac{a}{b} \rfloor b) = b + (a \bmod b) \geq F_{k+1} + F_k = F_{k+2} \quad \Leftarrow \quad a \bmod b \geq F_k$$

($\lfloor \frac{a}{b} \rfloor \geq 1$)

אלגוריתם אוקלידס המורחב

נרצה למצוא x, y כך ש- $gcd(a,b) = xa + yb$.

$$gcd(b, a \bmod b) = x_n b + y_n (a \bmod b) = x_n b + y_n \left(a - \lfloor \tfrac{a}{b} \rfloor b\right) =$$

$$= \underbrace{y_n}_{x_{n+1}} a + \underbrace{\left(x_n - \lfloor \tfrac{a}{b} \rfloor y_n\right)}_{y_{n+1}} b$$

Extended-Euclid(a,b)

```
if b = 0                          gcd  x  y
    return (a, 1, 0)
else
    (d, x, y) ← Extended-Euclid(b, a mod b)
    return (d, y, x - ⌊a/b⌋ y)
```

תזכורת והתחלות

הגדרה: יהי $(G,\circ)$ כאשר $G\neq\emptyset$ ו- $\circ: G\times G\to G$.

נקרא חבורה אם מתקיימים הבאים: לכל $a,b,c\in G$

1. אסוציאטיביות: $a\circ(b\circ c)=(a\circ b)\circ c$
2. קיים איבר (ניטרלי) $e\in G$ כך שלכל $a\in G$ מתקיים
$$a\circ e = e\circ a = a$$
3. לכל איבר $a\in G$ קיים איבר $b\in G$ כך ש-
$$a\circ b = b\circ a = e$$

נקרא הפכי ל- a ומסומן a^{-1}.

הגדרה: חבורה $(G,\circ)$ תיקרא קומוטטיבית (או אבלית) אם לכל $a,b\in G$ $a\circ b=b\circ a$

דוגמאות:

1. אם F שדה אז $(F,+_F)$ חבורה קומוטטיבית
2. אם F שדה אז $(F^*,\cdot_F)=(F\setminus\{0_F\},\cdot_F)$ חבורה קומוטטיבית.
3. $(\mathbb{Z},+)$ הדרכה: להזכר או מקרה פרטי של (1), בדיקה!
4. בהינתן m טבעי קבוע $\mathbb{Z}_m=\{0,1,\dots,m-1\}$.
$\mathbb{Z}_m$ עם חיבור מודולו m חבורה קומוטטיבית.
5. $\mathbb{Z}_m^*=\{1\le i\le m-1 : \gcd(i,m)=1\}$ עם כפל מודולו m.

תכף נוכיח שכל חבורה אבלית נקבל את המשפט היסודי של האריתמטיקה: לכל $n>1$ קיימת הצגה יחודית $n=p_1^{l_1}\cdot\dots\cdot p_r^{l_r}$ כך ש $p_1<\dots<p_r$ ראשוניים ו $0<l_i$.

$\Leftarrow$ אם $p\mid ab$ אזי a או b מתחלק אשורית ב $d>1$ כך קיים p ראשוני כך ש- $p\mid d$ אלמן $p\mid a$, $p\mid b$. ראוי, שאם p ראשוני כך ש $p\mid ab$ אז $p\mid a$ או $p\mid b$.

מתשו נראה ש- $\mathbb{Z}_m^*$ עם כפל מודולו m חבורה.

סימון: אם $a,b\in\mathbb{Z}_m^*$ אז $\gcd(a,m)=1=\gcd(b,m)$

אם כן שלזה $\gcd(ab,m)>1$ אז קיים מחלק אשורית של ab ו- m

שימוש נוסף: אם $p \mid ab$ אז $p \mid a$ או $p \mid b$.

אם p אינו מחלק אף אחד מגורמי a אז $(a,m)=1$ וגם $(b,m)=1$ ולכן $(ab,m)=1$.

אוסף האיברים ההפיכים - הוכחנו שאוסף האיברים ההפיכים ב- $\mathbb{Z}_m$ הוא $\mathbb{Z}_m^* \subseteq \mathbb{Z}_m - \{0\}$.

$\Leftarrow$ מוגדר היטב. אוסף האיברים ב- $\mathbb{Z}_m^*$.

איבר יחידה - כאמור $1 \in \mathbb{Z}_m^*$ והוא איבר ניטרלי ביחס לכפל מודולו m.

קיום הפכי - יהי $a \in \mathbb{Z}_m^*$ $\Leftarrow$ $gcd(a,m)=1$ $\Leftarrow$ קיימים $x,y \in \mathbb{Z}$ כך ש- $xa+ym=1$ (*). נקרא למודולו m ונקבל $xa+ym \equiv_m 1$ $\Leftarrow$ $xa \equiv_m 1$ $\Leftarrow$ $x \bmod m$ הפכי ל- a. נשים ♡ ש- $x \bmod m \in \mathbb{Z}_m^*$ כי (*) גורר שעבור x של m השארית ל- 1 מתקיים $gcd(x,m)=1$ (כי x,m הם הגורמים המשותפים היחידים של x ו- m).

$\Leftarrow$ $gcd(m, x \bmod m)=1$ $\Leftarrow$ $x \bmod m \in \mathbb{Z}_m^*$. ☺

כל ההוכחה הקונסטרוקטיבית לעיל טובה אם x אותנו יכולים לאתר מלירתם אלגוריתם אוקלידס המורחב. נקרא $Extended\text{-}Euclid(m,a)$ ונקבל (d,x,y). כאשר $x \bmod m = a^{-1}$.

סדר איבר בחבורה: יהי G סופית. אז לכל $a \in G$ יש בחבורה $1, a, a^2, a^3, \ldots, a^n, \ldots$

G סופית לכן קיימים $n<m$ כך ש- $a^m = a^n$.

נכפיל את שני האגפים ב- $(a^n)^{-1}$ ונקבל $a^{m-n} = a^m (a^n)^{-1} = 1$

אפשר להגדיר את חזקי החלקות.

נשקיר $ord(a) = \min\{n \geq 1 : a^n = 1\}$

הגדרנו חבורה $(G, \circ)$ -

G - קבוצה ! - $\circ$ פעולה בין שכנותיה ושמקיים:

- אסוציאטיביות
- איבר יחידה (e נקרא גם 1 -)
- קיום הופכי

הגדרה: תהי G חבורה. אם קבוצה $H \subseteq G$ ונקראת תת חבורה של G אם H חבורה ביחס לפעולה המושרית מ- G.

כדאי שקול לומר שמתקיים הדברים הבאים $H \neq \phi$ $H \subseteq G$ תת חבורה של G אם"מ H סגורה לפעולה • והופכי.

דוגמאות:

1) $\{1\}$ תת חבורה של G

2) אם H_1, H_2 תת"ח של G אז $H_1 \cap H_2$ תת"ח של G

3) אם $a \in G$ נגדיר את תת החבורה הנוצרת ע"י a:

$$(*) \quad 1, a, a^2, a^3, \ldots$$

a^m לכל m טבעי $a^{-m} = \underbrace{a^{-1} \cdot \ldots \cdot a^{-1}}_{m \text{ פעמים}}$ $\underbrace{a \cdot a \cdot \ldots \cdot a}_{m \text{ פעמים}}$

אפשר להוכיח בקלות באינדוקציה ש- $(a^n)^m = a^{nm}$; $a^m \cdot a^n = a^{m+n}$

אם G סופית אז קיימים $0 \leq m_0 < n_0$ כך ש- $a^{n_0} = a^{m_0}$

נכפיל ב- a^{-m_0} ונקבל $a^{n_0 - m_0} = a^{n_0} \cdot a^{-m_0} = a^{m_0} \cdot a^{-m_0} = 1$

לכן אם נתבונן ב-(*) אז העיקום בו היינו מגיעה שם ל-1:

$1, a, a^2, \ldots, a^{k-1}$ נקבל k איברים שונים ואז (*) מתנהגים כמו מחזור k. נגדיר $\text{ord}\, a = \min\{k \geq 1 : a^k = 1\}$

נשים ♡ ש- $\{1, a, \ldots, a^{\text{ord}\, a - 1}\}$ תת"ח של G:

סגירות לכפל: $a^n \cdot a^m = a^{m+n} = a^{(m+n) \bmod \text{ord}\, a}$

סגירות להופכי: עבור $n \geq 1$ $a^n \cdot a^{\text{ord}\, a - n} = a^{\text{ord}\, a} = 1$

משפט לגראנז': אם G חבורה סופית ו- H תת חבורה של G. אז $|H| \mid |G|$

הוכחה: נסתכל ב- $B = \{aH : a \in G\}$ ($aH = \{ax : x \in H\}$)

נוכיח שכל איבר $H \in B$ הוא אותו גודל של תת הקבוצה H. תזכורת: aH ($a = 1$ ניקח $eH = H$).

נטען ש- B אוסף של קבוצות זרות. נניח ש- $aH \cap bH \neq \emptyset$. אז קיימים $h_1, h_2 \in H$ כך ש- $ah_1 = bh_2$

$$\Leftarrow \quad h_1 h_2^{-1} = a^{-1}b \quad \Leftarrow \quad h_1 = a^{-1}ah_1 = a^{-1}bh_2$$

$\Leftarrow$ $a^{-1}b \in H$. יהי $ah \in aH$, אז

$$ah = a(a^{-1}b)(a^{-1}b)^{-1}h = b\underbrace{(a^{-1}b)^{-1}h}_{\in H} \in bH$$

$\Leftarrow$ $aH \subseteq bH$. באופן סימטרי $aH \supseteq bH$

$\Leftarrow$ $aH = bH$. $\Leftarrow$ הקבוצות אכן זרות.

