עבודת בית 3 בתכנון אלגוריתמים - תשובון

שם:	עידן ניצן	ת.ז.:	308564210
שם:	עדן רשף	ת.ז.:	305156531

שאלה 1

סעיף א': נסחו תת-בעיה אופיינית (הגדירו את OPT)

נסמן ב-$OPT(j)$ משקל פתרון חוקי מינימלי עבור j החוליות הראשונות, כאשר $b_1,...,b_j$ להן התקציב אשר הוקצה לאנטון בכל אחת מ-j החוליות הראשונות בשדרה.
נרצה למצוא את $OPT(n)$.

סעיף ב': נסחו נוסחת מבנה לחישוב OPT

$$OPT(j) = \begin{cases} \infty, & j<0 \\ 0, & j=0 \\ \min\limits_{1\le k\le 6} \{OPT(j-k)\} + b_j, & j\ge 1 \end{cases}$$

<u>סעיף ג': הוכיחו את נכונות נוסחת המבנה</u>

<u>ניתוח מקרים :</u>

נחלק את הפתרונות האפשריים עבור $\{1,...,j\}$ התנועות השונות לקבוצות. נגדיר 6 תת-קבוצות של פתרונות אפשריים, $S_1,...,S_6$, כאשר S_k $(1\le k\le 6)$ היא אוסף כל הפתרונות בהם התנועה ה-$j-k$ היא התנועה הבאה בתור לאחר התנועה ה-j.

<u>תת- הקבוצות הנ"ל מכסות את קבוצת כל הפתרונות האפשריים</u>

יהי u פתרון כלשהו לבעיה. אם $t_1=k$, כלומר בהלה הראשונה יצאו k $(1\le k\le 6)$, אזי $u\in S_k$.

<u>הסקת הצורה הסכמטית של נוסחת הרקורסיה</u>

נגדיר $cost_k(u)$ להיות משקל פתרון u עבור $j-k$ התנועות השונות. כמו כן, נגדיר $O^*(S_k)$ להיות המשקל המינימלי של פתרון ב-S_k, כלומר $O^*(S_k)=\min_{u\in S_k}\{cost_k(u)\}$. ניתן להסיק כי $OPT(j)=\min_{1\le k\le 6}\{O^*(S_k)\}$.

ניתוח האופטימום של כל קבוצה:

נוכיח כי לכל $1\le k\le 6$, $O^*(S_k)=b_j+OPT(j-k)$

כיוון 1: $\le$

יהי u פתרון עבור $j-k$ התחנות הראשונות כך ש $cost_k(u)=OPT(j-k)$. אזי הפתרון u^*, כו ערך התחנה הראשונה הוא k ושאר התחנות לפי u (בהזזת סדר) הוא פתרון עבור j התחנות הראשונות ב-S_k ועלותו $cost_0(u^*)=b_j+cost_k(u)$. לסיכום:

$$O^*(S_k)\le cost_0(u^*)=b_j+cost_k(u)=b_j+OPT(j-k)$$

מהגדרת O^*

כיוון 2: $\ge$

יהי $u^*\in S_k$ כך ש $O^*(S_k)=cost_0(u^*)$. נניח כי m = מספר התחנות ב-u^*, $|u^*|=m$. יהי u סדרת תחנות כך ש $|u|=m-1$, ולכל $1\le i\le m-1$ התחנה ה-i ב-u זהה לתחנה ה-$i-1$ ב-u^*. u הוא פתרון חוקי עבור $j-k$ התחנות הראשונות ומתקיים $cost_0(u^*)=b_j+cost_k(u)$. לכן:

$$O^*(S_k)=cost_0(u^*)=b_j+cost_k(u)\ge b_j+OPT(j-k)$$

מהגדרת OPT, $OPT(j-k)\le cost_k(u)$

□

סעיף ד': תארו אלגוריתם איטרטיבי לחישוב OPT

נאתחל מערך M בגודל n+1. לכל תא במערך נשמור סמך אשר j הוא חושב.

$M[j].Source$, $M[j].Value$. $M[0].Value \leftarrow 0$, $M[0].Source \leftarrow 0$

עבור $j=1$ עד 6 בצע: $M[j].Source \leftarrow 0$, $M[j].Value \leftarrow b_j$

בצע: עבור $j=7$ עד n בצע:

(א) $M[j].Value \leftarrow \infty$

(ב) עבור $k=1$ עד 6

(1) $q \leftarrow b_j + M[j-k].Value$

(2) אם $q < M[j].Value$

(I) $M[j].Value \leftarrow q$

(II) $M[j].Source \leftarrow j-k$

נתחו את זמן ריצת האלגוריתם

אתחול של M - $O(n)$

ריצת הלולאה -

לכל ערך יש מספר קבוע פעולות. יש $O(n)$ ערכים.

