אלגוריתמים 1

תוכן העניינים

פרק 1

אלגוריתמים חמדניים

גישה לפתרון בעיות אלגוריתמיות, שבה האלגוריתם בוחר בכל צעד את האפשרות הטובה ביותר כרגע.

1. שיבוץ משימות על מכונה אחת

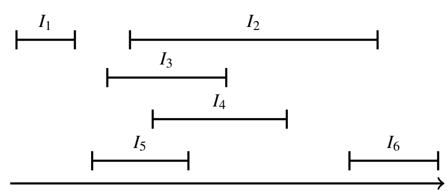
נתונות n משימות, כך שכל משימה i מיוצגת ע"י זמן התחלה s_i וזמן סיום f_i.

יש מכונה בודדת, שיכולה בכל רגע נתון לבצע לכל היותר משימה אחת, ונניח שרשימת הבקשות למשימות, כולל זמני ההתחלה והסיום, ידועה מראש.

מטרה: מה המספר הכי גדול של משימות שניתן לבצע?

1.1. תיאור הבעיה ע"י אינטרוולים.

דוגמה 1.1 נתבונן בדוגמה הבאה, כאשר נניח כי הקצה השמאלי מסמן s_i והימני f_i:



איור 1: בעיית שיבוץ משימות בייצוג של אינטרוולים

$\{I_2, I_5, I_6\}$

פתרון אופטימלי גודלו 3: $\{I_2, I_3, I_6\}$

$\{I_2, I_4, I_6\}$

תיאור אלטרנטיבי של הבעיה: רוצים לבחור תת-קבוצה גדולה ביותר של אינטרוולים כך שכל שניים לא נחתכים.

שאלה 1.1 מה המשמעות של הגישה החמדנית עבור בעיית השיבוץ שלנו?
נחליט על איזשהו סדר על האינטרוולים, ובאופן "חמדני" נעבור על האינטרוולים לפי סדר זה, ונוסיף לפתרון אינטרוול אם הוא לא נחתך עם האינטרוולים שנבחרו עד עכשיו.

1.2. הסדר החמדן.

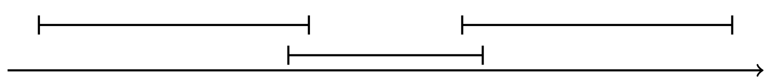
שאלה 1.2 מהו הסדר החמדן בו נעבור על האינטרוולים? מספר אפשרויות:

(1) ? לפי זמן סיום מהמוקדם למאוחר (או בסדר הפוך).
(2) ✗ לפי אורך האינטרוול, מהקצר לארוך.
(3) ✗ לכל אינטרוול נספור עם כמה אינטרוולים אחרים הוא נחתך, ונעבור מהמספר הקטן לגדול.
(4) ✗ לפי זמן התחלה מהמוקדם למאוחר (או בסדר הפוך).

נשים לב שחלק מהסדרים הללו לא יחזירו את התשובה הנכונה.
הנה מספר דוגמאות נגדיות לחלק מההצעות לעיל לסדרים חמדניים:

1.3. סדר חמדן לא נכון עלול להוביל לתוצאה שגויה.

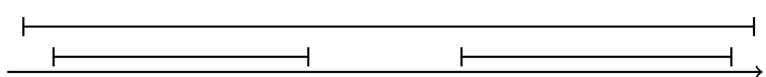
עבור הצעה מספר (2):



איור 2: דוגמה נגדית להצעה מספר (2).

הפתרון האופטימלי הוא 2, אך לפי סדר (2) יוחזר 1.

עבור הצעה מספר (4):



איור 3: דוגמה נגדית להצעה מספר (4).

הפתרון האופטימלי הוא 2, אך לפי סדר (4) יוחזר 1.

תרגיל: הוכיחו שגם הצעה (3) איננה נכונה.

ואמנם, הסדר החמדן (1) תמיד נותן פתרון אופטימלי (נראה זאת בהמשך).

1.4. האלגוריתם החמדן, הוכחת נכונות.