ניקח $a, b \in G$ כלשהם $|aH| = |bH|$. נגדיר ה"ע $\varphi: aH \to bH$ ע"י $\varphi(x) = ba^{-1}x$.

נראה ש- φ חח"ע ועל ולכן נקבל $|aH| = |bH|$.

φ חח"ע כי אם $\varphi(x_1) = \varphi(x_2)$ אז $ba^{-1}x_1 = ba^{-1}x_2$.

נכפול ב- $(ba^{-1})^{-1}$ ונקבל $x_1 = x_2$.

φ על כי אם $bh \in bH$ אז $\varphi(ah) = ba^{-1}ah = bh$.

$\Leftarrow$ B אוסף של קבוצות זרות השוות גודל. וכן $G = \bigcup_{A \in B} A$. לכן $|G| = |B| \cdot |H|$.

$\Leftarrow$ $|H| \mid |G|$ ☺

מסקנה: המשפט הקטן של פרמה:

אם p ראשוני ו- $1 \leq a \leq p-1$ אז $a^{p-1} \equiv_p 1$

(נניח שאם a כך ש- $a \nmid p$)

הוכחה: ניכנס מתחבורה $\mathbb{Z}_p^* = \{i \mid 1 \leq i \leq p-1 : gcd(i,p) = 1\}$

עם כפל מודולו p.

מכיוון ש- p ראשוני: $\mathbb{Z}_p^* = \{1, \ldots, p-1\}$. ויהי $a \in \mathbb{Z}_p^*$.

אנו יודעים ש- $\text{ord}(a)$ = אבל התכונה הנדרשת ע"י a.
ארגון, מאחר ש $|\mathbb{Z}_p^*| = p-1$ מתקיים $\text{ord}(a) \mid |\mathbb{Z}_p^*|$

$$\Rightarrow a^{|\mathbb{Z}_p^*|} = a^{\text{ord}(a)\cdot k} = 1$$

$$\Rightarrow a^{p-1} = a^{|\mathbb{Z}_p^*|} \equiv_p a^{\text{ord}(a)\cdot k} \equiv_p 1$$

(11)

משפט:

אם $m \in \mathbb{N}$ $m > 0$ ו- $\gcd(a,m) = 1$ אזי $a \in \mathbb{Z}$

$$a^{|\mathbb{Z}_m^*|} \equiv_m 1$$

מסקנה: $|\mathbb{Z}_m^*|$ הוא $\phi(m)$ - פונקציית אוילר.

משפט השאריות הסיני - Sun Tzu Suan Ching

הבעיה: יש לנו מספר צבאים.
אם נספור אותם בקבוצות של 2 אחדים
אם בקבוצות של 3 נשארו 1
אם בקבוצות של 2 נשארו 1.
כמה צבאים יש לנו?

אנחנו מחפשים $a \in \mathbb{Z}$ כך ש-

1) $a \bmod 3 = 2$
2) $a \bmod 5 = 3$
3) $a \bmod 7 = 2$

לדוגמא: $128 = 35 + 3\cdot 21 + 2\cdot 15$ ולכן $a = 35 + 3\cdot 21 + 2\cdot 15$

35 מתחלק ב-5 ו-7 אבל לא ב-3
21 מתחלק ב-3 ו-7 אבל לא ב-5
15 מתחלק ב-3 ו-5 אבל לא ב-7

מכיוון שכל הוספה או החסרה של $105 = 3\cdot 5\cdot 7$ מחזירה לנו אחד השארית.
$\Leftarrow$ כל $128 + 105k$ מהווה פתרון לבעיה.

משפט השאריות הסיני: יהי $n = n_1 \cdot n_2 \cdots n_k$ כך שלכל $i \neq j$ מתקיים $\gcd(n_i, n_j) = 1$. אז ההעתקה הבאה חח"ע ועל.

$$f: \mathbb{Z}_n \longrightarrow \mathbb{Z}_{n_1} \times \mathbb{Z}_{n_2} \times \cdots \times \mathbb{Z}_{n_k}$$
$$f(a) = (a \bmod n_1,\ a \bmod a_2,\ \ldots,\ a \bmod n_k)$$

הוכחה: נבנה את ההעתקה ההפוכה.

נניח שניתן $(a_1, \ldots, a_k) \in \mathbb{Z}_{n_1} \times \cdots \times \mathbb{Z}_{n_k}$. נסמן

$$m_i = \frac{n}{n_i} = n_1 \cdots n_{i-1} n_{i+1} \cdots n_k, \quad c_i = m_i \cdot \underbrace{(m_i^{-1} \bmod \mathbb{Z}_{n_i}^*)}_{\text{הפכי של } m_i \text{ ב}\mathbb{Z}_{n_i}^*}$$

(נשים ♡ ש- $m_i \in \mathbb{Z}_{n_i}^*$ כי m_i זר ל- n_i לכל i בפולשת.

לכאורה כי $m_i = n_k \cdots \hat{n_i} \cdots n_1$ לכל i בפולשת.

כמו כן $c_i \bmod n_j = 0$ מתקיים $j \neq i$

$$c_i \bmod n_i = (m_i \bmod n_i)(m_i^{-1} \bmod n_i) = 1$$

נקח את $c_i' = c_i \bmod n \in \mathbb{Z}_n$ ואז

$$f\left(\sum_{i=1}^{k} a_i c_i'\right) = (a_1 c_1 \bmod n_1, \ldots, a_k c_k \bmod n_k)$$
$$= (a_1, \ldots, a_k)$$

נשים ♡ ש- $|\mathbb{Z}_n| = n = n_1 \cdots n_k = |\mathbb{Z}_{n_1} \times \cdots \times \mathbb{Z}_{n_k}|$.

ולכן f פונקציה חח"ע בין שתי קבוצות סופיות שוות גודל ולכן היא תח"ע. (!!)

מסקנה: $\varphi(n) = n \prod_{\substack{p \mid n \\ \text{ראשוני}}} \left(\frac{1}{p} - 1\right)$

הוכחה: נשים ♡ ש- $\gcd(m, n) = 1$ אמ"מ אין מספר ראשוני p שמחלק את m ואת n.

כעת אם הפירוק לגורמים ראשוניים של n הוא $n = p_1^{e_1} \cdots p_r^{e_r}$ כך ש- $e_i > 0$ ו- p_i ראשוני, אז לכן $\varphi(n)$ סופר את מספר האיברים x ב- $\mathbb{Z}_n$ כך ש- $x \bmod p_i \neq 0$ לכל i.

כל מספר ב- $\mathbb{Z}_n$ שלא מתחלק ב-p_1, או ב-p_2, ... או ב-p_r נקבל מהיחידות ב-$\mathbb{Z}_{n_1} \times \cdots \times \mathbb{Z}_{n_k}$

(23)

$x \in \mathbb{Z}_n$ מספרים שלמים כך ש-

$x \bmod p_i^{\ell_i}$ שקול לפחות מספר אחד p_i לכל i.

$$\left(\ ,\ \ldots\ ,\ \right)$$

$p_1^{\ell_1} - \frac{p_1^{\ell_1}}{p_1}$ אפשרויות

$p_2^{\ell_2} - \frac{p_2^{\ell_2}}{p_2}$ אפשרויות

$p_r^{\ell_r} - \frac{p_r^{\ell_r}}{p_r}$ אפשרויות

$$\varphi(n) = \left(p_1^{\ell_1} - \frac{p_1^{\ell_1}}{p_1}\right) \cdots \left(p_r^{\ell_r} - \frac{p_r^{\ell_r}}{p_r}\right) = \quad \Leftarrow$$

$$= p_1^{\ell_1} \cdots p_r^{\ell_r} \prod_{p \mid n} \left(1 - \frac{1}{p}\right) = n \prod_{p \mid n} \left(1 - \frac{1}{p}\right)$$

מספרים ראשוניים

(11)

(24) 20.12.06
ת. שלמה

שם: אור בן-שפרוט
ת.ז: 039...

Ethernet

הגדרה: נתונים n תחנות $P_1, \dots, P_n$ שמתכוונים לשדר. הצעד אותן ל"סיבובים".
בכל סיבוב ניתן לכל תחנה בודדת או לכלום. לכולנו לשקר אין משמעות.
אם תחנה אחת מצליחה ישלח בהצלחה את הודעתה. אם יותר מאחת תשלח - יהיה התנגשות.
בקירוב: כל תחנה יכול מטבע מאזן הסיבוב p ל-1 באופן
מסוים ת"ל בהסתברות. אם "כן" 1 ישלח בהצלחה. אחרת לא קורה כלום.
כל תחנה מטבע מאזן מטבע אין שם מי בהצלחה ומי לא.
ניתוח: נגדיר $A_{i,t}$ - התחנה ה-i ניסה לשדר בזמן t.
נגדיר $F_{i,t}$ - הצליח ל-i בדיוק בזמן t.

