עבודת בית 2: תכנון אלגוריתמים 2021

תאריך הגשה: 22.11.20, 23:59. את העבודה יש להגיש במערכת ההגשה (עדיף מוקלד).

* מומלץ ביותר ***לא להמתין לרגע האחרון*** להגשת העבודה.

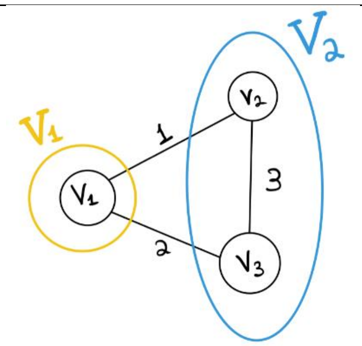
מתרגלים אחראיים: שחף פיינדר ועירית שלי

הוראות כלליות:

- כל עוד לא נאמר אחרת, כאשר הנכם מתבקשים לתאר אלגוריתם יש לספק את הבאות:
  1. תיאור מילולי של האלגוריתם.
  2. הוכחת נכונות.
  3. ניתוח זמן ריצה
- אלגוריתם עם זמן ריצה אקספוננציאלי לא נחשב יעיל ולכן בדרך כלל לא יתקבל.
- יש לרשום פתרון בדף התשובות הנלווה לעבודה.
- לתשומת לבכם, המקום שהוקצה בדף התשובות הינו פי 1.5 מהמקום המומלץ.

שאלה 1

הצעה 1



עבור הגרף G, האלגוריתם בחר לחלק את הצמתים לשתי קבוצות $V_1 = \{v_1\},\ \ V_2 = \{v_2, v_3\}$.

לפי האלגוריתם אנו נחבר בעזרת הצלע המינימאלית את קבוצות הקודקודים. כלומר, הצלע שתיבחר הינה $< v_1, v_2 >$ בעלת משקל 1.

וקיבלנו כי העץ מורכב מהצלעות $< v_1, v_2 >, < v_2, v_3 >$ ויהיה בעל משקל 4.

לעומת זאת אם נוריד את הצלע $< v_2, v_3 >$ ונחבר את הצלע $< v_1, v_3 >$ נקבל עץ פורש בעל משקל 3.

זו סתירה למינימאליות העץ הפורש, ולכן האלגוריתם אינו תקין.

הצעה 2

נוכיח. ננסח שתי טענות ונקרא לאלגוריתם המתואר קרוסקל מקוצר.
קיים MST כך שהגרף המוחזר מפרים לאחר $\left\lfloor\frac{\|V\|}{2}\right\rfloor$ איטרציות מוכל בו קרוסקל מקוצר מחזיר MST
הערה – קרוסקל מקוצר בוחר מתוך כל הצלעות (C) האפשריות שלא נבחרו בשלב שפרים פעל
1: נראה כי בכל שלב בפרים קיים MST שמכיל את כל בחירות האלגוריתם ולכן בפרט לאחר $\|V\|-1 \leq \left\lfloor\frac{\|V\|}{2}\right\rfloor$
באינדוקציה על מספר הקשתות שהאלגוריתם בחר (שמוכלות ב-B)
בסיס: |B|=0 ולכן קבוצה ריקה מוכלת בכל MST. נניח כי ל-$\|B\| = i-1$ קיים T MST שמכיל את B.
צעד: נניח כי קיים עץ $T=(V,F)$ שמכיל את $i-1$ הקשתות הראשונות שהאלגוריתם בחר, ותהי.$e=(v,w)$
הקשת ה- i שהאלגוריתם בחר. אם $e \in F$ אז סיימנו. אחרת ב-T יש מסלול מ-v ל-w ונסמנו $v = v_1, \dots v_l = w$
יהי j אינדקס כך ש-$v_j \in S$ ו-$v_{j+1} \notin S$ ונשים לב כי $v \in S$ ו- $w \notin S$. האלגוריתם בחר בקשת e ולא בקשת

$e' = (v_j, v_{j+1})$ ולכן $w(v,w) \le w(v_j, v_{j+1})$. נתבונן ב- $T' = (V, F \cup \{e\} \setminus \{e'\})$. מטענה שהוכחה בכיתה T’

הוא עץ פורש וגם $w(T') = w(T) + w(e) - w(e') \le w(T)$ ולכן T’ הוא עפ"מ המכיל את $B \cup \{e\}$ כנדרש.

