עבודת בית 3 בתכנון אלגוריתמים

תאריך הגשה , 28.03.2011 בשעה 12:00 בצהריים, תא מספר 68 בקומת כניסה של בניין 37.

שאלה 1

סעיף א':

הגדירו מהו ערך פתרון אופטימאלי ע"י שימוש במונח OPT

OPT(n,m) – שכן אנחנו רוצים את מספר פעולות העריכה המינימאלי בין כל P וכל T.
נסמן: $T[1:j]$ הרישא באורך j של T : $T[1:j] = < t_1, t_2 \dots, t_j >$. **כנ"ל** לגבי P.
נסמן $Sol_{i,j}$ – מרחב **כל** הפתרונות לתת הבעיה של מרחק עריכה בין *T[1:j]* ו-*P[1:i]*

סעיף ב':

מקרי בסיס ו-$P_i = T_j$

יהי $U \in Sol_{i,j}$ פתרון שהוא גם אופטימלי. הפעלת פעולות העריכה של U על T[1:j-1] יתן לנו את P' = P[1:i-1] אבל $P_i = T_j$ לכן הפעלת U על T[1:j] ישנה רק את T[1:j-1] ויתן לנו את P[1:i]. נניח בשלילה שקיים פתרון U' עבור T[1:j] ו-P[1:i]
כך ש- $Cost(U') < Cost(U)$. נסתכל על הפעולה $e_k \in U'$ הראשונה שלאחר שהפעלנו אותה על T[1:j] קיבלנו את P[1:i-1] (מפעילים את הפעולות לפי סדר וכל פעם משנים את המחרוזת). בהכרח $k \le Cost(U) = |U|$. נסתכל על $U'' = < e_1, e_2 \dots e_k >$ זהו פתרון עבור תת הבעיה P[1:i-1] ו-T[1:j-1] שמקיים (לפי ההנחה בשלילה) ש-$Cost(U'') < Cost(U)$ **בסתירה** לכך ש-U פתרון אופטימלי.

אם i=0 , אורך המחרוזת P הוא 0 ואז צריך רק למחוק את כל התוים מ-T.
אם j=0 , T הינה המחרוזת הריקה ופשוט צריך להעתיק את P.

הגדרת תתי קבוצות של פתרונות אפשריים לפי חלוקה למקרים

בכל המקרים הכוונה לפתרון עבור מרחק עריכה בין רישא באורך i של P ורישא באורך j של P.
נסתכל על המקרה ש-$P_i != T_j$
נגדיר שלוש קבוצות של מקרים, בהתאם לפעולת העריכה **האחרונה** בפתרון.
יהי U רצף פעולות עריכה כפי שהוגדר בשאלה. נגדיר:

$$S_{insert} = \{U : U = < e_1 \dots e_k > and\ e_k = Insert(P_i, j)\}$$
$$S_{Delete} = \{U : U = < e_1 \dots e_k > and\ e_k = Delete(j)\}$$
$$S_{Replace} = \{U : U = < e_1 \dots e_k > and\ e_k = Replace(P_i, j)\}$$

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

אם $P_i != T_j$ אז בהכרח חייבים לבצע פעולת עריכה על T_j כדי להפוך את T ל-P. לכן כל פתרון U עבור מקרה זה יקיים לפי הגדרה ש-$U \in S_o : o \in \{Insert, Delete, Replace\}$.

הסקת הצורה הסכמטית של נוסחת הרקורסיה

נגדיר לכל $S_o : o \in \{Insert, Delete, Replace\}$ את $O*(s_o) = \min_{U \in S_o}\{Cost(U)\}$ כך ש-$Cost(U) = \|U\|$.   לכן מהגדרת OPT(i,j) להיות מרחק עריכה מינימאלי (Cost(U) מינימאלי) בין $P' = < p_1, .., p_i >$ ו– $T' =< t_1, \dots t_j >$ וכל פתרון U מקיים שהוא שייך לקבוצת S_o כלשהי, נסיק ש – $OPT(i,j) = \min\{O*(s_{Insert}), O*(S_{Delete}), O*(S_{Replace})\}$