האלגוריתם:

(1) מיין את האינטרוולים לפי זמן סיום לא יורד ($f_1 \leq f_2 \leq f_3 \leq \ldots \leq f_n$).
(2) הגדר $X \leftarrow \emptyset$.
(3) עבור $1 \leq j \leq n$:
אם I_j לא נחתך עם אף אינטרוול ב-X, בצע $X \leftarrow X \cup \left\{I_j\right\}$.
(4) הפלט זה X.

סיבוכיות זמן ריצה: $O\left(n\log n\right)$.

- מיון האינטרוולים לפי זמני הסיום: מתבצע ב-$O\left(n\log n\right)$.
- אתחול X: מתבצע ב-$O\left(1\right)$.
- מעבר על האינטרוולים ובניית X: מתבצע ב-$O\left(n\right)$.

(הסבר: כדי לבדוק האם אינטרוול נחתך עם X, מספיק לבדוק האם נחתך עם האינטרוול המסתיים מאוחר ביותר ב-X. באמצעות מידע נוסף זה, ניתן לממש את המעבר על האינטרוולים בזמן לינארי).

משפט 1.1 (הזדהות של האלגוריתם החמדן עם אופטימום כלשהו בכל איטרציה)
בסיום איטרציה k, קיים פתרון אופטימלי X^* כך שמתקיים:

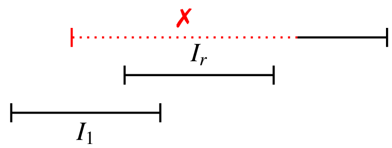
$$I_j \in X^* \iff I_J \in X$$

מסקנה 1.1 בסיום האיטרציה ה-n, קיבלנו ש-X שווה לאיזשהו פתרון אופטימלי X^*.

הוכחת המשפט. באינדוקציה על k.

- בסיס: $k = 1$.
בסיום האיטרציה הראשונה, $x = \{I_1\}$.
יהי X^* איזשהו פתרון אופטימלי לבעיה.
  - אם $I_1 \in X^*$, סיימנו.
  - אחרת $I_1 \notin X^*$, ויהי $I_r \in X^*$ האינטרוול בעל זמן הסיום המוקדם ביותר.
נסתכל על $X^* \setminus \{I_r\} \cup \{I_1\}$, ונטען שאוסף אינטרוולים זה הוא פיזיבילי (חוקי),
ומכיוון שגודלו זהה לגודל X^* הוא גם אופטימלי.
מספיק שנראה ש-I_1 לא נחתך עם אף אינטרוול ב-$X^* \setminus \{I_r\}$.

כל אינטרוול ב-$X^* \setminus \{I_r\}$ מתחיל אחרי סיום I_r, ו-I_r מסתיים לא אחרי זמן הסיום של I_r,
ולכן I_1 לא נחתך עם אף אינטרוול ב-$X^* \setminus \{I_r\}$.



איור 4: כל אינטרוול שאינו I_r ב-X^* בהכרח לא נחתך עם I_r, לכן גם לא יחתך עם I_1.

- <u>צעד האינדוקציה:</u> נניח נכונות עד סיום האיטרציה ה-k,
ויהי X^* פתרון אופטימלי שקיומו מובטח מהנחת האינדוקציה, כלומר: $I_j \in X, \ \forall j = 1, \ldots, k \iff I_j \in X^*$.
ישנם שני מקרים:
  (1) האלגוריתם בחר את I_{k+1}, כלומר: $I_{k+1} \in X$.
  - אם $I_{k+1} \in X^*$, סיימנו את הצעד.
  - אחרת, $I_{k+1} \notin X^*$, ואז נבחר $I_r \in X^*$ עם $r > k+1$ בעל זמן סיום קטן ביותר
  (קיום I_r שכזה מובטח כי אחרת $|X^*| < |X|$ בסיום האיטרציה ה-$k+1$, וזוהי סתירה לאופטימליות X^*).

  נסתכל על: $X^* \setminus \{I_r\} \cup \{I_{k+1}\}$.
  כל מה שנשאר להראות זה ש-$X^* \setminus \{I_r\} \cup I_{k+1}$ פתרון פיזיבילי:
  נראה ש-I_{k+1} לא נחתך עם אף אינטרוול ב-$X^* \setminus \{I_r\}$.