2: נוכיח קודם כי בכל שלב באלגו' קיים MST כך שאוסף הצלעות שנבחרו מוכל בו. נראה באינ' על גודל B.

בסיס: B היא בחירת הצלעות של פרים בעל $\left\lfloor \frac{|V|}{2} \right\rfloor$ איטרציות ומטענה 1 קיים MST כך ש-B מוכלת בו.

הנחה: נניח כי באיטרציה ה-(i-1) קיים T MST כך B מוכלת בו.

צעד: תהי $e = (v, w)$ הקשת שהתווספה ל-B באיטרציה ה-i. נחלק למקרים אם $e \in T$ אז סיימנו. אחרת $e \notin T$

נתבונן ב-$H = (T \cup \{e\})$, H מכיל מעגל ו-B לא מכילה מעגל. לכן קיימת קשת e’ במגעל ב- H שלא נמצאת בB.

הקשת e’ שייכת ל-T שמכיל את B ולכן B לא מכילה מעגל. האלגורתם בחר קשת על משקל מנמאלי שלא סוגרת מעגל

ולכן $w(e) \le w(e')$. נסתכל על $T' = T \setminus \{e'\} \cup \{e\}$ מטענה שהוכחנו בכיתה T’ הוא עף פורש.

לכן $w(T') = w(T) + w(e) - w(e') \le w(T)$ כלומר T’ הוא עפ"מ ומכיל את כל הקשתות ב-B.

נראה כי האלגוריתם קרוסקל מקוצר מחזיר עפ"מ.

אם האלגוריתם מסתיים אז הוא מחזר גרף חסר מעגלים שמכיל $|V| - 1$ קשתות. לכן הוא מחזיר עף פורש שלפי טענת

העזר מוכל בעפ"מ ולכן מוחזר עפ"מ

נראה שהאלגוריתם מסתיים כלומר משלים את הגרף המתקבל מפרים ל-$|V| - 1$ צלעות:

נניח כי בשלב מסויים $|B| < |V| - 1$ ונראה כי אז $|C| > 0$.

לפי טענת העזר קיים עץ פורש T המכיל את קשתות B. מכיוון שב-T יש $|V| - 1$, קיימת קשת e המופיעה ב-T

אבל לא ב-B. אז $e \in C$ ולכן $C \neq \emptyset$.

יהי j אינדקס כך ש-$v_j \in S$ ו-$v_{j+1} \notin S$ ונשים לב כי $v \in S$ ו- $w \notin S$. האלגוריתם בחר בקשת e ולא בקשת

$e' = (v_j, v_{j+1})$ ולכן $w(v,w) \le w(v_j, v_{j+1})$. נתבונן ב- $T' = (V, F \cup \{e\} \setminus \{e'\})$. מטענה שהוכחה בכיתה T’

הוא עץ פורש וגם $w(T') = w(T) + w(e) - w(e') \le w(T)$ ולכן T’ הוא עפ"מ המכיל את $B \cup \{e\}$ כנדרש.

2: נוכיח קודם כי בכל שלב באלגו' קיים MST כך שאוסף הצלעות שנבחרו מוכל בו. נראה באינ' על גודל B.

בסיס: B היא בחירת הצלעות של פרים בעל $\left\lfloor \frac{|V|}{2} \right\rfloor$ איטרציות ומטענה 1 קיים MST כך ש-B מוכלת בו.

הנחה: נניח כי באיטרציה ה-(i-1) קיים T MST כך B מוכלת בו.

צעד: תהי $e = (v, w)$ הקשת שהתווספה ל-B באיטרציה ה-i. נחלק למקרים אם $e \in T$ אז סיימנו. אחרת $e \notin T$

נתבונן ב-$H = (T \cup \{e\})$, H מכיל מעגל ו-B לא מכילה מעגל. לכן קיימת קשת e’ במגעל ב- H שלא נמצאת בB.