סה"כ זמן הריצה הוא $O(n)$.

הוכיחו את נכונות האלגוריתם

בכל שלב בעת חישוב ערכי $M[j]$ (עבור $7\le j\le n$), הערכים בהם משתמש האלג'
($M[j-k].Value$ עבור $1\le k\le 6$) כבר חושבו בשלבים קודמים,
עבור $1\le j\le 6$ אין צורך בחישוב ערכים קודמים.

טענה: עבור $0\le j\le n$, $M[j].Value=OPT(j)$. נוכיח באינדוקציה שלמה על j.

בסיס: $j=0$ - $OPT(0)=M[0]=0$ נובע מהגדרת $OPT(j)$.

$1\le j\le 6$ - $OPT(j)=b_j+\min_{1\le k\le 6}\{OPT(j-k)\}=b_j+OPT(0)=b_j$

הנ"א: עבור $l<j$, $M[l].Value=OPT(l)$. עבור j, $b_j\ge 0$

צעד: מתיאור האלג':

$$M[j].Value=\min_{1\le k\le 6}\{M[j-k].Value\}+b_j=\min_{1\le k\le 6}\{OPT(j-k)\}+b_j=OPT(j)$$

מה"א.   מנכונות נוסחת הנסיגה

$\square$

תארו אלגוריתם לשחזור הפתרון האופטימלי והסבירו את נכונותו בקצרה

איתחול: $Sol\leftarrow\langle\rangle$, $j\leftarrow n$

צעד: כל עוד $j>0$ בצע:

1. הוסף את $j-M[j].Source$ כקלט אחרון ברשימה Sol
2. $j\leftarrow M[j].Source$

החזר את Sol.

האלג' מסתיים משום שבכל $0<j$, $j>M[j].Source$ (מתיאור האלג' האיטרטיבי), ולכן בכל צעד j מקטין את ערכו. בנוסף, האלג' האיטרטיבי $M[j].Source$ מעדכן אל סוף ה-OPT המינימלי מבין 6 הערכים הקודמים ל-j ולכן, לפי נוסחת הנסיגה, לולאי ההגעה הכי קצרה על מנת לקבל פתרון חוקי באורך מינימלי. ערך הקלט הדרוש כדי לעבור מהתא j ל-$M[j].Source$ הוא $j-M[j].Source$.

<u>ניתוח זמן ריצה</u>

האלג' מבקר לכל היותר n כדורים.
כל כדור מבקר מספר קבוע של פעמים.
לכן זמן הריצה הכולל הוא $O(n)$.

שאלה 2

סעיף א': נסחו תת-בעיה אופיינית (הגדירו את OPT)

נגדיר $OPT(i,j,p)$ להיות מספר ההקשות המינימלי הדרוש לכתיבת המילה בה מופיעים התווים $c_i, c_{i+1}, \ldots, c_j$ כל אחד $f_i, f_{i+1}, \ldots, f_j$ פעמים בהתאמה, כאשר התווים מחולקים ל-p מקשים זרים המכסים את כל $j-i+1$ התווים הנ"ל תוך שמירה על יחס הסדר. נציין כי $1 \le i \le j \le n$ ו- $1 \le p \le 12$. אנו מחפשים את $OPT(1,n,12)$.

סעיף ב': נסחו נוסחת מבנה לחישוב OPT

$$OPT(i,j,p) = \begin{cases} f_i, & i=j \quad (a) \\ \sum_{k=i}^{j} (k-i+1) \cdot f_k, & p=1 \quad (b) \\ \min_{i \le k < j} \{OPT(i,k,1) + OPT(k+1,j,p-1)\}, & p>1,\ i<j \quad (c) \end{cases}$$

סעיף ג': הוכיחו את נכונות נוסחת המבנה

ניתוח מקרים :

הוכחת (a): $j=i$ ולכן יש אות אחת בלבד. במקרה הזה האות הראשונה במקש ולכן מס' הפעמים שיש להקיש הוא f_i.

הוכחת (b): יש מקש אחד בלבד ובו כל האותיות. עבור כל אות יש ללחוץ על המקש (מס' הפעמים שיש לכתוב את האות)*(מיקום האות).

הוכחת (c) לפי ההסכמה.