ניתוח כל תת קבוצות פתרונות בנפרד והוכחת המרכיבים המתאימים בנוסחת המבנה:

טענה: $\boldsymbol{O*(S_{Insert}) = OPT(i-1,j) + 1}$ כיוון $\le$ נסמן $P' = P[1:i-1]$. אז OPT(i-1,j) הינו מרחק העריכה **המינימאלי** P[1,i-1] ו-T[1:j] נסמן ב-$U* = < e_1, \dots e_k >$ את הפתרון כך ש-$Cost(U*) = OPT(i-1,j)$. הפעלת פעולות העריכה של U* על T[1:j] תיתן לנו את P'. לכן כדי לקבל את P[1:i] נגדיר את $U** =< e_1, e_2, \dots e_k, \boldsymbol{e_{k+1}} >$ כך ש-$e_{k+1} = Insert(P_i, j)$ וע"י הפעלת רצף העריכה U** על T[1:j] נקבל את P[1:i]. קיבלנו $Cost(U**) = Cost(U*) + 1$ ו-$U** \in S_{Insert}$ ולכן Cost(U**) הוא עלות של פתרון ב-S_{Insert} ולכן בהכרח מתקיים האי שוויון , מש"ל.   כיוון $\ge$ יהי $U* \in S_{Insert}$ פתרון כך ש-$\boldsymbol{Cost(U*) = O*(S_{Insert})}$ נסמן $U* =< e_1, e_2 \dots e_k >$ ולפי ההגדרה של U* אז $e_k = Insert(P_i, j)$. נגדיר $U*' =< e_1, e_2 \dots e_{k-1} >$. מכיוון ש-U* ממיר את T[1:j] ל-P[1:i] והפעולה האחרונה היא הכנסה של P_i אז בהכרח U*' ממיר את T[1:j] ל-P[1:i-1] לכן בהכרח $Cost(U*') \ge OPT(i-1,j)$ מכאן   נקבל ש-$Cost(U*') + 1 \ge OPT(i-1,j)$ ו-$Cost(U*') = Cost(U*) + 1$ מכאן נובע ש-$O*(S_{Insert}) = Cost(U*) \ge OPT(i-1,j) + 1$ מש"ל.   **המקרה** $\boldsymbol{OPT(i,j-1) + 1}$ סימטרי ולא נוכיח אותו. ההוכחה תהיה זהה.   טענה: $\boldsymbol{O*(S_{Replace}) = OPT(i-1,j-1) + 1}$ כיוון $\le$ נסמן $P' = P[1:i-1]$ ו-$T' = T[1:j-1]$ יהי $U* =< e_1, e_2, \dots . e_k >$, $U* \in Sol_{i-1,j-1}$ פתרון כך ש-$C(U*) = OPT(i-1,j-1)$. U* פתרון לתת בעיה T' ו-P'. הפעלת $U*$ על T' יתן לנו את P'. לכן $U** = U* \circ (e_{k+1} = Replace(P_i, j))$ הוא פתרון עבור T[1:j] ו-P[1:i] שפעולת העריכה האחרונה שלו היא החלפה לכן $U** \in S_{Replace}$ לכן $Cost(U**) \ge O*(S_{Replace})$ נקבל לכן $O*(S_{Replace}) \le Cost(U**) = Cost(U*) + 1 = OPT(i-1,j-1) + 1$ מש"ל.   כיוון $\ge$ נסמן $U* \in S_{Replace} : Cost(U*) = O*(S_{Replace})$ לכן $U* \in Sol_{i,j}$. נסמן $U* =< e_1, e_2 \dots e_k >$ נגדיר $U** =< e_1, e_2, \dots e_{k-1} >$. *U** פתרון שממיר את *T[1:j]* ל-*P[1:i]* שפעולת העריכה האחרונה שלו היא $e_k = Replace(P_i, j)$. לכן בעת הפעלת e_k ואחרי הפעלת שאר הפעולות כבר קיבלנו את $P' = P[1:i-1]$ ע"י המרת *T[1:j-1]*. לכן $U**$ הוא פתרון **כלשהו** לתת בעיה *T[1:j-1]* ו-*P[1:i-1]*.