הקשת e’ שייכת ל-T שמכיל את B ולכן B לא מכילה מעגל. האלגורתם בחר קשת על משקל מנמאלי שלא סוגרת מעגל

ולכן $w(e) \le w(e')$. נסתכל על $T' = T \setminus \{e'\} \cup \{e\}$ מטענה שהוכחנו בכיתה T’ הוא עף פורש.

שאלה 2

סעיף א'

תת בעיה: לכל $v \in V$ נגדיר תת בעיה (v) להיות בעיית מציאת המסלול מs לv בעל ההפרש המינימאלי בין הצלעות
הצהובות לירוקות.
$Sol(v)$: קבוצת כל המסלולים מs לv, כלומר $I \in Sol(v)$ אם: $I = \{(s, u_1, u_2, \dots, u_m, v) \mid <u_i, u_{i+1}> \in E\}$
$OPT(i, F, j)$: המסלול מs ל v בעל ההפרש המינימאלי, כלומר: $OPT(v) = \min\limits_{I \in Sol(v)} \{diff(I)\}$
כאשר הפונקציה מוגדרת: $diff(I) = \sum_{<u_i, u_{i+1}> \in I} c(<u_i, u_{i+1}>)$

סעיף ב'

עבור קבוצת השכנים U עם קשתות נכנסות של קודקוד v (כלומר, $U = \{u \mid <u, v> \in E\}$) נחלק את $Sol(v)$
באופן הבא ל $\lvert U \rvert$ תתי קבוצות: $Sol(v) = \bigcup_{u \in U} \{T \cup <u, v> \mid T \in Sol(u)\}$ נוכיח כי הקבוצות שקולות:
$\subseteq$: יהי מסלול $I \in Sol(v)$, לכן $I = \{(s, u_1, u_2, \dots, u_m, v) \mid <u_i, u_{i+1}> \in E\}$. נסמן:
$T = \{s, u_1, u_2, \dots, u_m\}$ לכן, $I = T \cup \{<u_m, v>\}$, לכן: $I \in \bigcup_{u \in U} \{T \cup \{<u, v>\} \mid T \in Sol(u)\}$.
$\supseteq$: יהי מסלול $T \in Sol(u)$, לכן $T = \{s, u_1, u_2, \dots, u_m, u) \mid <u_i, u_{i+1}> \in E\}$,
ומכיוון ש$u \in U$ אז: $E \ni <u, v>$ ולכן $T \cup <u, m> \in Sol(v)$

סעיף ג'

<u>נוסחת המבנה:</u>
עבור קבוצת השכנים $U = \{u \mid <u, v> \in E\}$:
$OPT(v) = \begin{cases} 0 & , v = s \\ \min\limits_{u \in U} \{OPT(u) + diff(\{<u, v>\})\} & , v \neq s \end{cases}$

סעיף ד'

<u>הוכחת נכונות נוסחת המבנה:</u>
נראה כי בשני המקרים של נוסחאות המבנה, $OPT(v)$ שווה לאגף ימין של הנוסחה.
<u>מקרה א':</u> $v = s$: במקרה זה $Sol(v) = \emptyset$ ולכן הפתרון הוא יחיד, ומשקל ההפרש שלו הוא 0.
<u>מקרה ב':</u> $v \neq s$:
<u>אבחנה 1</u>: $diff(u_1, u_2, \dots u_{m-1}, u_m) = diff(u_1, u_2, \dots u_{m-1}) + diff(u_{m-1}, u_m)$