ניתוח מקרים: נחלק את הפתרונות ל- $j-i$ האפשריים קבוצות $S_i, S_{i+1}, \ldots, S_{j-1}$ כאשר S_k היא אוסף כל הפתרונות בהם המקש הראשון מורכב מהאותיות $c_i, c_{i+1}, \ldots, c_k$.

תת- הקבוצות הנ"ל מכסות את קבוצת כל הפתרונות האפשריים

ניקח פתרון כלשהו לבעיה, U שהוא חלוקה של $c_i,\dots,c_j$ ל-p מקטעים רציפים. נתבונן במקטע הראשון בחלוקה, נניח כי הוא מורכב מהאותיות $(c_i,\dots,c_k)$, אזי $U\in S_k$.

הסקת הצורה הסכמטית של נוסחת הרקורסיה

לכל פתרון $U\in S_k$ נגדיר את $cost(U)$ להיות מספר הקשות הדרוש לכתיבת ההודעה כך שיופיעו התווים $c_i,\dots,c_j$ לאחר $f_i,\dots,f_j$ פעמים בהתאמה, כאשר התווים מחולקים ל-p מקטעים לפי החלוקה המוגדרת בפתרון U. נגדיר את $O^*(S_k)$ להיות מס' הקשות המינימלי של הפתרונות בקבוצה S_k, כלומר $O^*(S_k)=\min_{U\in S_k}\{cost(U)\}$.

ניתן להסיק כי $OPT(i,j,p)=\min\{O^*(S_i),O^*(S_{i+1}),\dots,O^*(S_{j-1})\}$ לכל $2\le p\le$ בו.

ניתוח האופטימום של כל קבוצה:

ולכל $2\le p\le$ בו

נוכיח כי לכל $i\le k<j$ מתקיים $O^*(S_k)=OPT(i,k,1)+OPT(k+1,j,p-1)$.

כיוון 1: $\le$

תהי U חלוקה של האותיות $\{c_{k+1},\dots,c_j\}$ ל-$p-1$ מקטעים כך ש- $cost(U)=OPT(k+1,j,p-1)$, אזי החלוקה U^* של האותיות $\{c_i,\dots,c_j\}$ ל-p מקטעים כך שהאותיות $\{c_i,\dots,c_k\}$ מופיעות במקטע הראשון והאותיות $\{c_{k+1},\dots,c_j\}$ מחולקות ל-$p-1$ מקטעים לפי החלוקה U היא פתרון ב-S_k ועלותו $cost(U^*)=OPT(i,k,1)+cost(U)$

מהגדרת (b) בנוסחת הגדרה.

לסיכום: $O^*(S_k)\le cost(U^*)=OPT(i,k,1)+cost(U)=OPT(i,k,1)+OPT(k+1,j,p-1)$

מהגדרת O^*

כיוון כי: $\geq$

יהי $u^* \in S_k$ כך ש $O^*(S_k)=cost(u^*)$. כלומר, החיתוך הראשון

החלוקה המושרית ע"י u^* מכילה את $\{c_i,\dots,c_k\}$, תהי u

חלוקת השאריות $\{c_{k+1},\dots,c_j\}$ ל- $p-1$ חלקים ב- u^*.

עלות u^* היא $opt(i,k,1) + cost(u)$

ולכן

ⓑ מהגדרת

הנוסחה

הגדרה

$O^*(S_k)=cost(u^*)=opt(i,k,1)+cost(u)$.

לפי הגדרת $opt(i,j,p)$ מתקיים כי $opt(k+1,j,p-1)\leq cost(u)$

ולכן $O^*(S_k)\geq opt(i,k,1)+opt(k+1,j,p-1)$

□

סעיף ד': תארו אלגוריתם איטרטיבי לחישוב OPT

נאתחל מערך m בגודל n×n כאשר כל תא במערך מכיל 12 ערכים $p_{12}, \dots, p_2, p_1$.

```
for i<-1 to n do
    m[i,i].p1<-f_i, m[i,i].p2<-f_i, ..., m[i,i].p12<-f_i
for l<-1 to n-1 do
    for i<-1 to n-l do
        j<-i+l
        sum<-0
        for k<-i to j
            sum<-sum+(k-i+1)*f_k
        m[i,j].p1<-sum
        m[i,j].pr<-∞
        for k<-1 to j-1 do
            if (m[i,k].p1 + m[k+1,j].p(r-1) < m[i,j].pr)
                m[i,j].pr<-m[i,k].p1 + m[k+1,j].p(r-1)
return m
```

יש לבצע את שורות אלו לכל $2 \le r \le 12$

נתחו את זמן ריצת האלגוריתם והסבירו את נכונותו

ניתוח זמן ריצה:

- אתחול: אורך האלכסון הראשי הוא $O(n)$, כל תא מכיל מספר קבוע של ערכים. סה"כ $O(n)$.
- כל תא $[i,j]$ מצריך בדיקה של $j-i$ תאים בשורה ובעמודה, ובמקרה הכללי $O(n)$ פעולות לכל תא.
- סה"כ זמן מילוי הטבלה: $O(n^3)$.