לכן $1 + OPT(i-1, j-1) \geq Cost(U **) + 1 = Cost(U *) = O * (S_{Replace})$. מש"ל.

-

שאלה 2

סעיף א':

הגדירו את תתי הבעיות באופן מילולי והגדירו את OPT

יהי P פתרון לבעיה. P הינו מסלול פשוט בין x ל-y. **נסמן** $S_v = \{P : P \text{ is a simple path from } x \text{ to } v\}$
$P =< e_1 = < x, v_1 >, e_2 = < v_2, v_3 > \cdots, e_k = < v_{k-1}, v_k = y \gg$. **נגדיר** עלות פתרון $C(P) = \min_{e_i \in P}\{w(e_i)\}$. **נגדיר** $OPT(v) = \max_{P \in S_v}\{C(P)\}$

סעיף ב':

נסחו את נוסחת המבנה, כולל מקרי בסיס **ומיקום הפתרון לבעיה המקורית**

יהי $(v_1, v_2, \dots . v_i = x, \dots . v_{j-1}, v_j = y, \dots . v_n)$ מיון טופולוגי של קודקודי הגרף. מעתה התייחסות לקודקוד v_k כלשהו הינה לקודקוד ה-k-י במיון הטופולוגי.תמיד $v_i = x$
$$OPT(v_j) = \begin{cases} if\ j = i\ then\ \infty \\ if\ j > i\ then & \max_{<v_t, v_j> \in E\ and\ i \le t < j}\{\min\{OPT(v_t), w(< v_t, v_j >)\}\} \\ if\ j < i\ then (-\infty) \end{cases}$$

סעיף ג': הוכיחו את נוסחת המבנה שניסחתם בסעיף ב'

הגדרת תתי קבוצות של פתרונות אפשריים לפי חלוקה למקרים

לתת בעיה של מסלול בין $x = v_i$ לקודקוד כלשהו $v_t : t \ge i$ (האינדקסים הם לפי המיון הטופולוגי) נסמן $S_t = \{P : P =< v_i = x, \dots . v_t, v_j = y > and\ \forall k! = s\ (v_k! = v_s)\}$.
נגדיר $Sol = \ \cup_{i \le t < j} S_t$
(הבהרה: בעצם אמרנו ש-
-קבוצת הפתרונות (מסלולים פשוטים בין x ל-y) שהקשת האחרונה בהם היא $(\boldsymbol{S_t} = \ < v_t, v_j = y >$.

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

יהיה P פתרון – מסלול פשוט בין x ל-y. $P =< v_i = x, \dots v_t, v_j = y >$ - אינדקסים לפי המיון הטופולוגי. לכל $i \le t < j$ נקבל $P \in Sol$ <= $P \in S_t$. אם $t > j$ אז v_t מופיע אחרי y במיון ולכן P מעגל ולכן P לא פתרון. אם $t < i$ אז v_t מופיע לפני x במיון ולכן שוב P מעגל ולא פתרון חוקי. לכן לכל פתרון **חוקי** P מתקיים $P \in Sol$ ולכן כיסינו את כל הפתרונות.

הסקת הצורה הסכמטית של נוסחת המבנה

נגדיר $O * (S_t) = \max_{P \in S_t}\{C(P)\}$. מכיוון שלכל פתרון P מתקיים $P \in Sol$
אז $OPT(v_j) = \max_{P \in Sol}\{P\} = \max_{i \le t < j}\{O * (S_t)\}$

<u>ניתוח כל תת קבוצות פתרונות בנפרד והוכחת המרכיבים המתאימים בנוסחת המבנה:</u>

צריך להראות $O*(S_t) = \min\{\mathrm{OPT}(\mathrm{v_t}), \mathrm{w}(<\mathrm{v_t},\mathrm{v_j}>)\}$.