<u>טענת עזר</u>: אם עבור $I' = (v_1, v_2, \dots, v_m)$ מתקיים $I' = \min_{I\in Sol(v)}\{diff(I)\}$ אזי לכל $1 \le i \le j \le m$ מתקיים:
$diff(v_i, v_{i+1}, \dots, v_{j-1}, v_j) = \min\{diff(v_i, u_1, u_2, \dots, u_{j-i}, v_j) \mid <u_k, u_{k+1}>\in E\}$
<u>הוכחת טענת עזר</u>:
נניח בשלילה כי $diff(v_i, v_{i+1}, \dots, v_{j-1}, v_j)$ לא המינימאלי, לכן קיים מסלול P מ v_i ל v_j עבורו מתקיים:
$diff(v_i, v_{i+1}, \dots, v_{j-1}, v_j) \ge diff(P)$, אך כעת נקבל מאבחנה 1 כי:
$diff(v_1, \dots, v_{i-1}, P, v_{j+1}, \dots, v_m) \le diff(v_1, v_2, \dots, v_m)$ וזו סתירה למינימליות I.
<u>טענה</u>:
$\min_{I\in Sol(v)}\{diff(I)\} \overset{\text{טענה}}{\hat{=}} \min_{u\in U}\left\{\min_{T\in Sol(u)}\{diff(T)\} + diff(\{<u,v>\})\right\}$
<u>הוכחת הטענה</u>:
עבור מסלול $I' = \{s, u_1, u_2 \dots, u, v\}$ כך ש $diff(I') = \min_{I\in Sol(v)}\{diff(I)\}$.
מטענת העזר, עבור מסלול $T = \{s, u_1, u_2 \dots, u\}$ נקבל כי $diff(T') = \min_{T\in Sol(u)}\{diff(T)\}$
מאבחנה 1 נקבל כי $diff(T') + diff\{(<u,v>)\} = diff(I')$
ולכן, מסעיף ב' וממינימאליות מסלול I' , נקבל כי
$\min_{u\in U}\left\{\min_{T\in Sol(u)}\{diff(T)\} + diff(\{<u,v>\})\right\} = \min_{I\in Sol(v)}\{diff(I)\}$
<u>כעת מציאת השיוויון</u>: עבור קבוצת השכנים $U = \{u \mid <u,v>\in E\}$:
$OPT(v) \overset{\text{הגדרה}}{\hat{=}} \min_{I\in Sol(v)}\{diff(I)\} = \overset{\text{טענה}}{\hat{=}} \min_{u\in U}\left\{\min_{T\in Sol(u)}\{diff(T)\} + diff(\{<u,v>\})\right\}$
$\overset{\text{הגדרה}}{\hat{=}} \min_{u\in U}\{OPT(u) + diff(\{<u,v>\})\}$

סעיף ה'

<u>אלגוריתם</u>:
נגדיר מערך V ומערך M, כל אחד כגודל מספר הקודקודים n.
<u>אתחול</u>:
נקבע אינדקס $i \leftarrow 1$
נבצע מעבר DFS על הגרף, כאשר ברגע סיום עבודה על קודקוד v (כלומר סיום מעבר על כל קבוצת שכינו) בצע:
◦ $V[i] = v$
◦ $i \leftarrow i++$
נקבע $M[1] \leftarrow 0$
עבור $i = 2$ עד n בצע:
◦ $M[i] \leftarrow \infty$
<u>צעד</u>:

עבור $i=1$ עד n בצע:
∘ עבור כל $u\in U$ כך ש: $U=\{u\mid <u,V[i]>\in E\}$ בצע:
∘ אם $M[u]+diff(\{<u,V[i]>\}\le M[V[i]]$
∘ קבע $M[V[i]]\leftarrow M[u]+diff(\{<u,V[i]>\}$
<u>החזר</u>: מערך M.

סעיף ו'

<u>נכונות האלגוריתם:</u>
נשים לב, כי מאופן ביצוע DFS, נסיים לעבוד עבור כל קודקוד בגרף בזמן שונה, ולכן, כל i ייצג קודקוד שונה ב $V[i]$.
בנוסף, נשים לב כי מכיוון שאנו מתחילים את המעבר על הקודקודים מקודקוד s, יובטח כי $s=V[1]$.
באיטרציה הi, אנו בודקים עבור הקודקוד $V[i]$ כי עבור כל אחד משכניו u עם קשת נכנסת האם המסלול מs העובר בשכן זה
ומסתיים ב$V[i]$ הינו המינימאלי. בהסתמך על נוסחת המבנה שהוכחנו קודם.
מאופן ביצוע DFS אנו יודעים כי עבור כל שכן u שכזה, כבר מובטח שמשקל המסלול המינימאלי אילו נקבע והוא
לא ישתנה גם בהמשך (זאת מכיוון שאין מעגלים בגרף).
בנוסף, מכיוון שנתון כי קיים מסלול מs אל כל קודקוד בגרף, לאחר סיום מעבר הDFS בהכרח עבור כל קודקוד $V[i]$ נקבל כי
$M\big[V[i]\big]\neq\infty$
<u>זמן ריצה:</u>
DFS: חסום על ידי $O(\lvert V\rvert+\lvert E\rvert)$
אתחול המערך M לערכי אינסוף: חסום על ידי $O(\lvert V\rvert)$
צעד: חסום על ידי $O(\lvert V\rvert+\lvert E\rvert)$
לכן, האלגוריתם חסום על ידי $\boldsymbol{O(\lvert V\rvert+\lvert E\rvert)}$.

