אלגוריתמים חמדניים

<u>בעיית אופטימיזציה-</u> בהנתן קלט, יש מס' פתרונות חוקיים.

לכל פתרון חוקי יש ערך.

פלט: בבעיית מקסימיזציה הפתרון החוקי בעל הערך המקסימלי.

בבעיית מינימיזציה הפתרון החוקי בעל הערך המינימלי.

<u>אלגוריתמים חמדניים</u>

אלגוריתמים חמדניים לרוב פותרים בעיית אופטימיזציה

באינטואיציה אלגוריתם חמדני הוא אלגוריתם שבכל שלב מקבל את ההחלטה שנראית הכי טובה לפי המידע שיש כרגע.

<u>שלבים של אלגוריתם חמדני</u>

1. ממיינים קבוצת מועמדים מהטוב לגרוע.
2. מתחילים מפתרון ריק
3. עוברים על המועמדים לפי הסדר הממויין שלהם ובכל שלב מוסיפים את המועמד לפתרון אם הוספתו תשאיר את הפתרון חוקי.

<u>בעיית תרמיל הגב - *KnapSack*</u>

<u>קלט</u>-n חומרים/חפצים לכל חפץ i יש ערך $v\_i > 0$ ומשקל $w\_i > 0$ ונתון משקל סף W.

<u>פתרון חוקי</u> -וקטור $(x\_1, x\_2, \dots x\_n)$

כך שלכל i מתקיים $0 \leq x\_i \leq 1$

וכך ש $\sum_{i=1}^{n} x_i w_i \leq W$

<u>ערך</u>-לכל פתרון חוקי $X = (x\_1, x\_2, \dots x\_n)$ הערך הוא $V(X) = \sum_{i=1}^{n} x_i v_i$

<u>פלט</u> -פתרון חוקי בעל ערך מקסימלי .

<u>אלגוריתם חמדני לבעית תרמיל הגב</u>

מי יהיו המועמדים? שקי החומרים השונים.

<u>רעיונות לסידור המועמדים מהטוב לגרוע</u>

1. רווחיות יורדת .
2. משקל עולה .
3. $\frac{v_i}{w_i}$ יורד .

הסידור שעובד הוא <u>3</u>

<u>תובנות</u>

1. אם $\sum_{i=1}^{n} w_i \leq w$ אז הפתרון האופטימלי הוא לקחת הכל.
   נניח שהבעיה מעניינת, כלומר $\sum_{i=1}^{n} w_i > w$

2. לפתרון האופטימלי תמיד יהיה משקל w , כי תמיד נעדיף לקחת עוד מחומר כלשהו.

האלגוריתם הבא מקבל את החפצים אחרי שכבר מוינו:

```
KnapSack(w[1...n],v[1...n],w)
{
    for i <- 1...n
        x[i] <- 0
    weight <- 0
    i=1
    while(weight < w AND i <= n)
    {
        if (weight + w[i] <= w)
            x[i] <- 1
        else
            x[i] <- (w-weight)/w[i]

        weight <- weight +x[i]*w[i]
        i+1
    }
    return x
}
```

דוגמה

$$w = 100 \quad \{1,2,3,4,5\}$$

i	1	2	3	4	5
w_i	10	20	30	40	50
v_i	20	30	66	40	60
$\frac{v_i}{w_i}$	2	1.5	2.2	1	1.2

$\frac{v_i}{w_i}$ יורד - 3,1,2,5,4

$$\left(1,1,1,0,\frac{4}{5}\right)$$

$$1 \cdot 20 + 1 \cdot 30 + 1 \cdot 66 + \frac{4}{5} \cdot 60 = 156$$

סיבוכיות

מיון – $\Theta(nlogn)$

לולאה ראשונה $\Theta(n)$

לולאה שניה $\Theta(n)$

סה"כ $\Theta(nlogn)$

משפט

האלגוריתם מחזיר פתרון אופטימלי . (בעל הרווח המקסימלי מבין הפתרונות החוקיים)

הוכחה

יהי $x = (x_1, \dots x_n)$ הפתרון שהאלגוריתם מחזיר. נניח שהפריטים כבר ממוינים לפי $\frac{v_i}{w_i}$.