1) $\geq$: יהיה P פתרון לתת בעיה של מסלול בין x ל-v_t ו-$C(P) = OPT(v_t)$. מכיוון ש-P מסלול פשוט ו-G חסר מעגלים אז גם $P' = P \circ < v_t, v_j = y >$ מסלול פשוט בין x ל-y שהקשת האחרונה בו היא $< v_t, v_j >$ לכן $P' \in S_t$ לכן $O*(S_t) \geq C(P')$
לכן $O*(S_t) \geq C(P') = \min\{C(P) = OPT(v_t), w(< v_t, v_j >)\}$ מש"ל (1).

2) $\leq$ יהי $P* \in S_t : C(P*) = O*(S_t)$ פתרון (מסלול פשוט בין x ל-y).
נגדיר $P' = P*\backslash < v_t, v_j >$ (הורדנו קשת אחרונה). P מסלול פשוט ורק הורדנו קשת לכן גם P' מסלול פשוט בין x ל-v_t לכן $C(P') \leq OPT(v_t)$
וגם $O*(S_t) = C(P*) = \min\{C(P') \leq OPT(v_t), w(< v_t, v_j >)$.
אם $C(P*) = C(P')$ נקבל $O*(S_t) \leq OPT(v_t)$ אם $C(P*) = w(< v_t, v_j >)$ אז $OPT(v_t) \geq C(P') \geq w(< v_t, v_j >$ ובכל מקרה נקבל $O*(S_t) \leq OPT(v_t)$. מש"ל

<u>סעיף ד':</u>

<u>נסחו אלגוריתם **איטראטיבי** למציאת **ערך** פתרון אופטימאלי</u>

1) יהי $< v_1, v_2, \ldots v_i = x , \ldots . v_j = y, \ldots v_n >$ מיון טופולוגי של קודקודי הגרף.
2) אתחל מערך $M[1..n]$. כך ש- $M[i] = -\infty$ ולכל התאים $M[t].\pi = Nil$
3) אתחול $\forall (< v_i = x, v_t > \in E)\ M[t] = w(< v_i, v_t >)$.
4) עבור $i \le k < j$ בצע:
   a. יהי v_k הקודקוד הנשקל כעת
   b. לכל $i \le t < k$ וגם $< v_t, v_k = y > \in E$ בצע:
      i. $val = \min\{M[t], w(< v_t, v_k >)\}$
      ii. אם $val \ge M[k]$ אז $M[k] = val$
      iii. $M[k].\pi = t$

<u>הוכחת אלגוריתם איטרטיבי (חישוב נכון של המערך M):</u>

אנחנו סורקים לפי סדר המיון הטופולוגי, לכן בעת חישוב ערך M[k] אנחנו מתבססים על M[t] ו-t מופיע לפני k במיון, לכן כבר M[t] חושב.
נניח באינדוקציה שלכל $i \le t < k$ $M[t] = OPT(v_t)$. צוואר הבקבוק יהיה במסלול בין x ל-v_t או הקשת האחרונה $< v_t, v_k > \in E$. מהגדרת צוואר בקבקוק
נקבל $OPT(v_k) = \max_{< v_t, v_k > \in E}\{\min\{OPT(v_t), w < v_t, v_k >\}\}$ ו- $OPT(v_t) = M[t]$

<u>זמן ריצה:</u>

שלב 1- מיון טופולוגי – O(|V| + |E|)
שלב 2 -3 , סריקה של הקודקודים – O(V).
שלב 4 – יש לשים לב שבגלל המיון הטופולוגי אנחנו עוברים על כל צלע פעם אחת. לכל בשלב 4 אנחנו עבורים לכל היותר על כל הצלעות לכן שלב זה הוא O(|E|).
לכן נקבל סה"כ O(|V| + |E|)

<u>סעיף ה':</u>

כתבו אלגוריתם המשחזר את הפתרון.

מלא את M בעזרת האלגוריתם מסעיף ד.
1) יהי *P* הפתרון $P = <>$
2) $c = j$
3) כל עוד $c\ != \infty$ או *c != i*
   a. $d = M[c].\pi$
   b. $P = < v_d, v_c > \circ P$
   c. $c = d$
4) אם *c = i* החזר *P*.
5) אם $c = \infty$ החזר אין מסלול.