<u>שאלה 3</u>

סעיף א'

תת בעיה: לכל $1\le i\le n$ ולכל $1\le j\le m$ נגדיר את תת הבעיה ה(i,j,dir) להיות זהה לבעיה המקורית פרט
לכך שנקודת ההתחלה של השליח היא בנקודה $[j,i]$ כאשר הוא הגיע מכיוון dir.
$\boldsymbol{Sol(i,j,dir)}$**:** קבוצת כל המסלולים המתחילים בכיוון dir מנקודה $[j,i]$ ומסתיימים בנקודה $[n,m]$.
כלומר $I\in Sol(i,j,dir)$ אם"ם $I\subseteq\{<([j,i],\dots,[n,m]),dir>\mid the\ path\ is\ legal\}$
$\boldsymbol{OPT(i,j,dir)}$**:** העלות המינימאלית מבין המסלולים שב $Sol(i,j,dir)$. כלומר:
$OPT(i,j,dir)=\min\limits_{k\in sol(i,j,dir)}\{g(k)\}$

סעיף ב'

נתבונן ב$Sol(i,j,dir)$, נחלק אותו לשתי קבוצות, הראשונה מכילה את כל המסלולים שכיוון תנועתם בצומת הראשון

אינו משתנה. השנייה מוגדרת כך שכיוון המסלול משתנה בצומת הראשון.

אבחנה: כל מסלול, בכל שלב קיימת שתי אפשרויות שינוי כיוון ושמירה על הכיוון לכן קבוצות אלה זרות ואיחודן הוא

$Sol(i,j,dir)$

סעיף ג'

<u>נוסחת המבנה:</u>

נגדיר $g\ (< x,y >,< r,v >,dir)$ המחזירה את ערך המעבר בצומת $c[y,x]$ בהתאם להאם התבצעה פניה או לא

$$OPT(i,j,dir) = \begin{cases} 0 & ,i = n, j = m \\ g(< i,j >,< i+1,j >,dir) + \sum_{k=j+1}^{m-1} c[k,i] & ,i = m, j \neq n \\ g(< i,j >,< i,j+1 >,dir) + \sum_{k=i+1}^{n-1} c[j,k] & ,i \neq m, j = n \\ \min\{OPT(i+1,j,V) + g(< i,j >,< i+1,j,dir), \\ \quad OPT(i,j+1,H) + g(< i,j >,< i,j+1,dir)\} & ,else \end{cases}$$

V מסמן כיון כניסה אנכי, ו- H כיוון כניסה אופקי

סעיף ד'

<u>הוכחת נכונות נוסחת המבנה:</u>

ישנם 3 מקרי בסיס:

1. מציאת עלות מסלול מהנקודה הנוכחת לנקודה הנוכחית – עלות 0
2. אם הגענו לעמודה הימנית ביותר של המטריצה אין אפשרות למסלול אחר פרט לתנועה אנכית ולכן נסכום את
   עלות המעבר בצומת הנוכחי (בתלות אם בצענו פניה) ואת עלות המעבר בכל שאר הצמתים במסלול פרט לאחרון
3. אם הגענו לשורה התחתונה של המטריצה אין אפשרות למסלול אחר פרט לתנועה אופקית ולכן נסכום את
   עלות המעבר בצומת הנוכחי (בתלות אם בצענו פניה) ואת עלות המעבר בכל שאר הצמתים במסלול פרט לאחרון

פרט למקרי הבסיס, בכל שלב בו נרצה להקטין את גודל הבעיה נבחין בין שני מקרים: שינוי כיוון ואי שינוי כיוון.