מכיוון ש- $F_{i,t} = A_{i,t} \cap \bigcap_{j \neq i} \overline{A_{j,t}}$

אם i מצליח — על כל האחרים שלא ניסו

$P(A_{i,t}) = p$ $\quad P(\overline{A_{i,t}}) = 1-p$ $\quad \forall i = 1, \dots, n$

ומכיוון שהאירועות הנ"ל הם בלתי תלויים אפ

$$P(F_{i,t}) = P(A_{i,t} \cap \bigcap_{j \neq i} \overline{A_{j,t}}) = P(A_{i,t}) \cdot \prod_{j \neq i} P(\overline{A_{j,t}}) = p \cdot (1-p)^{n-1}$$

נמצא p שמאקסם את $P(F_{i,t})$. את נמצא מקסימום ע"י $f(p) = p(1-p)^{n-1}$
לכאורה $p \in (0,1)$.

$$f'(p) = (1-p)^{n-1} - (n-1)p(1-p)^{n-2} = (1-p)^{n-2}((1-p) - (n-1)p) = (1-p)^{n-2}(1-np)$$

לכן לכאורה שמקסימום מתקבל עבור $p = \frac{1}{n}$.

$$P(F_{i,t}) = \frac{1}{n}(1-\frac{1}{n})^{n-1}$$

אז אם $p = \frac{1}{n}$ אז $P(F_{i,t}) \geq \frac{1}{2e}$ $\Leftarrow$ $(1-\frac{1}{n})^{n-1} \geq \frac{1}{2}$.
$\Leftarrow$ נתבונן על מה קורה בסיבוב יחיד. הסתברות שהוא יצליח.

נסמן ב- $B_{i,t}$ את האירוע שתחנה i כשל בכל הסיבובים $1, \dots, t-1$.

$$P(B_{i,t}) = P(\bigcap_{r=1}^{t} \overline{F_{i,r}}) = \prod_{r=1}^{t}(1-P(F_{i,r})) = (1-\frac{1}{n})^{n-1})^{t} \geq (\frac{1}{2e})^{t}$$

נשים ♡ שאם ניקח $t = 2en\ln 2 = \Theta(n)$ נקבל ש-

$$P(B_{i,t}) \le \left(1-\frac{1}{en}\right)^{cen} = \left[\left(1-\frac{1}{en}\right)^{en}\right]^{c} \xrightarrow{n\to\infty} \frac{1}{e^c}$$

ניקח $t = cen\log n$ ואז

$$P(B_{i,t}) \le \left(1-\frac{1}{en}\right)^{cen\log n} = \left[\left(1-\frac{1}{en}\right)^{en}\right]^{c\log n} \approx \frac{1}{e^{c\log n}} = \frac{1}{n^c} \to 0$$

נבדוק מה קורה כשאנחנו מגדילים את t יותר.
אפשר להשתמש באותו אינדיקטור I_i לפעולה ההפוכה, אפשר פשוט לחשוב מחדש.
נזכיר את הקסם שבאי-שוויון:
אם $A_1, \ldots, A_k$ מאורעות, אז $P(\bigcup_{i=1}^{k} A_i) \le \sum_{i=1}^{k} P(A_i)$

ליתר דיוק - B_t המאורע שלפחות אחד לא נפגע אחרי t סיבובים ואז

$$P(B) = P\left(\bigcup_{i=1}^{n} B_{i,t}\right) \le \sum_{i=1}^{n} P(B_{i,t}) \le n\left(1-\frac{1}{en}\right)^{t} = n\left(1-\frac{1}{en}\right)^{cen\log n}$$

ניקח $t = cen\log n$ (נקבע $c = \Theta(n\log n)$)

$$P(B) \le n\left(1-\frac{1}{en}\right)^{cen\log n} = \left(\left(1-\frac{1}{en}\right)^{en}\right)^{c\log n} \approx n\,\frac{1}{e^{c\log n}} = \frac{n}{n^c} = \frac{1}{n^{c-1}} \xrightarrow[\text{אם } c>1]{} 0$$

אי-שוויון צ'רנוף

נתון רצף של משתנים מקריים $X_1, X_2, \ldots$ אשר שווים
0 או 1 ולכל $\varepsilon > 0$

$$P\left(\left|\frac{\sum_{i=1}^{n} X_i - E\left[\sum_{i=1}^{n} X_i\right]}{n}\right| > \varepsilon\right) \xrightarrow{n\to\infty} 0$$

אי-שוויון צ'רנוף: אם $X_1, \ldots, X_n$ הם משתנים מקריים 0 או 1 אז

$$P\left(\left|\frac{\sum_{i=1}^{n} X_i - E\left[\sum_{i=1}^{n} X_i\right]}{n}\right| > \varepsilon\right) \le \frac{c}{n}$$

סכום צ'רנוף: יהיו $X_1, \ldots, X_n$ הם משתנים מקריים 0 או 1. (סה"כ $X = \sum_{i=1}^{n} X_i$
$\mu = E[X] = \sum_{i=1}^{n} E[X_i]$! נתון $\delta > 0$ כלשהו. אז

(1) $P(X \ge (1+\delta)\mu) \le e^{-\frac{\delta^2\mu}{3}}$

(2) $P(X \le (1-\delta)\mu) \le e^{-\frac{\delta^2\mu}{2}}$

Oracles - אורקלים

נראה שאותן אורקלים שניתן לממש ע"ג מקור של $\frac{1}{2}<d$ מספיק גם עבור מקור של d כלשהו. אותנו נרצה לממש באמצעות אלגוריתם שנקרא לו A. נשתמש את האלגוריתם A פעמים רבות כדי לקבל את הדרוש.

נניח שיש לנו אלגוריתם A שמקבל r מהמקור d ו- 0 אחרת, ומחזיר את ה-x_i המשויך אליו, שאותו אנו מחפשים.

נשתמש במסמה ברנולי:

$$P\left(\sum_{i=1}^{n} x_i \geq \frac{n}{2}\right) = P\left(\sum_{i=1}^{n} x_i \geq \underbrace{\left(\frac{1}{2d}-1\right)}_{m} \cdot dn\right) \leq e^{-2\left(\frac{1}{2d}-1\right)^2 dn} = e^{-\left(\frac{1}{2d}-1\right)^2 \cdot 2dn}$$

$$\frac{1}{2} \neq d \Rightarrow$$

$\Leftarrow$ לא מתאפס.

$\Rightarrow$ האלגוריתם A מחזיר תמיד תשובה.

אזון עומסים - Load-Balancing

הבעיה: נניח שיש לנו מערכת של n מעבדים שבה m עבודות מגיעות באופן סטוקסטי ובלתי תלוי אחת מהשנייה.

נרצה לתכנן את העבודה בצורה שתזער את ה מעבדים.

אורי: הבנתי...	
אוהד:	סטוקסטי שאני שלה.
אורי:	... אתה יודע
	מה זה תהליך ?
סטוקס:	כן, אני לומד
	עכשיו הסתברות.
	סטוקסטי אסמפטוטי

הבעיה: נניח שיש לנו n מעבדים ו- m עבודות. נסמן ב- X_j את מספר העבודות שהוקצו למעבד j. את X_j נגדיר כסכום $X_j = \sum_{i=1}^{m} Y_{i,j}$ כאשר $Y_{i,j}$ שווה 1 אם העבודה i הוצבה במעבד j. אז $E[X_j] = \frac{m}{n} = E[\sum_{i=1}^{m} Y_{i,j}]$

פתרון: עבור מעבד j נסמן X_j - מספר העבודות שהוקצו אליו. אז $\mu = \frac{m}{n} \cdot \frac{1}{2}$, $\delta = \frac{m}{n} \cdot \frac{1}{2}$, נשאר לחסום את ההסתברות שמעבד יקבל יותר מ- n עבודות. נשים לב שמתקיים $\mu + \delta = \frac{m}{n}$.

$$P(|X_j - E[X_j]| > \frac{1}{2}E[X_j]) = P(X_j < \frac{1}{2}E[X_j]) + P(X_j > \frac{3}{2}E[X_j]) \overset{\text{צ'רנוף}}{\leq} e^{-\frac{\left(\frac{1}{2}\cdot E[X_j]\right)^2}{2E[X_j]}} + e^{-\frac{\left(\frac{1}{2}\cdot E[X_j]\right)^2}{3E[X_j]}} = 2\cdot e^{-\frac{m}{12n}}$$

כאשר נבחר את הקבועים שהופיעו באי השוויונות כך שיתקיים

$$P(\bigcup_{j=1}^{n}\{|X_j - E[X_j]| \geq \frac{1}{2}\}) \leq \sum_{j=1}^{n} P(|X_j - E[X_j]| \geq \frac{1}{2}) = \left(\frac{n}{m}\right)\cdot 2e^{-\frac{n\varepsilon^2}{m}} = 2e^{\log n - \frac{n\varepsilon^2}{m}}$$

נבחר $m = c\cdot n\log n$! $c > \frac{1}{\varepsilon^2}$ אז מקבלים

$$\log n - \frac{n\varepsilon^2}{m} = \log n - \frac{1}{c}\log n = (1-\frac{1}{c})\log n \longrightarrow -\infty$$

$$\Rightarrow e^{\log n - \frac{n\varepsilon^2}{m}} = \frac{1}{n^{c-1}} \longrightarrow 0$$

<u>מסקנה:</u> יש $m = \Theta(n\log n)$ כאלו שמדגם בגודל m מקיים כי ההיסטוגרמה של המדגם קרובה לפי $\frac{1}{2}$ לזו האמיתית של ההתפלגות.

ההיסטוגרמה המדגמית קרובה להתפלגות האמיתית פי $\frac{1}{2}$ בהסתברות $1-\frac{1}{n}$.