  <u>I_{k+1} לא יכול להיחתך עם אינטרוולים עם אינדקס $\{1, \ldots, k\}$ ב-$X^* \setminus \{I_r\}$.</u>
  זאת משום ש-I_{k+1} נבחר ע"י האלגוריתם, לכן לכל $1 \le j \le k$, כך ש-$I_j \in X$,
  I_{k+1} לא נחתך אתו. מהנחת האינדוקציה $I_j \in X^*$, לכן הטענה נובעת.

  <u>כמו כן, I_{k+1} לא יכול להיחתך עם אינטרוולים עם אינדקס $\{k+1, \ldots, n\}$ ב-$X^* \setminus \{I_r\}$:</u>
  באופן דומה לבסיס האינדוקציה (איור 4), לכל $I_j \in X^* \setminus \{I_r\}$:

$$s_j \le f_j \underbrace{<}_{\substack{\text{אינטרוולים ב-}X^* \setminus \{I_r\} \\ \text{עם אינדקס } r+1 \text{ ומעלה} \\ \text{לא נחתכים עם } I_r}} s_{k+1} \le f_{k+1} \le f_{k+2} \le \ldots \le f_r$$

  שהרי מהמינימליות של I_r במובן של זמן סיום קטן ביותר כך שאינו I_{k+1},
  ב- $X^* \setminus \{I_r\}$ אין אינטרוולים עם אינדקס $k+1, k+2, k+3, \ldots, r$.

  ולכן, $X^* \setminus \{I_r\} \cup \{I_{k+1}\}$ הוא פתרון פיזיבילי ואופטימלי שמקיים:

$$I_j \in X^* \setminus \{I_r\} \cup \{I_{k+1}\}, \ \forall j = 1, \ldots, k+1 \iff I_j \in X$$

  (2) האלגוריתם לא בחר את I_{k+1}, כלומר $I_{k+1} \notin X$.
  נטען גם ש-X^* שקיומו מובטח מהנחת האינדוקציה מקיים את מה שצריך: $I_{k+1} \notin X^*$.
  האלגוריתם לא בחר את I_{k+1} כי הוא נחתך עם אינטרוולים ב-X,
  ולכן I_{k+1}, לפי הנחת האינדוקציה, נחתך גם עם אינטרוולים ב-X^*, ולכן $I_{k+1} \notin X^*$.

■

שאלה 1.3 (פתרון חמדני בתוספת משקלים אי שליליים)
נניח שלאינטרוול I_j יש רווח $P_j \geq 0$.
כיצד נמצא אוסף אינטרוולים כך שכל שניים באוסף לא נחתכים, שממקסם את סך הרווחים?

לפי רועי, קשה (עד בלתי אפשרי) למצוא פתרון חמדני שיגיע לתשובה הנכונה במקרה זה (משקלים אי-שליליים).
נתייחס לבעיה זו בהמשך הקורס.

בכל אופן, נשים לב ששינוי קטן מאוד בניסוח הבעיה עלול להערים קשיים רבים על הגישה החמדנית.

2. קידודים

המטרה היא לקודד קובץ המורכב מתווים (למשל, בשפה האנגלית) בעזרת $\{0,1\}$, כך שאורך הקובץ המקודד יהיה כמה שיותר קטן.

שאלה 1.4 כיצד מקודדים? - כל תו נתון יועתק למילה מעל הא"ב $\{0,1\}$.

הגדרה 1.1 (מילת קוד) מעל א"ב $\{0,1\}$, מילת קוד היא רצף של אפסים ואחדים:

$$w = a_1 a_2 \dots a_\ell \qquad a_i \in \{0,1\} \ \ \forall i = 1, \dots, \ell$$

הגדרה 1.2 (אורך של מילת קוד) נסמן ב-$\ell(w)$ את האורך של מילת הקוד w.

הגדרה 1.3 (קוד) אוסף של מילות קוד שונות יקרא קוד.

הערה 1.1 (סימון) עבור קוד $C = \{c_1, c_2, c_3\}$ ותווים $\{x_1, x_2, x_3\}$,
נסמן את כלל ההתאמה בין כל תו למילת קוד באופן הבא: $C = \left\{ \overset{x_1}{c_1}, \overset{x_2}{c_2}, \overset{x_3}{c_3} \right\}$.