נסמן ב j את האנדקס הנמוך ביותר שעבורו $x_j < 1$.

כלומר לכל $i < j$ $x_i = 1$

לכל $i > j$ $x_i = 0$

נשים לב ש $\sum_{i=1}^{n} x_i w_i = w$ ונסמן ב $V(x)$ את הרווחים של x כלומר:

$$V(x) = \sum_{i=1}^{n} x_i v_i$$

נסמן ב $y = (y_1, \dots, y_n)$ פתרון חוקי כלשהו. לכן

$$\sum_{i=1}^{n} y_i w_i \le w$$

ומכך נובע

$$\sum_{i=1}^{n} x_i w_i - \sum_{i=1}^{n} y_i w_i = \sum_{i=1}^{n} (x_i - y_i) w_i \ge 0$$

נסמן ב $V(y) = \sum_{i=1}^{n} y_i v_i$

$$V(x) - V(y) = \sum_{i=1}^{n} (x_i - y_i) v_i = \sum_{i=1}^{n} \frac{(x_i - y_i) v_i}{w_i} \cdot w_i$$

אם $i < j$

$x_i - y_i \ge 0$ כי $x_i = 1$

$\frac{v_i}{w_i} \ge \frac{v_j}{w_j}$ בגלל שממויין לפי $\frac{v}{w}$ יורד

ולכן $(x_i - y_i)\frac{v_i}{w_i} \ge (x_i - y_i)\frac{v_j}{w_j}$

אם $i > j$

$x_i - y_i \le 0$ כי $x_i = 0$

$\frac{v_i}{w_i} \le \frac{v_j}{w_j}$ בגלל שממויין בסדר יורד

ולכן $(x_i - y_i)\frac{v_i}{w_i} \ge (x_i - y_i)\frac{v_j}{w_j}$.

$$V(x) - V(y) = \sum_{i=1}^{n} \frac{(x_i - y_i) v_i}{w_i} \cdot w_i \ge \sum_{i=1}^{n} \frac{(x_i - y_i) v_j}{w_j} \cdot w_i = \frac{v_j}{w_j} \sum_{i=1}^{n} (x_i - y_i) \cdot w_i \ge 0$$

לכן הרווחיות של x גדולה או שווה לרווחיות של כל פתרון חוקי, כנדרש.

<u>תזמון</u>

<u>א. תזמון ללא דד-ליין</u>

נתונות n משימות שלא יכולות להתבצע במקביל.

לכל משימה יש משך זמן ביצוע t_i.

נגדיר תזמון- סדר של המשימות.

עבור תזמון X לכל i נסמן ב $T_i\ (X)$ את משך הזמן שהמשימה i נמצאת במערכת- כלומר כמות הזמן שהמשימה חיכתה להתבצע +הזמן שהתבצעה (t_i).

נסמן ב T את משך הזמן הכללי של כל המשימות במערכת כלומר : $T(X)=\sum_{i=1}^{n} T_i(X)$

נרצה תזמון של כל המשימות שבו משך הזמן של כל המשימות במערכת הכי קטן.

<u>בעיית האופטימיזציה</u>

קלט- n משימות $\{1,\dots,n\}$ לכל משימה i נסמן ב t_i את משך הפעולה של המשימה.
המשימות לא יכולות להתבצע במקביל.

פתרון חוקי-תזמון שבו כל המשימות מתבצעות.

ערך-עבור פתרון חוקי X הערך הוא $T(X)$

פלט-פתרון חוקי בעל ערך מינימלי.