אלגוריתמים הסתברותיים

יהי רשת שבה $G=(V,E)$
בין קודקוד v שלה הדרגה לקודקוד $d(v)$
ניתנה מילה בו מוקדם f כך שלכל $v_1, v_2 \in V$
$f(v_1) \neq f(v_2)$ אפשר ל- v_1 ו- v_2.
לאחר מילה שבדק $*$ שאלנו -
- $*$ נתן במדר x את שניהם לבין y ושלנו
התאמה ואנו נשלח אז נדע מהתאמה לבדוק.

אפשר גם אילו שאותו קודקוד עבור לשלוח באותו הכיוון יתר
מודעה אותה מאותו הצמן. אלא אם שני הודעות מגיעות
לקודקוד מאותו צמן ונשמו ביניהם שהוא מבדיר אותן כיון
אלא אותה מהשתות לרצה מתכות.

ניתנה של S בעזרת f -
עבור שום יהם S סדר $>$ על V. בקודקוד v שולח הודעה
f - $d(v)$ המסלול $(d(v), f)$
אם ההודעות u_1, u_2 מאלות לאותו קודקוד w והודעת
שהמשיכ מאותה בעם, אם $u_2 > u_1$ אז ההודעה u_1 - u מחכה
מתור. כמן ניתנה = אפשר הדרות עד שכל ההודעות מגיעות
לייעדן.

אותו נדבר ספציפית על - אחת הקלבות Hypercube - $Q_n = \{0,1\}^n$
$u, v \in V$ מחוברות
בדיוק אם u, v
שונות במקום אחד בדיוק.

ניתנה טבעי מקומיה : $u = (a_1, \ldots, a_n)$

$$(a_1 \; a_2 \ldots a_n)$$
$$\vdots$$
$$(b_1 \ldots b_i \; a_{i+1} \ldots a_n)$$
$$\vdots$$
$$v = (b_1, \ldots, b_n)$$

ם בדיוק משניים מיט את $\vee$ שך שמאפיים פסוק.
שמאל לימין

למשל $u = 0\,0\,1\,1\,0$

$1\,0\,0\,1\,1 = v$

$1\,0\,0\,1\,1$

$1\,1\,0\,0\,1$

$0\,0\,1\,1\,1$

$0\,1\,1\,1\,0$

אוכן האפשרים בניתנה בגלה בהם $\leq n$.

קיומה ניתנה שיוכה "מהה" למן :
(שיה ש- $\sigma(0,\ldots,0) = (1,0,\ldots,0)$ - סומן הקיבוצים הנתנים לאותו מקום.
למן הניתנה הזאת שפחות $\frac{n}{1-n}$ (כי σ פעם $(0,\ldots,0)$
מקבלת σ ויותר n הופעות).
נובע כי אלה בדיוק כל האלו, את אחדו (יחוד את שלמיני
שמרה ש- σ תאורה.

קיומה σ תאורה שלה למן ניתנה ארוך :

$$\sigma(a_1 \ldots a_n) = (a_n \ldots a_1)$$

ונמצא שם התאורה ה למה ניתן שיוכרו הנתנה שקיום.
הדרך שבאה מהקבוצה מהדורה $(0 \ldots 0 \; a_{\frac{n}{2}} \ldots a_1)$
עומרת בדיק קבוצה $(0 \ldots 0)$. לה ברור כי הזאת בדיוק
להיות $p = (a_1 \ldots a_{\frac{n}{2}} \; 0 \ldots 0)$ וכשהתקנים קומבינציות אותה
אותה את אחרי $\frac{n}{2}$ שמבים מיוחים σ $(0 \ldots 0)$.
אם פר הדרך מהדורה הזאת הזאת $2^{n/2}$ למן הניתנה הזאת σ
$\frac{2}{1-2^{\frac{n}{2}}}$.

אלגוריתם גיתלה הסתברותי של L.Valiant:

$g(x)$ קלקול x ואז שולח $f(x)$ קלקול x מאחר $f(x) \to g(x)$ $\Sigma \to \Sigma$ הוא שני גיתלה הבינים הם הניתלה הרשמי.

נראה שהסתברות ראשה פאן הניתלה הוא קבל.
נשים ❤ שכאם הודעה שאורה n מתכה לחלוקה $n-1$ אז לאחר $n-1$ ואחלוק $n-1$ ים בלק אשותפה.

$\Lambda \to \Lambda$ g תוצאה "ערוי" אם בים קלקול n כך שהאחלוק $g(n) \to n$ תותק (לומר יש בלק אשותפה) שהחות n אחלוקים אחרים.

לם אפשר לתאר Σ אבר ב-X ע"י אשפט (בינארי) באורך k אז $|x| \leq 2^k$.

נעית ש-g רצה. נניח שיש k_i אחלוקים שחותכים את האחלוק $g(n) \to n$ בלק ה-i הלא פנוי.

נשומ את האשפט הבא:

$g(n)$ n $\underbrace{n-1 \quad \dots \quad n-1 \quad n-1}_{= cn(n-1)}$ n n אם בינים

"$k_n \dots k_1$" $[\ ][\ ]\dots[\ ][\ ]$

אז לה נתנים כדי שאוכ נוכל לפענח את האשפט ואחלק לה אחלוקים אחלוק g כי $\frac{1}{2}$ מה תיאור כל $1+cn$ קלקולים. נשאר $1-cn-n^2$ קלקולים נוספ האשפט פאן הם הנלכים אל לה "בח $(1-cn-n^2)n$ קלקולים. נניח שהקלקולים אחרים לפי הסדר אל נכל אין לפענח את כה.

<u>אשפט</u> (לאהוכה) לכל נוכל בקלה קיאת תאורה Σ כך שאת הניתלה $\geq \frac{1}{2n^2}$.

יש $\binom{n+cn}{n}$ שמתקיים $\sum k_i = cn$ שמספרים טבעיים $k_1, \dots, k_n$ כאלה ש-

$$\begin{cases} k_1 + \dots + k_n = cn \\ cn \geq k_i \geq 0 \end{cases}$$

האם מיוסים בדיוק כמה מקרים אפשר לבחור את $k_1, \dots, k_n$?

נניח שאפשר $\binom{n+cn}{n} \geq y \geq 1$ עבור מספר הבוחרים

$$\binom{n(1+c)}{n} \geq \frac{n^{n(1+c)}}{(\frac{n}{e})^n} = n^{n(1+c)} e^n$$

$$\Rightarrow \text{מספר הבוחרים} \leq n^{n(1+c)} e^n = n(\log_2 e + (1+c)\log_2 n) \leq$$

$$\leq c\tfrac{1}{2} \cdot n$$

(נמוך $c > 10$)

נאמין לשמאל ונשים לב שמתקיים

$$cn^2 - \tfrac{1}{2}n^2 \geq n(1-cn-n^2) + nk_{n-1} + (1-n)cn + n+n$$

לסיכום, g הפלטנו לנו שיש $cn^2 - \frac{1}{2}n^2$ שאותו

g - g מקיים את התנאי אותה נרצה עם קונקרט

שאין הוא הנלמד עבור n^2 מיוסים.

אם X קבוצת n ו- g הדטרמיניסטי. אז

$$|X| \geq 2^{n^2 - cn^2}$$

$$\Rightarrow p(\text{מקרה } g) = \frac{|X|}{\text{מספר הבוחרים ל-}g} \geq \frac{2^{n^2 - cn^2}}{n^n 2^n} = 2^{-cn^2}$$

ונקה קטן!

אבל אם g לא רצה אבל n הוכחה מפרק $V \to g(V)$

את הכבוד פחות מ- cn פרמטרים.

נקח מחיקה g- $g(V)$ אבל עם n היותר $n(1+c)$ בשבוע.

את אותו הניתוח עושים. אם נחזיר משני של המטלה

אז ה"כ הסתברות לבחור קונקרטי.

שיטת ההצפנה של רבין (vsh)

שלבי ה-RSA ויתכן שגם ויש שניתן לפתוח אשרו על הקוד, עם השכנה שלו.

הוכחת שגיאות שעובדות. לאחר הדקלה אולפנית. את אחרי $q-2$, צג אפ. $M-$ זוגות $q-2$. לאחשוף יש תאיך אופק מלא אגה, עשים ϕ שיפ' הפן מגה של חד שכיות. אם M שעות שהא כווכאי פוי. $\Rightarrow 1 \neq 2 = \gcd(2, \phi(n))$.

שהחלקה של M תהיה לאי n- $\phi(n)$ מאפרה שלנו $\phi(n) = (1-q)(1-p)$ עשים ϕ שלא לא אגה פרטי של RSA כי שם יש פרישה n ונותן הוכחה $M > M \geq 0$ אלפנים $n \bmod M^2 = c$.

האפתח הצפורי היה $pq = n$.

ראשונים רצלה, אחרי לא מאוד ההפה פאן נאבא את שלך. אספר שאחר הלפיפות של רצלה היו $\frac{\log_2 n}{n} \Rightarrow$ פשוט מאוחר ש- $p \equiv q \equiv 3 \pmod 4$. אם אוכלים ראשונים כאלה ? מישהו תבין השגוי האפתח ופרטי מוחרים שני ראשונים קוד אשונקצ אלא כאופן יוקר שהיות אסוסק.

צפיכים שהפצים קשה שלחת ההוכחה שהטכים על אפתח הלפיפים יש את אותו אפתח. ההסרון בלה היה שפשו ראשונים לא נקבע שיטת הצפנה אסימטרית. שיטע הצפנה סימטרית לה רשפש, הוכחת האפתח הפרטי שלה.

אשונקצ אלא כאופן יוקר שהיות אסוסק.