דוגמה 1.2 נתון הקוד: $C = \left\{ \overset{x_1}{c_1}, \overset{x_2}{c_2}, \overset{x_3}{c_3} \right\}$, ע"י: $c_1 = 00, \ c_2 = 01, \ c_3 = 011$
הקידוד של שרשור שלושת התווים $x_1 x_2 x_3$ יתקבל בתור:

$$\underbrace{00}_{c_1}\underbrace{01}_{c_2}\underbrace{001}_{c_3}$$

הגדרה 1.4 (קוד חד-פענח) קוד יקרא חד-פענח, אם הקידוד של כל רצף תווים ניתן לפענוח באופן יחיד.

דוגמה 1.3 (דוגמה לקוד לא חד-פענח) נתבונן בקוד $C = \left\{ \overset{x_1}{c_1}, \overset{x_2}{c_2}, \overset{x_3}{c_3} \right\}$, עם: $c_1 = 01, \ c_2 = 0, \ c_3 = 10$.
ניתן לפענח את הקידוד 010 בשני אופנים:

(1) $c_2 c_3$, מתאים ל-$x_2 x_3$.
(2) $c_1 c_2$, מתאים ל-$x_1 x_2$.

נשים לב שבקוד זה, מילת הקוד $c_2 = 0$ הינה רישא של מילת הקוד $c_1 = 01$.

הגדרה 1.5 (קוד חסר רישאות) קוד יקרא חסר רישאות אם אין בו מילת קוד שהיא רישא של מילת קוד אחרת.

דוגמה 1.4 הקוד מדוגמה 4.2 הוא לא חסר רישאות.

אנחנו נתמקד בקודים מסוג חד-פענח, שהם חסרי רישאות.

2.1. פיענוח של קידודים בעזרת קודים חסרי רישאות.

שאלה 1.5 כיצד מפענים קידוד בעזרת קוד חסר רישאות?

נשים לב שהיות שהקוד הוא חסר רישאות, ישנה דרך יחידה לפענח כל מילת קוד שמופיעה בקידוד.

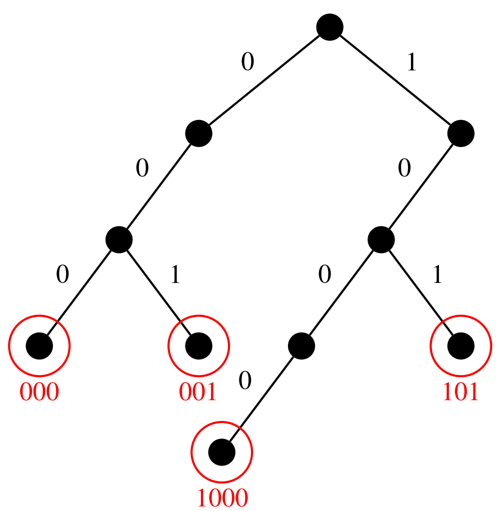
סורקים את הקידוד, וברגע שמזהים מילת קוד, מפענחים אותה וממשיכים.

הערה 1.2 (קוד חסר רישאות כעץ בינארי) קוד חסר רישאות ניתן לייצוג בעזרת עץ בינארי.

כל אחד מהילדים הישירים של צומת פנימי יותאם ל-0 או 1, ומילות הקוד יהיו העלים בעץ.

דוגמה 1.5 (עץ בינארי שמתאר קוד חסר רישאות)

$$C = \{\mathbf{000}, \mathbf{001}, \mathbf{1000}, \mathbf{101}\}$$



איור 5: עץ בינארי של קוד לדוגמה.

2.2. תיאור הבעיה.

- נתון: נתונים n תווים $x_1,\dots,x_n$, ולכל תו x_i מספר מופעים f_i.
- מטרה: למצוא קוד חד-פענח עם n מילות קוד $C=\{c_1,c_2,\dots,c_n\}$ שממזער את הביטוי $\sum_{i=1}^{n} f_i \cdot \ell(c_i)$ (כלומר, ממזער את אורך הקידוד).