נחפש את המינימום מבין הפתרונות האופטימלי על תזוזה ימינה ותזוזה מטה (שמייצגות שינוי או אי-שינוי כיוון) ולכן

מהם נוסיף את עלות המעבר בצומת הנוכחי.

סעיף ה'

<u>אלגוריתם לחישוב ערכי OPT:</u>

נגדיר מערך דו-ממדי M בגודל זהה למטריצה. בכל תא במערך זה ישמרו שני ערכים $< x,y >$ כך ש-x ייצג את עלות המסלול האופטימלי של מעבר בנקודה כך שכיוון ההגעה הוא אנכי. Y באופן דומה לאופקי.

צעד: עבור j מ-n עד 1 בצע:

עבור i מ-m עד 1 בצע:

אם $i=m$ וגם $j=n$: $M[n][m] \leftarrow < 0{,}0 >$

אחרת אם $j=n$: $M[i][j].x \leftarrow M[i][j].y + 2 * c[j,i]$

$M[i][j].y \leftarrow M[i+1][j] + C[j,i]$

אחרת אם $i=m$: $M[i][j].y \leftarrow M[i][j+1].y + 2 * C[j,i]$

$M[i][j].x \leftarrow M[i][j+1].x + C[j,i]$

אחרת: $M[i][j].y \leftarrow \min\{M[i][j+1].x + 2\cdot C[j,i] \quad , \quad M[i+1][j].y + C[j,i]\}$

$M[i][j].x \leftarrow min(M[i+1][j].x + C[j,i] \quad , M[i+1][j].y + 2\cdot C[j,i])$

החזר את M[i][j].y

סעיף ו'

הסבר נכונות האלגוריתם:
לכל תא במטריצה באלגוריתם האיטרטיבי נשמרים שני ערכים x שמציין את עלות המסלול המינימלי מהנקודה ללקוח בהגעה
מכיוון אנכי ובאופן דומה לערך y בהגעה מכיוון אופקי.
חישוב תאי המטריצה מבוצע בסדר כזה שבכל חישוב עלות תא (פרט למקרי הבסיס) התאים הסמוכים (מימין ומלמטה לתא) כבר
מחזיקים את ערך המסלולים האופטימליים כך שניתן לחשב את המסלול האופטימלי בתלות שינוי הכיוון בצומת.
מקרי הבסיס דואגים לבחור את הכיוון האפשרי היחיד במידה ואין בחירה. (כך שכל המסלולים יסתיימו בנקודה הנדרשת).
4 שומר את הערכים להגעה אנכית ואופקית מפני שבניית המסלול מהסוף לא יודע מאיזה כיוון הגענו בפתרון האופטימלי המחושב.
זמן ריצה:
האלגוריתם עובר בכל תא של המטרציה הדו-ממדית פעם אחד בלבד ומבצע כל לתא מספר קבוע של פעולות. לכן זמן הריצה של
האלגוריתם האיטרטיבי הוא $\boldsymbol{O(n\cdot m)}$.

שאלה 4

סעיף א'

תת בעיה: לכל $n \le i \le 1$ ולכל $m \le j \le 1$ נגדיר תת בעיה (i,j,F_j) להיות בעיה של מציאת מספר פעולות
הזזת קופסאות מינימאלי מקומה i ומטה, כך שנותרו j קומות אפסון לבחור. ועם F_j קבוצת הקומות שלא מאופסנות עדיין
$\boldsymbol{Sol(i,j,F_j)}$: היא קבוצת כל הפתרונות החוקיים עבור תת הבעיה ה(i,j,F_j), כלומר $R \in Sol(i,j,F_j)$
אם"ם: $R = \{(l_1,\dots,l_j) \mid \forall_{1\le k\le j}: 1 \le l_k \le i,\ \forall_{I\in F_j}: I \ge l_j\}$
$\boldsymbol{OPT(i,j,F_j)}$: הוא עלות העבודה המינימאלית של פתרון חוקי עבור תת הבעיה (i,j,F_j) , כלומר:
$OPT(i,j,F_j) = \min\limits_{R\in Sol(i,j,F_j)}\{p(R)\}$ כאשר: $P(R) = \sum_{k=1}^{j}\sum_{I\in F_k} c_I \cdot (I - l_k)$