הסבר נכונות האלג'

ניתן להוכיח באינדוקציה שבעת חישוב התא $m[i,j]$, כל 12 הערכים של התאים בשורה ה-i מהעמודה ה-i ועד העמודה ה-$j-1$ וכל התאים בעמודה ה-j מהשורה ה-$i+1$ ועד השורה ה-j כבר חושבו. כמו כן, מתיאור האלג', ניתן להגיד כי לכל i,j ולכל $2 \le r \le 12$, $m[i,j].p_r = OPT(i,j,r)$

שאלה 3

סעיף א'

קיים פתרון אופטימלי O עבורו קיימת G_i כך ש-

נניח בשלילה כי $p_i \neq \min\{x(c): c\in G_i\}$. אם $p_i > \min\{x(c): c\in G_i\}$ אזי

המיקום של הדוג"ה בעלת האינדקס הנמוך ביותר המשוייכת לקבוצה

לו בציר המספרים גדול מהמיקום ההתחלתי, בסתירה לכללים.

לכן בהכרח $p_i < \min\{x(c): c\in G_i\}$. נגדיר פתרון חדש O^* עם

חלוקה זהה לחלוקה ב-O, ולכל $1\le j\le k$ כך ש $j\neq i$,

$p^*_j = p_j$, אבל $p^*_i = \min\{x(c): c\in G_i\}(>p_i)$. אזי:

מיקום G_j ב-O^*

$$cost(O^*) = \sum_{i=1}^{k}\sum_{c\in G_i} w(c)\cdot(x(c)-p^*_i) < \sum_{i=1}^{k}\sum_{c\in G_i} w(c)\cdot(x(c)-p_i) = cost(O)$$

משום ש $p^*_i > p_i$

וכל שאר הפרמטרים זהים

בסתירה למינימליות של O.

□

סעיף ב'

נסמן את הפתרון האופטימלי בתור O. נניח בשלילה כי קיים

אינדקס j כך ש $m_i \le j \le M_i$ אבל $c_j \notin G_i$. נניח כי $c_j \in G_k$ $(k\neq i)$.

אם $p_k > p_i$ ניקח t כלשהו ב-G_i כך ש $j < t \le M_i$ (בהכרח

קיים כזה, אחרת $j = M_i$ ומכך נובע ש $c_j \in G_i$). נגדיר פתרון

O^* שבו לכל l $(l\neq i,k)$, $G^*_l = G_l$, אבל $G^*_i = G_i\setminus\{c_t\}$

ו $G^*_k = G_k\cup\{t\}$. זהו פתרון חוקי.

עדיין ב-O^*

משום ש $p_k \le j$ (כי $c_j \in G_k$) ולכן $t > p_k$.

$t-p_k < t-p_i$ ולכן $cost(O^*) < cost(O)$ בסתירה למינימליות O.

אם $p_k < p_i$ נגדיר פתרון O^* שבו לכל l $(l\neq i,k)$, $G^*_l = G_l$,

אבל $G^*_i = G_i\cup\{j\}$ ו $G^*_k = G_k\setminus\{j\}$. $p_i = m_i \le j$ ולכן זהו פתרון

חוקי. בנוסף, $j-p_k > j-p_i$ ולכן $cost(O^*) < cost(O)$

בסתירה למינימליות O.

□

סעיף ג': נסחו תת-בעיה אופיינית (הגדירו את OPT)

נגדיר את $OPT(j,p)$ להיות משקל פתרון חוקי מינימלי עבור חלוקה של העוגיות $\{c_1,\dots,c_j\}$ ל-p ערימות. אנו מחפשים את $OPT(n,k)$.