טענה 1.1 (ללא הוכחה) מבין כל הפתרונות האופטימליים, קיים לפחות אחד שהוא קוד חסר רישאות.

מסקנה 1.2 (בקוד חסר רישאות שהוא פתרון אופטימלי, אורך מילת קוד הוא עומק העלה בעץ)
רוצים לחפש עץ בינארי T עם n עלים שממזער את $\sum_{i=1}^{n} f_i \cdot \underbrace{d_T(c_i)}_{\text{עומק העלה ה-}i\text{ בעץ }T}$.

טענה 1.2 (אבחנה) עץ בינארי המייצג קוד אופטימלי הוא שלם (לכל צומת פנימי שאינו עלה יש שני ילדים ישירים).

הוכחה. נניח בשלילה שיש בעץ צומת פנימי לו ילד ישיר בודד, אז נמחק צומת זה ונבחר את הילד הישיר שלו להורה של הצומת. קיבלנו עץ בינארי חדש שאורך הקידוד שהוא מגדיר לא יותר ארוך מקידוד העץ המקורי. ■

באופן "איכותי", נרצה שלתווים נפוצים יתאים קידוד קצר ולתווים נדירים קידוד ארוך.

2.3. האלגוריתם של Huffman.

האלגוריתם (של Huffman):

- נתון: אוסף משקלים ממוינים $f_1 \ge f_2 \ge \dots \ge f_k$.
- תנאי עצירה: $k=1$, ואז נחזיר עץ שהוא צומת בודד ללא קשתות.
- רקורסיה:
  - נאחד את התווים ה-k וה-$k-1$ לתו חדש שמשקלו $f' = f_{k-1} + f_k$.
  - נפעיל את האלגוריתם רקורסיבית על $k-1$ המשקלים החדשים (המשקלים $f_1,\dots,f_{k-2}$ ו-f'), ומקבלים עץ T.
  - ניקח את T, ולעלה שמייצג את איחוד התווים ה-k וה-$k-1$ נוסיף שני ילדים ישירים:
    אחד מהם מייצג את התו ה-k והשני את התו ה-$k-1$.
  - מחזירים את העץ משלב (3).

דוגמה 1.6 (דוגמת הרצה) עבור המשקלים והתווים הבאים, נציג ריצה של האלגוריתם:

ערך	תו	משקל
5	x_1	f_1
4	x_2	f_2
3	x_3	f_3
2	x_4	f_4
1	x_5	f_5

ראשית, נציג את הצעדים הרקורסיביים:

(1)

משקל	תו	ערך
f_1	x_1	5
f_2	x_2	4
f_3	x_3	3
f_4	x_4	2
f_5	x_5	1

$\Longleftarrow$ (2)

משקל	תו	ערך
f_1	x_1	5
f_2	x_2	4
f_3	x_3	3
f'	$\{x_4, x_5\}$	3

$\Longleftarrow$ (3)