מישהו תבין השגוי האפתח ופרטי מוחרים שני ראשונים קוד

אספר שאחר הלפיפות של רצלה היו $\frac{\log_2 n}{n} \Rightarrow$ פשוט מאוחר

ראשונים רצלה, אחרי לא מאוד ההפה פאן נאבא את שלך.

האפתח הצפורי היה $pq = n$.

נוותן הוכחה $M > M \geq 0$ אלפנים $n \bmod M^2 = c$.

שהחלקה של M תהיה לאי n- $\phi(n)$ מאפרה שלנו $\phi(n) = (1-q)(1-p)$

נשים ϕ שלא לא אגה פרטי של RSA כי שם יש פרישה

$p-2$, צג אפ. $M-$ זוגות $q-2$. לאחשוף יש תאיך אופק

מלא אגה, עשים ϕ שיפ' הפן מגה של חד שכיות. אם M שעות

שהא כווכאי פוי. $\Rightarrow 1 \neq 2 = \gcd(2, \phi(n))$.

הוכחת שגיאות שעובדות. לאחר הדקלה אולפנית. את אחרי

$p-2$, צג אפ. $M-$ זוגות $q-2$. לאחשוף יש תאיך אופק

מלא אגה, עשים ϕ שיפ' הפן מגה של חד שכיות. אם M שעות

שהא כווכאי פוי. $\Rightarrow 1 \neq 2 = \gcd(2, \phi(n))$.

שהחלקה של M תהיה לאי n- $\phi(n)$ מאפרה שלנו $\phi(n) = (1-q)(1-p)$

נשים ϕ שלא לא אגה פרטי של RSA כי שם יש פרישה

נוותן הוכחה $M > M \geq 0$ אלפנים $n \bmod M^2 = c$.

האפתח הצפורי היה $pq = n$.

ראשונים רצלה, אחרי לא מאוד ההפה פאן נאבא את שלך.

אספר שאחר הלפיפות של רצלה היו $\frac{\log_2 n}{n} \Rightarrow$ פשוט מאוחר

ראשונים רצלה, אחרי לא מאוד ההפה פאן נאבא את שלך.

מישהו תבין השגוי האפתח ופרטי מוחרים שני ראשונים קוד

אשונקצ אלא כאופן יוקר שהיות אסוסק.

צפיכים שהפצים קשה שלחת ההוכחה שהטכים על אפתח

הלפיפים יש את אותו אפתח. ההסרון בלה היה שפשו ראשונים

לא נקבע שיטת הצפנה אסימטרית. שיטע הצפנה סימטרית לה רשפש,

הוכחת האפתח הפרטי שלה.

מישהו תבין השגוי האפתח ופרטי מוחרים שני ראשונים קוד

אשונקצ אלא כאופן יוקר שהיות אסוסק.

צפיכים שהפצים קשה שלחת ההוכחה שהטכים על אפתח

הלפיפים יש את אותו אפתח. ההסרון בלה היה שפשו ראשונים

לא נקבע שיטת הצפנה אסימטרית. שיטע הצפנה סימטרית לה רשפש,

הוכחת האפתח הפרטי שלה.

שיטת ההצפנה של רבין (vsh)

שלבי ה-RSA ויתכן שגם ויש שניתן לפתוח אשרו על הקוד,

עם השכנה שלו.

שאפשרויות לכל אחד את ההצלחה האחרונה וכמו, אבל רשימה קצרה

של הדרכות אפשרויות שאחרי אחד הכתובת.

אז איך אפשרויות? (אחרי):

$$m_p = C^{\frac{p+1}{4}} \bmod p$$

$$m_q = C^{\frac{q+1}{4}} \bmod q$$

$$\Rightarrow m_p^2 \bmod p = C^{\frac{p+1}{2}} \bmod p = (M^2)^{\frac{p+1}{4}} \bmod p =$$

$$= M^{p+1} \bmod p = M^2 \bmod p = C \bmod p$$

$$\text{כי } a = a^2 \bmod p$$

באותו אופן $m_q^2 \bmod q = M^2 \bmod q = C \bmod q$

אותנו אפשרויות x כך ש- $x^2 \bmod n = C$.

$$x^2 \bmod p = C \bmod p = m_p^2 \bmod p$$

$$x^2 \bmod q = C \bmod q = m_q^2 \bmod q$$

אולם משנה, לשאלה רמיזות יש בדיוק שתי אפשרויות לכן

$$x \bmod p = \pm m_p$$

$$x \bmod q = \pm m_q$$

ואפשר אפשר לאחד את ההודעה בדרכים אפשרי ההצלחה הכוללת.

$M_1 \equiv m_p \pmod p$	$M_1 \equiv m_q \pmod q$
$M_2 \equiv -m_p \pmod p$	$M_2 \equiv m_q \pmod q$
$M_3 \equiv m_p \pmod p$	$M_3 \equiv -m_q \pmod q$
$M_4 \equiv -m_p \pmod p$	$M_4 \equiv -m_q \pmod q$

דוגמה: $q=11$ $p=7$ $n=77$

$$C = 100 \bmod 77 = 23 \quad \Leftarrow \quad M = 10$$

לפתרון: $m_p = 23^{\frac{7+1}{4}} \bmod 7 = 2^2 = 4$

$m_q = 23^{\frac{11+1}{4}} \bmod 11 = 1^3 = 1$

אפשר בקלות לפתור את המשוואות

$$x \bmod 7 = \pm 4$$

$$x \bmod 11 = \pm 1$$

נראה כללי: אם $\gcd(n_1, n_2) = 1$ ומתקיים x כך ש-

$$x \bmod n_1 = a_1$$

$$x \bmod n_2 = a_2$$

אחפש c_1 כך ש- $c_1 \bmod n_1 = 1$ $\quad c_1 \bmod n_2 = 0$

ו- c_2 כך ש- $c_2 \bmod n_1 = 0$ $\quad c_2 \bmod n_2 = 1$

ואז $x = a_1 c_1 + a_2 c_2$

במקרה שלנו c_1, c_2 יהיה הפתרון של $c_2 = 56$, $c_1 = 22$

$$\Rightarrow M_1 = 4 \cdot 22 + 1 \cdot 56 \equiv 67 \pmod{77}$$

$$M_2 = -4 \cdot 22 + 1 \cdot 56 \equiv 45 \pmod{77}$$

$$M_3 = 4 \cdot 22 - 1 \cdot 56 \equiv 32 \pmod{77}$$

$$M_4 = -4 \cdot 22 - 1 \cdot 56 \equiv 10 \pmod{77}$$

<u>הערה</u>: אם אנחנו מאפשרים את קיומו של $M = 0$ $\quad \frac{n}{2} > M \geq 0$

אז בפיענוח נישאר רק עם 2 אפשרויות.

נשאלים שהפלגה הסתברותית היא $n = p \cdot q$ מתוך את p וכך את q. זה לא אומר לנשום כדי אבל מבין שניהם והאלגוריתם הטוב ביותר שידוע היום (לפחות ברחם ☺) לשבור את זה בערך ב- $O(2^{\sqrt[3]{k}})$ כאשר k $\log n$.

<u>טענה</u> זה קיים אלגוריתם A כך ש- $C = n \bmod [A(c)]^2$ בדרך אבן אפשר לחשב את הפירוק לגורמים של n בדרך יעילה (כאשר מניחים ש- A פועל ב- $O(1)$).

טענה: אם $M^2 \equiv M'^2 \pmod n$ ואם $M \neq \pm M'$

אז $1 < \gcd(M+M', n) < n$ (כלומר $\gcd = q$ או p)

הוכחה: לפי ההיתלה שעשינו בהגדרה יש שתי אפשרויות

1) $\begin{cases} M \equiv M' \pmod p \\ M \equiv -M' \pmod q \end{cases}$ ⇓ $M + M' \equiv 0 \pmod q$

אבל $M + M' \neq 0$

$\Rightarrow q = \gcd(M+M', n)$

2) $\begin{cases} M \equiv -M' \pmod p \\ M \equiv M' \pmod q \end{cases}$ ⇓ $M + M' \equiv 0 \pmod p$

אבל $M + M' \neq 0$

$\Rightarrow p = \gcd(M+M', n)$

☺

מסקנה: אם יש לנו M, M' כנ"ל אז נוכל לפרק את n לגורמים ראשוניים $p \cdot q = n$.

הוכחת הטענה: אלגוריתם לחישוב p ו q:

נבחר A כלשהו ונחשב $M' = A^{(M^2)} \bmod n$.

אם $M \neq M'$ ואם $M \neq -M'$ ניקח.

מסיכוי $\frac{1}{2}$ נבחר M כך ש $M \neq M'$ ואם $M \neq \pm M'$.

אחרי שנחזור על כך מספיק פעמים נמצא את מה שצריך.

☺

שאלה: יש n אנשים שרוצים לדעת את ממוצע המשכורות ביניהם אבל מצד שני, לכל אחד חשוב שאף אחד לא ידע את המשכורת שלו.

תשובה: לכל אחד i מהם נגריל מספר אקראי $x_i \in [1-N, \ldots, N]$ כאשר N גדול מאוד ביחס למשכורת שלהם.

לאחר מכן אדם מס' 1 יוסיף למשכורת שלו את x_1 ויעביר את הסכום לאדם מס' 2 שיוסיף את x_2 וכך הלאה עד שאדם n יעביר לאדם 1 את $M \bmod (x_1 + x_2 + \ldots + x_n)$.

כעת כל אחד מהם יכול לחסר את x_i שלו ולחלק ב n כדי לקבל את הממוצע.