שאלה 3

סעיף א':

הגדירו את תתי הבעיות באופן מילולי והגדירו את OPT

יהיה U פתרון <A,B> = U פתרון חוקי כפי שהוגדר בשאלה.
נגדיר $C(U) = \sum_{i\in A} a_i + \sum_{i\in B} b_i$. נגדיר
$Sol = \{U : U \text{ valid solution for } \{1..i\} \text{ items, and bags } T_1 \text{ and } T_2\}$
נגדיר $\mathrm{OPT}(i, T_1, T_2) = \max_{U\in Sol}\{\mathrm{C(U)}\}$.
במילים **ערך** הפתרון האופטימלי לתת הבעיה של פריטים {1..i} ומשקל שקים T_1 ו-T_2

סעיף ב':

נסחו את נוסחת המבנה, כולל מקרי בסיס **ומיקום הפתרון לבעיה המקורית**

$$OPT(i,T_1,T_2) = \begin{cases} if\ w_i \le T_1, T_2\ then \\ \qquad \max\{a_i + OPT(i-1, T_1 - w_i, T_2), \\ \qquad\quad b_i + OPT(i-1, T_1, T_2 - w_i), \\ \qquad\quad OPT(i-1, T_1, T_2) \\ if\ T_1 < 0\ or\ T_2 < 0\ \ then\ (-\infty) \\ if\ T_1 = 0\ and\ T_2 = 0\ then\ 0 \\ if\ i = 0\ then\ \ 0 \end{cases}$$

סעיף ג': הוכיחו את נוסחת המבנה שניסחתם בסעיף ב'

התייחסות למקרי בסיס וקצה

1) אם $T_1 < 0$ (בה"כ , אותו דבר לגבי T_2) זה אומר שקודם ביצענו קריאה ל- $OPT(i-1, T_1 - w_i, T_2)$ ו-$T_1 < w_i$ זה אומר שהפתרון שיחזור לא חוקי. במצב הזה נחזיר $-\infty$ ונגרום לכך שהערך שיחזור לעולם לא יבחר בתוך ה-max ולא יוכל להוות פתרון.
אם $T_1 = T_2 = 0$ אין לנו מקום להכניס שום דבר לכן בהכרח לא נוכל לבחור פריטים ונחזיר 0. אם i= 0 זה אומר שאין פריטים לבחור וגם אז הפתרון הכי טוב הוא 0.

הגדרת תתי קבוצות של פתרונות אפשריים לפי חלוקה למקרים

יהי U פתרון כפי שהוגדר בסעיף א לתת בעיה של פריטים {1..i} ומשקל שקים T_1, T_2.
נגדיר:
1) $S_A = \{U =< A, B >: i \in A \text{ and } i \notin B\}$
2) $S_B = \{U =< A, B >: i \notin A \text{ and } i \in B\}$
3) $S_\otimes = \{U =< A, B >: i \notin A \text{ and } i \notin B\}$

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

יהי U פתרון **חוקי** כלשהו לתת בעיה, $U =< A, B >$
1) אם $i \in A$ אז $U \in S_A$
2) אם $i \in B$ אז $U \in S_B$
3) אם $i \notin A \text{ and } i \notin B$ אז $U \in S_\otimes$.
בכל מקרה U נמצא באחת הקבוצות שהגדרנו.

הסקת הצורה הסכמטית של נוסחת המבנה

נסתכל על תת הבעיה {1..i} ומשקל שקים T_1, T_2. נגדיר:

1) $O*(S_A) = \max_{U \in S_A}\{C(U)\}$
2) $O*(S_B) = \max_{U \in S_B}\{C(U)\}$
3) $O*(S_\otimes) = \max_{U \in S_\otimes}\{C(U)\}$

לכן מהגדרת OPT נקבל $OPT(i,T_1,T_2) = \max\{O*(S_A), O*(S_B), O*(S_\otimes)\}$

<u>ניתוח כל תת קבוצות פתרונות בנפרד והוכחת המרכיבים המתאימים בנוסחת המבנה</u>

1) נראה $O*(S_A) = a_i + OPT(i-1, T_1 - w_i, T_2)$.