משקל	תו	ערך
f_1	x_1	5
f_2	x_2	4
f''	$\{x_3, x_4, x_5\}$	6

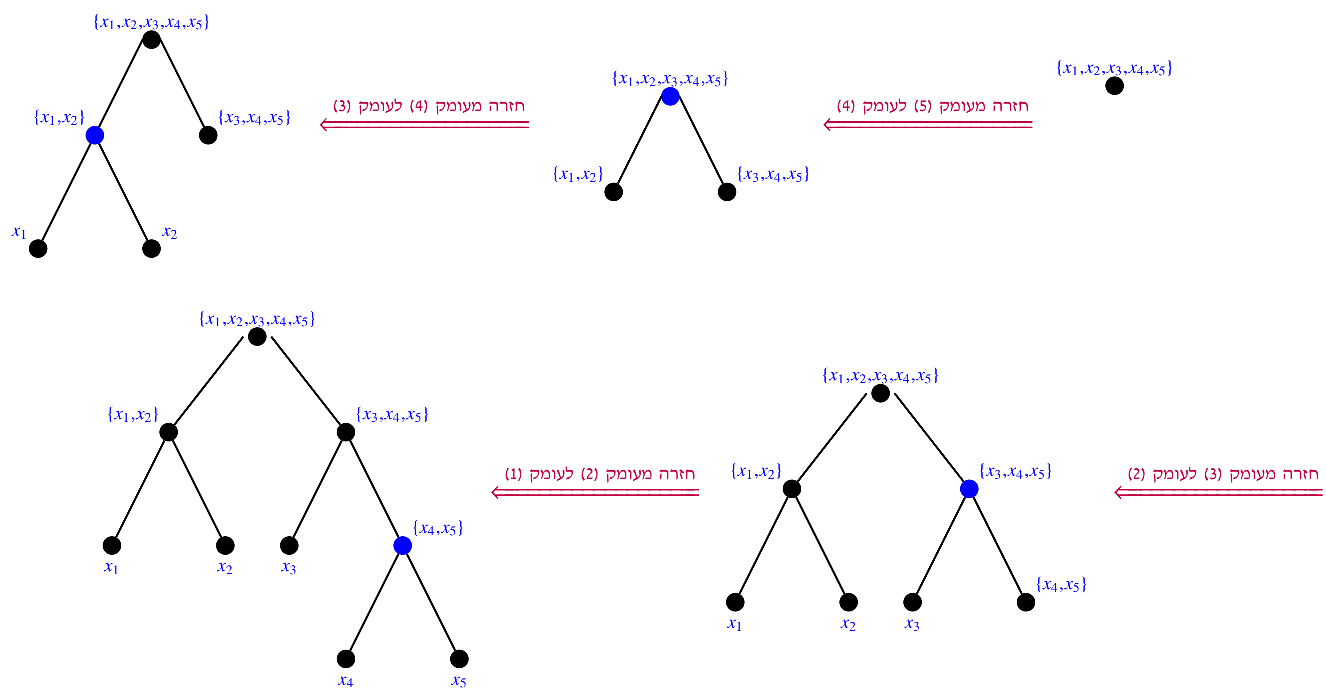
$\Longleftarrow$ (4)

משקל	תו	ערך
f'''	$\{x_1, x_2\}$	9
f''	$\{x_3, x_4, x_5\}$	6

$\Longleftarrow$ (5)

משקל	תו	ערך
f''''	$\{x_1, x_2, x_3, x_4, x_5\}$	15

כעת, נציג את החזרות מהרקורסיה (פתיחות של תווים):



2.3.1. זמן הריצה של האלגוריתם של Hoffman.

- מיון ראשוני: $O(n\log n)$.
- כל צעד רקורסיבי: $O(\log n)$ (למשל ע״י חיפוש בינארי).
- כל חזרה מרקורסיה: $O(1)$ (ע״י כך שנזכור מי שני התווים האחרונים שאיחדנו).

$\Longleftarrow$ סה״כ: $O(n\log n)$.

טענה 1.3 בהינתן משקלים ל-n תווים: $f_1 \geq f_2 \geq \ldots \geq f_n$, קיים עץ אופטימלי T בו התו ה-n והתו ה-n-1 הם עלים אחים עמוקים ביותר ב-T.

הוכחה. נניח בשלילה שאין עץ T שכזה, ויהי T^* עץ אופטימלי.
האבחנה גוררת שב-T^* יש שני עלים אחים עמוקים ביותר.

לפחות אחד משני התווים ה-n וה-n-1 אינו מיוצג ע"י שני עלים אלו.
ניקח את העלה שמייצג את התו שחסר מבין ה-n וה-n-1, ונחליף אותו עם העלה מבין השניים האחים העמוקים ביותר שאינו ה-n וה-n-1.

נבחין כי בהכרח אורך הקידוד של העץ החדש יכול רק לקטון, כי משקל כל עלה שאינו ה-n וה-n-1 הוא לפחות $f_{n-1} \geq f_n$.

- אם הערך (אורך הקידוד) קטן, זו סתירה לאופטימליות של T^*.
- אחרת, אם התו ה-n וה-n-1 בעץ החדש שקיבלנו אחים (עמוקים ביותר), קיבלנו סתירה וסיימנו.
- אם לא, נבצע שוב החלפה שכזו עבור התו השני שחסר מבין ה-n וה-n-1.