הטענה היא שמתוך אחד מהשתנים לא יוצא שום דבר
על המשכורות הספציפיות. ראו נקודה זו בעיין של הקופה N.
אם לא היה את זה אז השיטה השני היה מקבל מספר אחרי
מהרווח $[x_1, x_1 + N)$. אם עושים את הניסוי הזה פעמים
אז מספיק שהוה מיניאמום אחת התוצאות ש-2 מקבל הוא
המשכורת של הראשון (כי נניח את מספרים ישני $0 = r$).
כשעושים את התישומים אופקו N אז כן פשוט מתקבל שהם
שהוא מפורש בדרך אחידה על $[N, 0]$, ואפה הכר אי
אפשר לעלות שום דבר.

אלגוריתמי קירוב

תזכורת: אלגוריתם מקבל $0<\varepsilon<1$ אם לכל אפשרי I הוא מחזיר פתרון S כך ש $|S| \geq (1-\varepsilon)|Opt|$.
אפשר לחלק את זה בכל אי. שוויון בכיוון ההפוך אלגוריתם שהוא ε קירוב.
שהוא $\frac{1}{\varepsilon}$ קירוב - נקרא הקירוב הכי גרוע שקבלנו.

הגדרה: נאמר שאלגוריתם הוא סכמת אפרוקסימציה (approximation sheme) אם הוא מקבל כפרמטר קלט נוסף $\varepsilon > 0$ ומחזיר $(1+\varepsilon)$-קירוב לפחות.

- נאמר שסכמת אפרוקסימציה היא PTAS (polynomial time app. sheme) אם לכל $\varepsilon > 0$ קיים אלגוריתם בעל זמן הריצה הכי פולינומיאלי באורך הקלט.
- נאמר שסכמת אפרוקסימציה היא FPTAS (Fully PTAS) אם לכל זמן הריצה שלה פולינומיאלי ב- $\frac{1}{\varepsilon}$ ובאורך הקלט.

דוגמא - סכמת אפרוקסימציה שרצה ב- $O(n^{\frac{1}{\varepsilon}})$ היא PTAS אבל לא FPTAS.

- סכמת אפרוקסימציה שרצה ב- $O(\frac{1}{\varepsilon^2}n^{\frac{3}{\varepsilon}})$ היא PTAS ואף FPTAS.

Subset Sum

הגדרה: נתונים מספרים $a_1, \ldots, a_n \in \mathbb{N}$ ומספר $t \in \mathbb{N}$ אנו מחפשים את תת הסכום הכי קרוב אל t. כלומר מוצאים $S \subseteq \{1,\ldots,n\}$ שמקיים $\sum_{i \in S} a_i$ מקסימלי.

הערה: הבעיה היא מקרה פרטי של ה- knapsack 1-0 כאשר המשקלים שווים לערכים של התכיים.

צריך לדעת אלגוריתם פולינומיאלי לפתרון הבעיה.

פתרון אפשרי ואקספוננציאלי:

נסמן ב- P_i את אוסף הסכומים הזרים של $\{x_1,\dots,x_i\}$ שקטנים או שווים ל-t. ושם ♡ ש-

$$P_0 = \{0\}$$

$$P_1 = \{0, x_1\} \setminus \{\text{הסכומים שגדולים מ-}t\}$$

$$\vdots$$

$$P_i = \left(P_{i-1} \cup \{P_{i-1} + x_i\}\right) \setminus \{\text{הסכומים שגדולים מ-}t\}$$

Exact-Subset-Sum $(S=\{x_1,\dots,x_n\}, t)$

```
n ← |S|
L_0 ← <0>          // רשימה
for i = 1,...,n     // נוסיף את x_i לכל האיברים
    L ← Merge(L_{i-1}, L_{i-1} + x_i)
    remove from L_i elements
      greater than t
return max(L_n)
```

- אורך הרשימה L_n יהיה לכל היותר 2^n (לאחר כל הסכומים הזרים של $x_1,\dots,x_n$ - שם כולם שונים ובטוחים מ- t).
לכן כאן הריצה תהיה אקספוננציאלית.
- אם אורכי הרשימות L_i הן פולינומיאליים ב-n, הריצה כולה תהיה כזו (למשל, אם t פולינומיאלי ב-n) את כאן הריצה תהיה כזו אם הגודל פולינומיאלי ב-n. כי Merge פועל ב- $O(|L_i|)$.

נרצה לסכם בקצרה הכול. נתחיל קודם כל ב-L_i כך שאורכה לכל היותר אחד יותר מהקודמת לה. והיה כל אחד.

הרעיון לקרוא לזה trimming הוא שאנחנו מוחקים מ- L את הערכים שקרובים זה לזה מספיק כדי שנוכל להתעלם מאחד מהם.
נאמר ש- $L' \subseteq L$ הוא trimming של L אם לכל $y \in L$ קיים $z \in L'$ כך ש- $\frac{y}{1+\delta} \le z \le y$.
(כאשר δ פרמטר שקובע את איכות הקירוב).

מכיוון שנרצה להגדיר כמה שיותר אופרטורים על trimming אשר לא יפגעו באיכות הקירוב יותר מדי.

דוגמה:

$$L = \langle 10, 11, 12, 15, 20, 21, 22, 23, 24, 29 \rangle$$
$$\delta = 0.1$$

$$L' = \langle 10, 12, 15, 20, 23, 29 \rangle$$

```
Trim(L, δ)        // L - sorted list
    m ← |L|
    L' ← <L[1]>
    last ← L[1]
    for i = 2, ..., m
        if (L[i] > (1+δ)·last)
            append L[i] to the end of L'
            last ← L[i]
    return L'
```

סיבוכיות הקירוב הוא $O(|L|)$.

נוכל להשתמש בזה כדי לבנות FPTAS ל- subset sum.
נאמר $\delta = \frac{\varepsilon}{2n}$.

Approx.- Subset-Sum ($S = \{x_1, \ldots, x_n\}$, t, ε)

$n \leftarrow |S|$

$L_0 \leftarrow \langle 0 \rangle$

for $i = 1, \ldots, n$

$L_i \leftarrow \text{Merge}(L_{i-1}, L_{i-1} + x_i)$

$L_i \leftarrow \text{Trim}(L_i, \varepsilon/2n)$

remove from L_i elements greater than t

return $\max(L_n)$

טענה: Approx-Subset-Sum הוא FPTAS.

הוכחה: נניח כי $\varepsilon > 1$.

1) האלגוריתם $(\varepsilon+1)$-קירוב:

נוכיח באינדוקציה שלכל i לכל $y \in P_i$ קיים $z \in L_i$ כך ש- $y \geq z \geq \frac{y}{(1+\frac{\varepsilon}{2n})^i}$.

נסמן ב- L'_i את הרשימה L_i לפני הביצוע.

עבור $i = 0$ כל ערך נמצא גם ב- L_0 והוא trimming של P_0.

נניח עבור $i-1$. עבור i יהי $y \in P_i$. אז $y \in P_{i-1}$ או $y - x_i \in P_{i-1}$.

לפי הנחת האינדוקציה קיים $z \in L_{i-1}$ המקיים $y \geq z \geq \frac{y}{(1+\frac{\varepsilon}{2n})^{i-1}}$ ולכן $z \in L'_i$ המשך ל- L_i הוא trimming של L'_i קיים z' כך ש- $y \geq z \geq z' \geq \frac{z}{1+\frac{\varepsilon}{2n}} \geq \frac{y}{(1+\frac{\varepsilon}{2n})^i}$

אותו $x_i + z$ שייך ל- L_i לפי הגדרת האינדוקציה $y - x_i \in P_{i-1} \Rightarrow y \in P_{i-1} + x_i$

קיים $z \in L_{i-1}$ כך ש- $x - y \geq z \overset{(**)}{\geq} \frac{x-y}{(1+\frac{\varepsilon}{2n})^{i-1}}$ מכיוון ש- $z \in L_{i-1}$

אז $x_i + z \in L'_i$ לכן קיים $z' \in L_i$ כך ש-

$y \overset{(*)}{\geq} x_i + z \geq z' \geq \frac{x_i + z}{1+\frac{\varepsilon}{2n}} \overset{(**)}{\geq} \frac{x_i + \frac{x-y}{(1+\frac{\varepsilon}{2n})^{i-1}}}{1+\frac{\varepsilon}{2n}} = \frac{x_i - y + x}{(1+\frac{\varepsilon}{2n})^i}$. וזה מה שרצינו.

בפרט הטענה הנכונה עבור $y = OPT \in P_n$ לכן קיים $z \in L_n$ כך ש- $OPT \geq z \geq \frac{OPT}{(1+\frac{\varepsilon}{2n})^n}$. אם כן, כל ערך מתקבל עבור $z^* = \max L_n$ (הפלט של האלגוריתם).

(33)

$OPT \le z^* \left(\frac{\varepsilon}{2n}+1\right)^n \le e^{\varepsilon/2} \cdot z^* = \left(\frac{\varepsilon}{3}\cdot\left(\frac{3}{\varepsilon}+1\right)+1\right) z^*$

$= \left(\frac{2}{3}+1\right) z^*$

$\le \left(\frac{2}{3}e+1\right) z^*$

$\le (3+1)\, z^*$

2) לאור העובדה שאין יכולה פונקציה מ- $\frac{1}{\varepsilon}$ ו-n הקטנים. אזי הקטנים הוא $O(n+\log t)$.