סעיף ד': נסחו נוסחת מבנה ונמקו בקצרה את נכונותה

לכל $1 \le i \le j \le n$ נגדיר: $sum(i,j)=\sum_{t=i}^{j} w(c_t)\cdot(x(c_t)-x(c_i))$

$$OPT(j,p)=\begin{cases}0, & j\le 1\\ 0, & p\ge j,\ j>1\\ sum(1,j), & j>1,\ p=1\\ \min_{1\le t<j}\{OPT(t,p-1)+sum(t+1,j)\}, & j>1,\ 1<p<j\end{cases}$$

מסעיפים א' ו-ב', הפתרון הוא למעשה חלוקה של $\{c_1,\dots,c_n\}$ כאשר כל קבוצה בחלוקה הינה רצף עוגיות. עלות הפתרון מורכבת מסכום של עלות היצירה של כל אחת מהקבוצות בחלוקה, כאשר עלות היצירה של קבוצה היא, לפי סעיף א', סכום עלויות ההעברה של כל עוגיה לנקודה בה נמצאת העוגיה הראשונה בקבוצה. כלומר, עלות היצירה של הערימה $\{c_i,\dots,c_j\}$ היא $sum(i,j)$. עלות פתרון מינימלי היא עלות יצירת הערימה העליונה ביותר + עלות חלוקה מינימלית של שאר העוגיות ל-p-1 ערימות. דורשים את הערימה העליונה כך שהעלות הכוללת תהיה מינימלית.

סעיף ה': תארו אלגוריתם איטראטיבי לחישוב OPT

נשתמש במערך דו-מימדי m בגודל n×n כאשר כל תא במערך מכיל 2 שדות: val ו- src.

```
for i←1 to n do
    for p←i to n do
        m[i,p].val←0 , m[i,p].src←0
```

נשתמש במערך דו-מימדי נוסף בגודל n×n בשם sum.

```
for i←1 to n do
  for j←1 to n do
    sum[i,j]←0
    for t←i to j do
      sum[i,j]←sum[i,j] + w(c_t)·(x(c_t) − x(c_i))
for i←2 to n do
  m[i,1].val←sum(1,i), m[i,1].src←0
המשך
for j←2 to n do
  for p←2 to j-1 do
    m[j,p].val←∞
    for t←1 to j-1
      q←m[t,p-1].val+sum[t+1,j]
      if q<m[j,p].val then
        m[j,p].val←q
        m[j,p].src←t
return m
```

נתחו את זמן ריצת האלגוריתם והסבירו את נכונות האלגוריתם

ניתוח זמן ריצה:

אתחול m - $O(n^2)$, אתחול sum - $O(n^2)$.
מילוי כל תא ב-sum לוקח $O(n)$ לכל היותר, לכן מילוי sum
דורש $O(n^3)$. מילוי כל תא שנמצא מתחת לאלכסון הראשי ב-m
דורש זמן קבוע - סה"כ $O(n^2)$. העמודה ואופן מילוי השורה
העליונה (m[i,1]) דורש $O(n)$. מילוי כל תא ב-m שנמצא
מעל האלכסון הראשי דורש בדיקה של j-1 תאים שכבר חושבו,
ובמקרה הגרוע $O(n)$ פעולות לכל תא. סה"כ מילוי m דורש
$O(n^3)$. לסיכום זמן הריצה הוא $O(n^2+n^2+n^3+n^2+n+n^3) = \boxed{O(n^3)}$

הסבר נכונות:

ניתן להוכיח באינדוקציה כי לכל j,p, בעת חישוב התא
m[j,p] כל התאים עליהם הוא מסתמך (m[i,p-1], 1≤i≤j-1) כבר חושבו.
בנוסף, כל המערך sum כבר חושב גם הוא. כמו כן, מתיאור
האלג', ניתן לראות כי לכל j,p, m[j,p] = OPT(j,p).

סעיף ו': תארו אלגוריתם לשחזור הפתרון האופטימלי ונתחו את זמן ריצתו

```
O←∅
if n≤k then
  for i←1 to n
    G_i←{c_i}, O←O∪{G_i}
  for i←n+1 to k
    G_i←∅, O←O∪{G_i}
```

```
else
    curr ← n
    for i ← k to 1: G_i ← ∅
        for j ← curr to (m[curr, i].src + 1)
            G_i ← G_i ∪ {c_j}
        O ← O ∪ {G_i}
        curr ← m[curr, i].src
return O
```

ניתוח זמן ריצה:

אם תנאי, יוצרים את קבוצות, שלוקח זמן קבוע, סה"כ $O(k)$.

אחרת, יצירת קבוצה G_i דורשת $O(|G_i|)$. סה"כ

יצירת כל הקבוצות דורשת $O(\sum|G_i|) = O(n)$

סה"כ זמן הריצה הוא $O(n+k)$.