■

משפט 1.2 יהיו n תווים עם משקלים $f_1 \geq f_2 \geq \ldots \geq f_n$,
ויהי T' עץ אופטימלי עבור הבעיה המצומצמת עם n-1 תווים שמשקליהם: $f_1, f_2, \ldots, f_{n-2}$ ו-$f' = f_{n-1} + f_n$.

יהי T העץ עבור n התווים המתקבל מ-T' באופן הבא:

(1) לוקחים את T'.
(2) לעלה שמשקלו f', מוסיפים שני ילדים ישירים שמייצגים את התווים ה-n וה-n-1.

אזי T עץ אופטימלי עבור המשקלים: $f_1, \ldots, f_n$.

הוכחה. נסמן עבור T את אורך הקידוד שהוא משרה ע"י $\text{cost}(T)$.
נניח בשלילה שקיים עץ T'' עבור n התווים כך שמתקיים $\text{cost}(T'') \leq \text{cost}(T)$.

בה"כ, לפי הטענה, התווים ה-n וה-n-1 הם אחים עמוקים ביותר ב-T''.
מתקיים:

$$\text{cost}(T) = \text{cost}(T') + f_{n-1} + f_n$$

ניצור בעזרת T'' עץ חדש T''' עבור n-1 התווים (התווים 1 עד n-2 ותו חדש שמייצג את האיחוד של התווים ה-n וה-n-1),
ע"י זה שנמחק את שני העלים האחים (עמוקים ביותר) ב-T'' שמייצגים את התווים ה-n וה-n-1,
ונכריז על ההורה הישיר שלהם כעלה שמייצג את התו שהוא האיחוד שלהם:

$$\underbrace{\text{cost}(T''')}_{\text{עבור } n\text{-1 התווים}} = \text{cost}(T'') - f_{n-1} - f_n \underbrace{<}_{\text{הנחת השלילה}} \text{cost}(T) - f_{n-1} - f_n = \text{cost}(T')$$

וזאת סתירה לאופטימליות של T'.

■

פרק 2

תכנון דינאמי

טכניקה שמאפשרת פתרון בעיות כאשר לבעיה הנתונה יש מבנה רקורסיבי.

1. דוגמה של כפל מטריצות

- נתון: $A_1 \cdot A_2 \cdot \ldots \cdot A_n$, כאשר A_i זו מטריצה במימדים $n_i \times n_{i+1}$.
- מטרה: למצוא דרך לחישוב המכפלה, שממזערת את מספר הכפלים הסקלאריים שמתבצעים.

אם כופלים $A \cdot B$ ו-A במימדים $p \times q$ ו-B במימדים $q \times r$, מספרים הכפלים הסקלאריים שמתבצעים הינו $p \cdot q \cdot r$.

דוגמה 2.1 $A_1 \cdot A_2 \cdot A_3$, והמימדים $A_1 = 10 \times 100,\ A_2 = 100 \times 5,\ A_3 1 = 5 \times 50$

מספר המכפלות יהיה אחד משניים:

- $(A_1 \cdot A_2) \cdot A_3$: $10 \cdot 100 \cdot 5 + 10 \cdot 5 \cdot 50 = 7500$
- $A_1 \cdot (A_2 \cdot A_3)$: $100 \cdot 5 \cdot 50 + 10 \cdot 100 \cdot 50 = 75000$

שאלה 2.1 מדוע לא ניתן לעבור על כל האפשרויות?

נקבל שמספר האפשרויות לביצוע המכפלה עם n מטריצות הינו:

$$p(n) = \begin{cases} 1 & n = 1 \\ \sum_{k=1}^{n-1} p(k)\, p(n-k) & n > 1 \end{cases}$$

הפתרון הוא מספרי קטלן ($p(n) = C(n-1)$), וניזכר שמתקיים:

$$C(n) = \frac{1}{n+1}\binom{2n}{n} = \Omega\left(\frac{4^n}{n^{\frac{3}{2}}}\right)$$

$\Longleftarrow$ לא נרצה לבצע חיפוש ממצה על פני כל האפשרויות לחישוב המכפלה באורך n
(מספר האפשרויות הוא אקספוננציאלי באורך n)!