יש לכתוב פתרון פונקציה מ- $\frac{1}{\varepsilon}$, n ו- $\log t$.

נשים ♡ שלא i אחסנים שב- L_i האחרון הוא שאחרי הריקים (אחרי 0) הוא סכומי $\frac{n^2}{\varepsilon}+1$.

$L_i = \langle 0, a_1, a_2, a_3, \ldots, a_r \rangle$

a_1 ; $a_2 \le a_1(1+\delta)$; $a_3 \le a_1(1+\delta)^2$; $a_r \le a_1(1+\delta)^{r-1}$

אזי שמ $t \ge a_r > a_1\left(1+\frac{\varepsilon}{n^2}\right)^{r-1}$.

$\Rightarrow \log_{1+\frac{\varepsilon}{n^2}}(a_1) + (r-1) < \log_{1+\frac{\varepsilon}{n^2}}(t)$

$\Rightarrow r < 1 + \log_{1+\frac{\varepsilon}{n^2}}(t) = 1 + \frac{\ln t}{\ln\left(1+\frac{\varepsilon}{n^2}\right)} \le 1 + \ln t \left(\frac{n^2}{\varepsilon}+1\right) =$

$\ln(1+x) \le \frac{x}{x+1} = \frac{1}{\frac{1}{x}+1}$ $\Leftarrow$ $x > -1$

$= 1 + \ln t \left(\frac{n^2}{\varepsilon}+1\right) = O\left(\frac{n \ln t}{\varepsilon}\right)$

כאן הכיוון של האלגוריתם הוא

$O\left(\frac{n^2 \ln t}{\varepsilon}\right)$ $\Leftarrow$ $O(n \cdot \max|L_i|)$

(11)

טענה: עבור $x > -1$ מתקיים $\ln(1+x) \geq \frac{x}{1+x}$

הוכחה:

$$f(x) = \ln(1+x) - \frac{x}{1+x}$$

$$f'(x) = \frac{1}{1+x} - \frac{1}{(1+x)^2} + \frac{x}{(1+x)^2} = \frac{x}{(1+x)^2} = 0$$

$\Leftarrow$ 0 הינו מינימום מקומי יחיד של f

$x > -1$ כלומר $f(x) \geq f(0) = 0$ ⑪

דבר שרצינו להוכיח - סיימנו את ההוכחה.

מהקשר של אלגוריתמים שנתנו אפשר לחשב את העבודה במצב הקשרים.

- אלגוריתמים שנותנים תשובה שונה מהעבודה (למשל קביעת ראשוני)
- אלגוריתמים שאינתואיטיבי מאפשרים אבחנויות
- אלגוריתמים שמקבלים קלט אקראי (למשל את מניית הבית)

מודל מקבילי

נניח לרגע פשוט שאנחנו מתמשים במודל PRAM (Parallel RAM) ונפתח כאשר מעבדים שונים יכולים לגשת לאותו תא בזיכרון המשותף בו-זמנית. כאשר מדברים על גישה בו-זמנית, ישנן כמה אפשרויות שונות:

- אותו תא - CR (Concurrent Read) - כל המעבדים רוצים לקרוא את אותו תא בזיכרון המשותף.
- לגבי כתיבה אל אותו תא, ישנן כמה אפשרויות:
  1) CW (Concurrent Write) - כל המעבדים רוצים לכתוב לתא בזיכרון המשותף. כאשר כולם כותבים את אותו ערך - פשוט נכתוב אותו. אחרת, נצטרך להחליט מה נכתוב.
  2) EW (Exclusive Write) - לא מאפשר את הגישה לתא בזיכרון המשותף. אם שניים מנסים - נשגר שגיאה.

תרגיל:

1) נתון מערך A בגודל n בתים. נרצה לבדוק האם A מכיל לפחות אחד. נשתמש ב-OR bitwise בזמן O(1).

תהי 1 : מעבד אחד כותב 0 לתא output.

תהי 2 : בסוף כולם, אם יש בו 1 הוא כותב 1 ← output.

2) נאותו שימוש נרצה לבדוק AND bitwise בזמן O(1).

תהי 1 : מעבד אחד כותב 1 לתא output

תהי 2 : בסוף כולם, אם יש בו 0 הוא כותב 0 ← output

ואז נשאר לחשב את הפעולה הבאה:
אם יש EW בסוף התהליך אפשר לדעת בדיוק בסדר את, ואם לא יש (כלומר תהי פשוט אפשר לשים

לפיכך הפלט הוא כמו שצריך.

כעת נחשב את זמן הריצה האסימפטוטי של האלגוריתם. נשתמש במשפט האב לנוסחת הנסיגה:

$T(n) = T(n/2) + O(1)$

$\Rightarrow T(n) = O(\log n)$

$W(n) = 2 \cdot W(\frac{n}{2}) + O(n)$

$\Rightarrow W(n) = O(n \log n)$

תרגיל שני: $x_1, x_2, x_3, x_4, \ldots, x_{n-1}, x_n$

$y_1 = x_1 * x_2$ $\quad y_2$

$y_1 * x_3 * x_4 = y_2$ $\quad y_3$

$\vdots$

$y_{n/2-1} * x_{n-1} * x_n = y_{n/2}$

פתרון: נחשב רקורסיבית את PrefixSums של $y_1, \ldots, y_{n/2}$ ונקבל $\{s'_1, \ldots, s'_{n/2}\}$. לאחר מכן נחשב את התוצאות עבור $\{s_1, \ldots, s_n\}$: $s_i = s'_{\frac{i}{2}}$ עבור i זוגי. עבור i אי-זוגי נשים לב כי $s_i = s'_{\frac{i-1}{2}} * x_i$.

ניתוח:

$T(n) = T(n/2) + O(1)$

$T(n/2)$: קריאה רקורסיבית על מערך בגודל חצי

$O(1)$: עבודה נוספת לכל איבר

$\Rightarrow T(n) = O(\log n)$

$W(n) = 2 \cdot W(\frac{n}{2}) + O(n)$

$\Rightarrow W(n) = O(n \log n)$

תרגיל: נתונים n מטלות עם משכי זמן $d_1, \ldots, d_n$ בהתאמה. נתון כי m מעבדים זהים זמינים לביצוע המטלות. מצאו אלגוריתם שמחשב את זמן הסיום המינימלי האפשרי t של ביצוע המטלות.

(ניתן להניח כי $m \ll n$)

פתרון: נשתמש ב-Scheduling של LPT: נמיין את המטלות לפי משכי הזמן בסדר יורד. נקצה כל מטלה למעבד שיסיים ראשון.

1) $W(n)$ - מספר הפעולות האלגוריתם עבור קלט בגודל n.

2) $T(n)$ - מספר הצעדים הנדרשים לחישוב בקלט בגודל n.

נתון: n מטלות, m מעבדים

נסמן ב-t את זמן הסיום המינימלי האפשרי.

הוכחה: נסמן ב-i_m את מספר המטלות שהוקצו למעבד i $(1 \leq i \leq m)$

מספר d המקסימלי שניתן לקבל הוא $\left\lceil \frac{i_m}{m} \right\rceil$ ולכן סכום כל המטלות הוא לכל היותר $\sum_{i=1}^{m} \left\lceil \frac{i_m}{m} \right\rceil$

$t + \frac{d}{m} = t + d/m \cdot (\sum_{i=1}^{m} i_m) = (1 + \frac{d}{m}) \sum_{i=1}^{m} i_m \geq \sum_{i=1}^{m} \left\lceil \frac{i_m}{m} \right\rceil$ משפט

Prefix Sums

הבעיה: נתונים $x_1, \ldots, x_n$ איברים מתוך קבוצה S. עם S מוגדרת פעולה בינארית אסוציאטיבית $S \leftarrow S^2 * *$. אותן מחשבים את $(s_1, \ldots, s_n)$ כך ש- $s_i = x_1 * x_2 * \ldots * x_i$ $\quad (1 \leq i \leq n)$

רעיון ראשון: נפעיל באופן רקורסיבי את PrefixSums על $y_1 = x_1 * x_2, \ldots, y_{n/2} = x_{n-1} * x_n$. נאחסן את התוצאות ב- $(s'_1, \ldots, s'_{n/2})$; $(s_1, \ldots, s_n)$ $\{x_1, \ldots, x_n\}$. לאחר מכן נחשב את s_i עבור i זוגי ואי-זוגי בנפרד. עבור i זוגי נשים לב כי $s_i = s'_{\frac{i}{2}}$. עבור i אי-זוגי $i > 1$ נשים לב כי $s_i = s'_{\frac{i-1}{2}} * x_i$.

אפשר להוריד מעלה

1 if (n-1)

$S_1 \leftarrow X_1$

2 for $1 \le i \le \frac{n}{2}$

$y_i \leftarrow X_{2i-1} * X_{2i}$

$T(\frac{n}{2})$ Recursively compute prefix sums of $\{y_1, .., y_{n/2}\}$ and assign them to $(S'_1, .., S'_{n/2})$

$T(\frac{n}{2})+3$ for $i \leftarrow 1 .. n$

if (i is even)

$S_i = S'_{i/2}$

else if $(i = 1)$

$S_i = X_1$

else

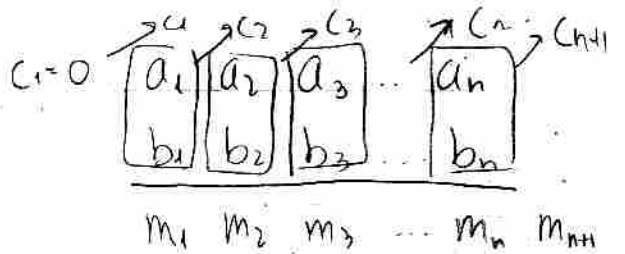
$S_i = S'_{\frac{i-1}{2}} * X_i$

$T = O(\log n) \Leftarrow T(n) = T(\frac{n}{2}) + O(1)$ נוסחת נ

$W(n) = O(n) \Leftarrow W(n) = W(\frac{n}{2}) + O(n)$

- Prefix Sums תזכורת שבה אפשרים

נתונים שני מספרים בני n סיביות לחבר ביניהם.