<u>דוגמה</u>

$t_1=5 \quad t_2=10 \quad t_3=3$
$\mathrm{T}(123)=5+15+18=38$
$T(132)=5+8+18=31$
$\mathrm{T}(213)=10+15+18=43$
$T(231)=10+13+18=41$
$\boldsymbol{T(312)=3+8+18=29}$ <====== הקצר ביותר!
$T(321)=3+13+18=34$

<u>האלגוריתם</u>

<u>1.</u> נמיין את המשימות לפי t_i עולה.
<u>2.</u> נחזיר תזמון שהסדר שלו הוא לפי סעיף 1.

<u>משפט</u>

האלגוריתם מחזיר תזמון אופטימלי.
נוכיח משפט חזק יותר:
בכל תזמון אופטימלי המשימות מסודרות לפי t_i עולה.

הוכחה

יהיו n משימות בעלי זמני ביצוע $t_1, t_2, \dots, t_n$ ויהיה $P = (p_1, \dots, p_n)$ תזמון אופטימלי.
נסמן ב $S_i = t_{p_i}$.

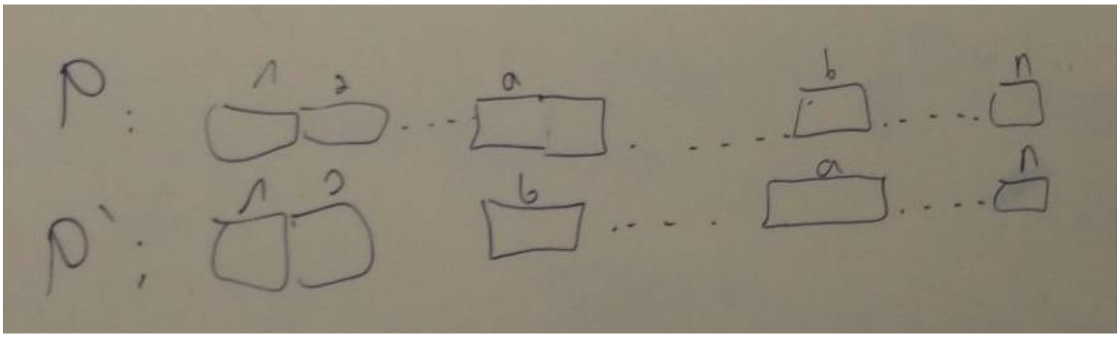
נסמן ב $T(P)$ את זמן השהיה הכללי של כל המשימות לפי התזמון P.

$$T(P) = S_1 + (S_1 + S_2) + (S_1 + S_2 + S_3) + \cdots + (S_1 + S_2 + S_3 + \cdots + S_n)$$

$$\mathrm{T(P)} = \sum_{i=1}^{n} (n - i + 1) S_i$$

נניח בשלילה ש P לא מסודר לפי S_i עולה.
כלומר קיימים a ו b כך ש $a < b$ אך $S_a > S_b$.

התזמון P' יהיה זהה לתזמון p חוץ מההחלפה של המשימות a ו b .



נשים לב:

$$T(P) = \sum_{i=1, i\neq a, i\neq b}^{n} (n - i + 1) S_i + (n - a + 1) S_a + (n - b + 1) S_b$$

$$T(P') = \sum_{i=1, i\neq a, i\neq b}^{n} (n - i + 1) S_i + (n - b + 1) S_a + (n - a + 1) S_b$$

$$\mathrm{T}(P) - T(P') = (n - a + 1) S_a + (n - b + 1) S_b - (n - b + 1) S_a - (n - a + 1) S_b$$

$$\begin{aligned} \mathrm{T}(P) - T(P') &= (n - a + 1 - n + b - 1) S_a + (n - b + 1 - n + a - 1) S_b \\ &= (b - a) S_a + (a - b) S_b = (b - a)(S_a - S_b) > 0 \end{aligned}$$

בסתירה לאופטימליות של P . ולכן כל תזמון אופטימלי מסודר לפי t_iעולה,כנדרש.