סעיף ב'

נחלק את הקבוצה $Sol(i,j,F_j)$ ל2 תתי קבוצות:
1) כל הפתרונות $R \in Sol(i,j,F_j)$ בהן $i \in R$, כלומר: הקומה (הגבוהה ביותר בתת-בעיה) i היא קומת אפסון.
2) כל הפתרונות $R \in Sol(i,j,F_j)$ בהן $i \notin R$, כלומר: הקומה i היא איננה קומת אפסון.
נשים לב כי הקבוצות זרות ואיחודן מהווה את כל קבוצת הפתרונות $Sol(i,j,F_j)$.
כלומר, נקבל:
$Sol(i,j,F_j) = Sol(i-1,j,F_j) \cup Sol(i-1,j-1,j-1)$

סעיף ג'

<u>נוסחת המבנה:</u>

$$OPT(i,j,F_j)=\begin{cases}0 & ,i=j\\ OPT(i-1,j,F_j\cup\{i\})+\sum_{I\in F_j}c_I & ,i>j \wedge j=1\\ \min\left\{OPT(i-1,j,F_j\cup\{i\})+\sum_{I\in F_j}c_I\,,OPT(i-1,j-1,F_{j-1})\right\} & ,i>j \wedge j\neq 1\end{cases}$$

סעיף ד'

<u>הוכחת נכונות נוסחת המבנה:</u>

לכל אחד משלושת המקרים בנוסחאות המבנה ננתח את $OPT(i,j,F_j)$ ונראה שהוא שווה לאגף ימין של הנוסחה:

<u>אבחנה:</u> לא ייתכן כי $i < j$ מכיוון שלא יהיו יותר קומות אפסון מאשר מספר הקומות עצמן.

<u>מקרה א':</u> $i=j$

במקרה זה נקבע כי כל קומה היא קומת אפסון, לכן יידרשו 0 העברות ארגזים, וזהו יהיה הפתרון האופטימלי.

<u>מקרה ב':</u> $i>j \wedge j=1$

אבחנה 1: קבוצת הפתרונות ב$Sol(i,j,F_j)$ בהן i אינה חלק מהפתרון היא בדיוק :

$Sol(i-1,j,F_j\cup\{i\})+\sum_{I\in F_j}c_I$, מכיוון שהקומה iה אינה קומת אפסון, נשלם על הורדת הארגזים מקומות F_j קומה אחת נוספת לקומה $i-1$ ונשאר עדיין עם j קומות אפסון לחלק.

המשך מקרה ב': מכיוון ש $j=1$ אז יש מחסן יחיד. ומכיוון שאי אפשר להעלות ארגזים לקומות אפסון מקומות נמוכות יותר, ומכיוון ש $i>j=1$, (כלומר ישנן קומות נמוכות יותר מi), לא נוכל לבחור בקומה זו כקומת אפסון, ונחייב להוריד את ארגזי קומה i, צעד זה יעלה להוריד כל אחד מקבוצות הארגזים קומה נוספת. כלומר: $\sum_{I\in F_j}c_I$.

ולכן, נקבל מסעיף ב', הפתרון $Sol(i,j,F_j)$ זהה לפתרון $Sol(i-1,j,F_j)+\sum_{I\in F_j}c_I$. ולכן:

$$OPT(i,j,F_j)\overset{\text{הגדרה}}{\triangleq} min_{R\in Sol(i,j,F_j)}\{p(R)\}\overset{\text{אבחנה 1}}{\triangleq} min_{R\in Sol(i-1,j,F_j)}\{p(R)\}+\sum_{I\in F_j}c_I$$

$$\overset{\text{הגדרה}}{\triangleq} OPT(i-1,j,F_j)+\sum_{I\in F_j}c_I$$

<u>מקרה ג':</u> $i>j \wedge j\neq 1$

אבחנה 2: קבוצת הפתרונות ב$Sol(i,j,F_j)$ בהן i חלק מהפתרון, היא בדיוק: $Sol(i-1,j-1,F_{j-1})$,