$\geq$ יהי $U* = <A,B>$ פתרון חוקי כך ש-$C(U*) = OPT(i-1,T_1-w_i,T_2)$. $U*$ פתרון חוקי לכן $w_i \leq T_1$. נגדיר $U' =<A',B>$ ו-$A' = A \cup \{i\}$ לכן $U' \in S_A$ (סכום המשקלים של $U*$ קטן שווה ל-$T_1 - w_i$ לכן סכום המשקלים של U' קטן שווה T_1 ולכן U' פתרון חוקי). לכן $O*(S_A) \geq C(U') = OPT(i-1,T_1-w_i,T_2) + a_i$.

$\leq$ נבחר $U* \in S_A$ כך ש-$C(U*) = O*(S_A)$. $U* = <A,B>$.
נגדיר $A' = A \ \{i\}$, $U** = <A',B>$ מכיוון ש-$U*$ פתרון חוקי גם $U**$ פתרון חוקי (הורדנו פריט, לכן סכום המשקלים רק קטן). U** פתרון חוקי לתת בעיה {1..i-1}.
לכן נקבל $O*(S_A) = C(U*) = C(U**) + a_i \leq OPT(i-1,T_1-w_i,) + a_i$.
מש"ל

המקרה עבור S_B זהה לחלוטין.

2) נראה $O*(S_\otimes) = OPT(i-1,T_1,T_2)$

$\geq$ יהי $U* = <A,B>$ פתרון לתת בעיה {1..i-1} כך ש-$C(U*) = OPT(i-1,T_1,T_2)$ לכן $i \notin A, i \notin B$. לכן $U* \in S_\otimes$ לכן $O*(S_\otimes) \geq C(U*) = OPT(i-1,T_2,T_2)$
מש"ל.

$\leq$ יהי $U* = <A,B>$ כך ש- $C(U*) = O*(S_\otimes)$ ו-$U* \in S_\otimes$ לכן $i \notin A, i \notin B$. לכן $U*$ גם פתרון לתת בעיה {1…i-1} לכן $O*(S_\otimes) = C(U*) \leq OPT(i-1,T_1,T_2)$.
מש"ל.

<u>סעיף ד':</u>

<u>נסחו אלגוריתם **איטראטיבי** למציאת **ערך** פתרון אופטימאלי</u>

נגדיר מערך תלת מימדי $M[n,W_1,W_2]$. אתחל הכול ל-0. וגם $M[*,*,*].\pi = nil$

```
1. for i : 1 to n do
2.   for T1: 1 to W1 do
3.     for T1 : 1 to W2 do
4.          a, b, c = (−∞)
5.          if wi ≤ T1: a = ai + M[i − 1, T1 − wi, T2]
6.          if wi ≤ T2: b = bi + M[i − 1, T1, T2 − wi]
7.          c = M[i − 1, T1, T2]
8.          ind = (i, T1, T2)
9.          if a ≥ b ≥ c : M[ind].π = (i − 1, T1 − wi, T2), M[ind] = a , M[ind].S = A
10.         else if b ≥ a ≥ c : M[ind].π = (i − 1, T1, T2 − wi), M[ind] = b, M[ind].S = B
11.         else if c ≥ a ≥ b : M[ind].π = (i − 1, T1, T2), M[ind] = c, M[ind].S = nil
12. return M
```

<u>הוכחת אלגוריתם איטרטיבי (חישוב נכון של המערך M):</u>

אנחנו סורקים את כל האינדקסים בכל מימד של המערך מהקטן לגדול ותמיד ניגשים לאינדקסים שהם יותר קטנים מכל הערכים הנוכחיים של האיטרטורים לכן ערכים אלו כבר חושבו באיטרציה קודמת. נשים לב ששורות 5-10 שקולות בעצם ל-$M[i,T_1,T_2] = \max\{a_i + M[i-1], T_1 - w_i, T_2], b_i + M[i-1,T_1,T_2-w_i], M[i1,T_1,T_2]\}$
וניתן להוכיח אינדוקטיבית מפה ש-$M[i,T_1,T_2] = OPT(i,T_1,T_2)$.