בכל שלב $i=1$ ניצור x_i שהקלדה של y_i ו- y_{i+1} תאפשר

לקבל c (carry) ואחרת ה carry של a_i+b_i+c

$x_i(c) \equiv 1$ $a_i = b_i = 1$

$x_i(c) \equiv 0$ $a_i = b_i = 0$

$x_i(c) = c$ $a_i = 1$ $b_i = 0$ $\vee$ $a_i = 0$ $b_i = 1$

נרצה לחשב את

$$C_1 = 0$$

$$C_2 = X_1(0)$$

$$C_3 = X_2 \circ X_1(0)$$

$$\vdots$$

$$C_{n+1} = X_n \circ X_{n-1} \circ \dots \circ X_1(0)$$

נשים ♡ שפעולת ההכפלה הזו אסוציאטיבית ולכן ניתן לחשב את כל ה-prefix sums של $X_n, \dots, X_1$, אחרי שנדע ש $C_1, \dots, C_{n+1}$ שווה ל $m_i = (a_i + b_i + c_i) \bmod 2$ עבור $1 \le i \le n$ נסיים. $m_{n+1} = c_{n+1}$!-

סה"כ נקבל $W(n) = O(n)$ ו $O(\log n)$ זמן ומספר הפעולות.

אפיונות אטומיות ב- $O(1)$ זמן במודל CRCW

נרצה לחשב אטומיות של $X_1, \dots, X_n$ אסוציאטיביות שמגדירים באמצעות מערך A.

1) for $1 \le i,j \le p$
   if $(A(i) \ge A(j))$
   $B(i,j) \leftarrow 1$
   else
   $B(i,j) \leftarrow 1$

2) for $1 \le i \le n$
   $M(i) = B(i,1) \wedge \dots \wedge B(i,n)$

בסוף הריצה יש ב- $M(i)$ 1 רק עבור i האינדקס של X_i המינימלי.

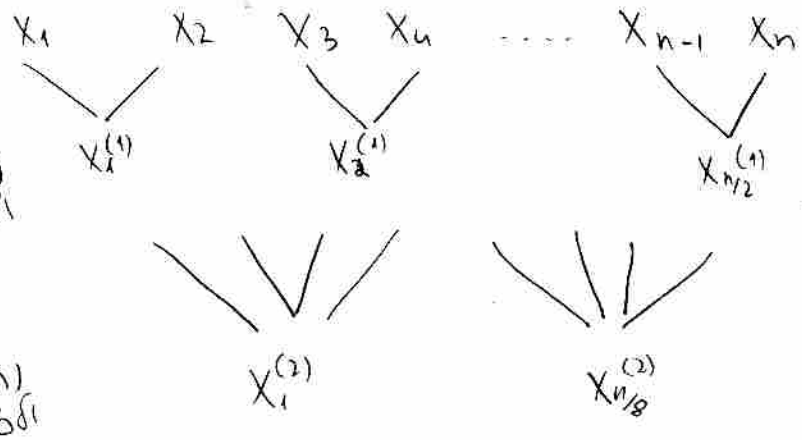
ניתוח

$$T(n) = O(1)$$

$$W(n) = O(n^2)$$

(96)

אלגוריתם מקסימום בזמן $O(\log\log n)$ מעלה $O(n\log\log n)$



נעלים אותם לבעלויות של 4 ומבצעים בקבוצה של 4 את אותה הפעולה

נעלים לבעלויות של 2 ומבצעים בקבוצה של 2 את אותה הפעולה

וכן הלאה.

מספר הפעולות מרמה ה- k שווה ל- $\frac{n}{2^{2^k}}$ לכן סך הכל
זמן $O(\log\log n)$ כי אנו עושים $O(1)$ פעולות
$$T(n) = O(\log\log n)$$
מכך שלמה השתמשנו ב- $O(n)$ מעבדים לכן
$$W(n) = O(n\log\log n)$$

מכך שלמה מספר המעבדים הוא n.

אלגוריתם אמיתי כזה פולינומיאלי, כלומר שגודלו מאוכלס.
השאלה כיום מה לה אורך הקלט?
אם קלט לבעיה הוא $a_1, \ldots, a_n$ ואמין כל שם "אלגוריתם" הוא $O(n)$ לא הוא פולינומיאלי, מאחר הפרמטרים הם מספר המספרים הוא $n+1$, ולא מספר ספרות t.

דוגמה - יש שאלות מתוך 5 ואתה חושב יותר קלים וחלקם יותר קשים.
מהשאלות יפות קדמות ניתן לדעת פחות מהפה.

הרעיון שעומד מאחורי האלגוריתם הוא קצת מסובך.

משפט השאריות הסיני

יהיו $n_1, \ldots, n_k$ מספרים טבעיים כלשהם כלומר $\gcd(n_i, n_j) = 1$ לכל $i \neq j$. אז לכל $a_1, \ldots, a_k$ כך ש $0 \leq a_i \leq n_i$ קיים x יחיד $0 \leq x \leq N = \prod_{i=1}^{k} n_i$ כך ש- $x \bmod n_i = a_i$ לכל i.
הוכחה יתנו אלגוריתם יעיל למציאת x.

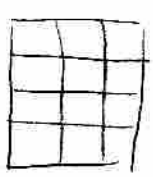
הוכחה: נמצא מספרים $c_1, \ldots, c_k$ כך ש- $0 \leq c_i \leq n$ כך ש-
$$c_i \bmod n_j = \begin{cases} 1 & i=j \\ 0 & i \neq j \end{cases}$$
מכיוון שעשינו את זה סיימנו כי אם נציב $x = \sum a_i c_i \bmod n$ אז הוא מקיים את מה שדרוש: $x \bmod n_i = \sum_{j=1}^{k} a_j c_j \bmod n_i = a_i c_i \bmod n_i = a_i \bmod n_i$.
איך נמצא c_i-ים באלגוריתם?
נגדיר $m_i = \prod_{j \neq i} n_j$ אזי לכל $i \neq j$ $m_i \bmod n_j = 0$
$\gcd(n_i, m_i) = 1 \Leftarrow \gcd(m_i \bmod n_i, n_i) = 1$ כי קיים b_i כך ש $b_i m_i \equiv 1 \pmod{n_i}$. נגדיר $c_i = b_i m_i \bmod n$.
קל לראות ש- $c_i \bmod n_j = 0$ לכל $i \neq j$. ומאידך $c_i \bmod n_i = 1$.
יחידות של הפתרון. נניח ש $x' \neq x$ כך ש $x \bmod n_i = a_i = x' \bmod n_i$
$\Leftarrow n_i \mid x - x' \Leftarrow x - x' \bmod n_i = 0 \Leftarrow x = x'$ ☺

שאלה יש מילים $w_1, \ldots, w_n$ כך ש $l_i = |w_i|$
צריך לסדר אותן על דף שרוחבו L כך שרווח המינימלי
בין כל שתי מילים בשורה יהיה כמה שיותר שווה
בדיוק תהיה מינימלית.

שלשורות i נגיד שהמחיר הוא c_i ונרצה למצוא את c_n.

$$c_i = \min_{1 \le j \le i} \left\{ c_{j-1} + (L - \sum_{k=j}^{i} l_k)^2 \right\}$$
$$L \ge \sum_{k=j}^{i} l_k$$

דוגמא הלוח: a, b, c, d



$$\Rightarrow c_1 = 4, \quad c_2 = 1 \quad c_3 = 0$$

$$c_4 = \min \left\{ \begin{matrix} 4 \\ j=2 \end{matrix}, \begin{matrix} 2 \\ j=3 \end{matrix}, \begin{matrix} 4 \\ j=4 \end{matrix} \right\} = 2$$

מכפלת פולינומים

f פולינום מעל $\mathbb{C}$ כך ש- $\deg f < 2^n$ (המקדמים השקדנים)
$f = \sum_{i=0}^{2^n-1} a_i z^i$

הפלט הוא $\{ f(\omega_{2^n}^j) \}_{0 \le j < 2^n}$.

FFT: אלגוריתם רקורסיבי. נגדיר f_e, f_o ע"י

$$f_e(z) = \sum_{i=0}^{2^{n-1}-1} a_{2i} z^i, \quad f_o(z) = \sum_{i=0}^{2^{n-1}-1} a_{2i+1} z^i$$

מריצים FFT על f_e, f_o, נותר לדעת $\{f_e(\omega_{2^{n-1}}^j)\}$, $\{f_o(\omega_{2^{n-1}}^j)\}$ $0 \le j < 2^{n-1}$

$$f(\omega_{2^n}^j) = \sum_{i=0}^{2^n-1} a_i (\omega_{2^n}^j)^i = f_e(\omega_{2^n}^{2j}) + \omega_{2^n}^j f_o(\omega_{2^n}^{2j}) = f_e(\omega_{2^{n-1}}^j) + \omega_{2^n}^j f_o(\omega_{2^{n-1}}^j)$$