מכיוון שהקומה iה קומת אפסון, לא נעביר אף ארגז מטה, וכעת נעבוד על קבוצת הקומות של המחסן הבא: F_{j-1}.
המשך מקרה ג':
$OPT(i,j,F_j) \overset{\text{הגדרה}}{\triangleq} \min_{R\in Sol(i,j,F_j)}\{P(R)\} \overset{\text{סעיף ב'}}{\triangleq} \min\left\{\min_{\substack{R\in Sol(i,g)\\ i\notin R}}\{P(R)\}+c_i\ ,\ \min_{\substack{R\in Sol(i,g)\\ i\in R}}\{P(R)\}\right\} \overset{\text{אבחנה 2,1}}{\triangleq}$
$= \min\left\{\min_{R\in Sol(i-1,g)}\{P(R)\}+c_i\ ,\ \min_{R\in Sol(i-1,g-1)}\{P(R)\}\right\} \overset{\text{הגדרה}}{\triangleq} \min\{OPT(i-1,g)+c_i, OPT(i-1,g-1)\}$

סעיף ה'

<u>אלגוריתם לחישוב ערך OPT:</u>
נקבל מערך $C=[c_1,c_2,\dots,c_n]$ עם משקלי קומות ומספר m של קומות אפסון, ונגדיר מטריצה $M[i,j,k]$
כאשר i מסמן את הקומה, j מסמן את מס המחסנים, וk מסמן את מספר החדרים בF_j. נשים לב $k\in\{0,1,\dots,n-i\}$
אתחול:
עבור $j=1$ עד m בצע: //מקרה בסיס א'
∘ עבור $i=1$ עד j בצע:
∘∘ עבור $k=0$ עד $n-i$ בצע:
∘∘∘ $M[i,j,k]\leftarrow 0$
צעד:
עבור $i=2$ עד n בצע: //מקרה ב'
∘ עבור $k=0$ עד $n-i$ בצע:
∘∘ $M[i,1,k]\leftarrow M[i-1,1,k+1]+\sum_{l=i}^{i+k}C[l]$
עבור $j=2$ עד m בצע: //מקרה ג'
∘ עבור $i=j+1$ עד n בצע:
∘∘ עבור $k=0$ עד $n-i$ בצע:
∘∘∘ נקבע $M[i,j,k]\leftarrow M[i-1,j-1,k+1]$
∘∘∘ אם $M[i,j,k]>M[i-1,j,k+1]+\sum_{l=i}^{i+k}C[l]$
∘∘∘∘ אז $M[i,j,k]\leftarrow M[i-1,j,k+1]+\sum_{l=i}^{i+k}C[l]$
החזר: $M[n,m,0]$

סעיף ו'

<u>אלגוריתם לשחזור פתרון אופטימאלי:</u>
נקבל מטריצה $M[i,j,k]$ ונגדיר מערך L בגודל m.
נאתחל עבור כל $j = 1$ עד m: $L[j] = 0$
נקבע $j \leftarrow m, i \leftarrow n$
עבור $j = m$ עד 1
∘ נקבע $count \leftarrow 0$
∘ כל עוד $M[i,j,0] < M[i - count, j - 1, count]$ בצע:
∘∘ $count + +$
∘ נקבע $L[j] \leftarrow (i - count + 1)$
∘ נקבע $i \leftarrow (i - count)$
<u>זמן ריצה:</u>
אנו מבצעים איטרציה עם אינדקס j מm עד 1, כאשר האינדקס i מציין את הקומה אשר לה אנו משווים את הפתרון
האופטימלי עם הקומה האחרונה הידועה שהיא קומת אפסון.
אנו נעבור איטרציה מj ל$j - 1$ רק כאשר הקומה $i + 1$ היא קומת אפסון. וכל עוד היא לא, האינדקס i בכל צעד יורד,
קומה. אם $i = j$ נקבל כי כל קומה היא מחסן ונתקדם באיטרציה על אינדקס j עד שנגיע לקומה הראשונה ונעצור.
כלומר זמן הריצה הוא כמספר הקומות. ולכן: זמן הריצה הוא: $\boldsymbol{O(n)}$

בהצלחה!