<u>זמן ריצה:</u>

הלולאות 1-3 עבורות על כל התאים במערך לכן, ובכל איטרציה אנחנו מבצעים O(1) פעולות. לכן סה"כ נקבל $O(n^3)$.

<u>סעיף ה':</u>

כתבו אלגוריתם המשחזר את הפתרון (<u>אם שילבתם את זה בסעיף ד', אז אין צורך למלא את הסעיף הזה</u>)

1. $index = (n, W_1, W_2), A =<>, B =<>$
2. $While\ M[index].\pi\ != nil$
3. $\quad if\ M[index].S == A\ then\ A.add(index.i)$
4. $\quad else\ if\ M[index].S == B\ then\ \ B.add(index.i)$
5. $\quad current = M[current].\pi$

שאלה 4

סעיף א':

הגדירו את תתי הבעיות באופן מילולי והגדירו את OPT

נגדיר Sol_i - מרחב הפתרונות לתת הבעיה של שליחת טיסות $\{1..i\}$
נסמן U פתרון חוקי לתת בעיה (אם $U \in Sol_k$) אז U פתרון לתת בעיה של שליחת טיסות $\{1 \dots k\}$.
נגדיר $OPT(i) = \max_{U \in Sol_i}\{C(U)\}$
נגדיר $C(U) = \sum_{i \in U} p_i$
נניח שהזמנים ממוינים מהקטן לגדול, שכן למיין אותם ולאחר מכן לעדכן את האינדקסים של המחירים יהיה $O(nlogn)$ ולא יפגע בזמן הריצה הסופי.

סעיף ב':

נסחו את נוסחת המבנה, כולל מקרי בסיס **ומיקום הפתרון לבעיה המקורית**

מיקום הפתרון OPT(n)
נגדיר: לכל i את a(i) = j להיות האינדקס j **המקסימאלי** כך ש-t_i $t_j + m \le t_i$
נגדיר: $OPT(i) = \max\{p_i + OPT(a(i)), OPT(i-1)\}$

סעיף ג': הוכיחו את נוסחת המבנה שניסחתם בסעיף ב'

הגדרת תתי קבוצות של פתרונות אפשריים לפי חלוקה למקרים

יהי $U \in Sol_i$ **לכל** $\boldsymbol{Sol_i}$
נגדיר: $S_{in} = \{U : t_i \in U\}$
נגדיר $S_{out} = \{U : t_i \in U\}$

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

יהי $U \in Sol_i$. נסתכל על טיסה t_i. לגביה יש רק שתי אפשרויות:
1) $t_i \in U \Rightarrow U \in S_{in}$
2) $t_i \notin U \Rightarrow U \in S_{out}$
מכאן נובע ש-$Sol_i = S_{in} \cup S_{out}$

הסקת הצורה הסכמטית של נוסחת הרקורסיה

נגדיר: לכל קבוצת פתרונות S נגדיר אופרטור $O*$ כך ש-$O*(S) = \max_{U \in S}\{C(U)\}$
למשל $O*(S_{in}) = \max_{U \in S_{in}}\{C(U)\}$
ראינו לכל $U \in Sol_i \Rightarrow U \in S_{in} or\ U \in S_{out}$
לכן **מהגדרת** OPT נקבל $OPT(i) = \max\{O*(S_{in}), O*(S_{out})\}$

ניתוח כל תת קבוצת פתרונות בנפרד והוכחת המרכיבים המתאימים בנוסחת המבנה:

טענה: $O*(S_{in}) = OPT(a(i)) + p_i$
כיוון $\ge$ יהי $U* \in Sol_{a(i)}: C(U*) = OPT(a(i))$. נגדיר $U** = U* \circ t_i$ מהגדרת a(i)
נקבל $t_{a(i)} + m \le t_i$ לכן $U** \in Sol_i$ וגם $t_i \in U$ לכן $U** \in S_{in}$
לכן $O*(S_{in}) \ge C(U**)$ וגם $C(U**) = C(U*) + p_i$
לכן $C(U**) = OPT(a(i)) + p_i$. נאחד ונקבל $O*(S_{in}) \ge OPT(a(i)) + p_i$
כיוון $\le$ יהי $U* \in S_{in}: C(U*) = O*(S_{in})$. יהי $U** \in Sol_{a(i)}$. נגדיר $U' = U* \setminus \{t_i\}$
לכן $U' \in Sol_{a(i)}$ לכן $C(U') \le OPT(a(i))$

לכן $O*(S_{in}) \le OPT(a(i)) + p_i \Leftarrow C(U') = C(U*) - p_i = O*(S_{in}) - p_i$ <u>טענה:</u> $O*(S_{out}) = OPT(i-1)$ **<u>כיוון ≥</u>** יהי $C(U*) = OPT(i-1)$: $U* \in Sol_{i-1}$. $t_i \notin U$ **ולכל** $t_j \in U*$ **מתקיים** $t_j + m \le t_j + 1$ **וגם** $t_i \notin U$ **לכן** $U* \subseteq \{t_1 \ldots . t_{i-1}\}$ **לכן** $U* \in S_{out}$ **לכן** $O*(S_{out}) \ge C(U*) = OPT(i-1)$ **<u>כיוון ≤</u>** יהי $C(U*) = O*(S_{out})$:$U* \in S_{out}$. **לכן** $t_i \notin U$ **לכן** $U \subseteq \{t_1 \ldots . t_{i-1}\}$ **ו-U** **פתרון חוקי לכן** $U* \in Sol_{i-1}$ **לכן** $O*(S_{out}) = C(U*) \le OPT(i-1)$

<u>סעיף ד':</u>

<u>נסחו אלגוריתם **איטראטיבי** למציאת **ערך** פתרון אופטימאלי</u>

נגדיר M מערך חד מימדי [M[0..n. איתחול $M[0] = 0, M[1] = p_1$ $for\ i\ in\ 2\ to\ n$: $M[i] = \max\{p_i + M[a(i)], M[i-1]\}$

<u>הוכחת אלגוריתם איטרטיבי (חישוב נכון של המערך M):</u>

המערך נסרק מ-2 עד n, כאשר הערכים [M[0 ו-[M[1 כבר חושבו. כל פעם אנחנו מסתמכים על תא [M[j אבל רק כך ש-$j < i$ ($0 \le a(i) < i$ וגם $0 \le i-1 < i$) לכן [M[j כבר חושב באיטרציה קודמת של הלולאה לכן הערך החדש הוא חוקי. באינדוקציה: נניח שלכל $j < i$ מתקיים $M[j] = OPT(j)$. לכן עבור i לפי הנחת האינדוקציה $M[i] = \max\{p_i + M[a(i)], M[i-1]\} = \max\{p_i + OPT(a(i), OPT(i-1)\} = OPT(i)$

<u>זמן ריצה:</u>

האלגוריתם סורק את כל המערך, ובכל איטרציה מבצע O(1) פעולות. לכן סה"כ O(n).

<u>סעיף ה':</u>

יהיה t_i זמן טיסה מקסימאלי.

באיזה תנאי t_i שייך לפתרון אופטימאלי (כלשהו)? באיזה תנאי t_i שייך **לכל** פתרון אופטימאלי:

נגדיר {Sol* = {U: C(U) = OPT(n. יהי $U \in Sol*$ כלשהו. נסתכל על המערך M. $t_i \in U$ אם במערך M **מתקיים** $M[a(i)] + p_i \ge M[i-1]$. **לכל** $U \in Sol*$ מתקיים $t_i \in U$ אם לכל $a(i) < j < i$ מתקיים $M[a(i)] + p_i > M[j]$