

אלגוריתמים הרצאה 9

תכנון דינאמי

טכניקה המאפשרת לפרק בעיה באופן רקורסיבי לתתי-בעיות קטנות מאותו סוג.

דוגמה 1:

שיבוץ משימות ממושקלות.

נתון: n אינטרוולים, כאשר כל אינטרוול נתון ע"י התחלה s_i וזמן סיום f_i . לאינטרוול ה- i נתון רווח w_i .

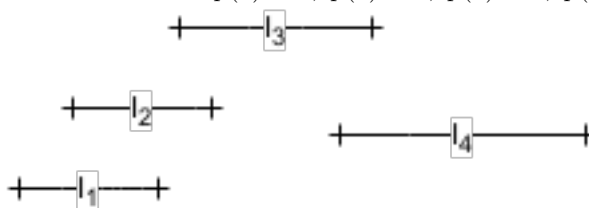
מטרה: למצוא אוסף אינטרוולים S כך שכל שני אינטרוולים ב- S לא נחתכים שממקסם $\sum_{j \in S} w_j$.

נזכר שמיינו את האינטרוולים: $f_1 \leq f_2 \leq \dots \leq f_n$.

עבור אינטרוול I_j , נסמן ב- $p(j)$ את האינדקס הכי גדול כך ש- $p(j) \leq s_j$ (כלומר, אינטרוול $I_{p(j)}$ לא נחתך עם אינטרוול I_j). אם לא קיים אינטרוול כזה, p_j יוגדר להיות 0.

דוגמה:

$$p(1) = 0, p(2) = 0, p(3) = 1, p(4) = 2$$



שאלה:

מה הן תתי הבעיות שנוצרו ?

תתי-הביות שיעניינו אותנו יהיו רישאות, כלומר לכל $0 \leq j \leq n$ הראישר ה- j תכיל $\{I_1, I_2, \dots, I_j\}$

הגדרה:

נסמן ב- $A(j)$ את ערך הפתרון האופטימלי עבור תת-הבעיה ה- j .

שאלה:

האם יש קשר רקורסיבי בין $A(0), A(1), \dots, A(n)$?

$$A(j) = \begin{cases} 0 & j = 0 \\ \max\{A(j-1), w_j + A(p_j)\} & 1 \leq j \leq n \end{cases}$$

טענה:

$A(0), \dots, A(n)$ מקיימות את הנוסחה הרקורסיבית לעיל.

הוכחה:

אם $j = 0$ אזי $A(0)$ הוא ערך הפתרון האופטימלי עבור קלט ריק, כלומר $A(0) = 0$, כפי שאכן מגדירה הנוסחה הרקורסיבית.

אחרת $1 \leq j \leq n$. נסמן ב- S_j^* פתרון אופטימלי כלשהו לתת-הבעיה $\{I_1, I_2, \dots, I_j\}$.

* אם $I_j \notin S_j^*$ אזי S_j^* פתרון אופטימלי לבעיה ה- $j-1$ (שכן אחרת S_j^* לא היה פתרון אופטימלי לתת-הבעיה ה- j).

$$\sum_{i \in S_j^*} w_i = A(j-1) \Leftarrow$$

$$A(j) = A(j-1) \Leftarrow$$

$$\text{האם יתכן כי } \sum_{i \in S_j^*} w_i < w_j + A(p(j))?$$

לא, מפני שבמקרה זה היה פתרון עדיף מ- s_j ל- $\{I_1, I_2, \dots, I_j\}$.

* אם $I_j \in S_j^*$ אזי $S_j^* \setminus \{I_j\}$ הוא פתרון אופטימלי לתת-הבעיה $p(j)$ (שכן אחרת S_j^* לא היה פתרון אפשרי עבור תת-הבעיה ה- j).

$$\sum_{i \in S_j^*} w_i = w_j + A(p(j)) \Leftarrow$$

$$A_j = w_j + A(p(j)) \Leftarrow$$

$$\text{האם יתכן כי } A(j-1) > \underbrace{w_j + A(p(j))}_{A_j}?$$

לא! $A(0), \dots, A(n)$ מקיימת את הנוסחה הרקורסיבית.

שאלה:

האם ניתן לחשב את $A(n)$ במהירות בעזרת הנוסחה הרקורסיבית? הרעיון: בדומה לסדרת פיבונאצ'י, ניתן לשלם בזכרון ולחשב את $A(n)$ בזמן לינארי.

$$A \text{ (סדר חישוב)} \rightarrow \begin{array}{|c|c|c|c|c|c|c|} \hline 0 & & & & \dots & & \\ \hline 0 & 1 & 2 & \dots & n-1 & n & \\ \hline \end{array}$$

האלגוריתם:

1. חשב את $p(1), \dots, p(n)$

2. עבור $j = 0$ עד n בצע:

(א) חשב את $A(j)$ לפי הנוסחה הרקורסיבית.

3. החזר את $A(n)$

(ללא המיון $f_1 \leq \dots \leq f_n$ זמן הריצה הוא $O(n)$)

דוגמה 2:

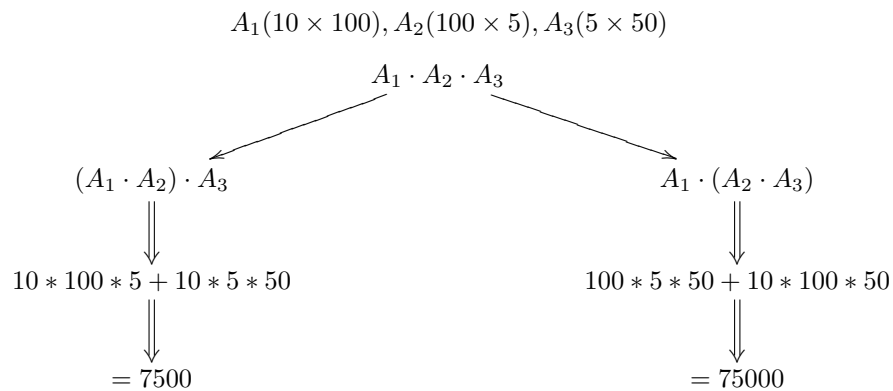
סדר ביצוע כפל מטריצות

נתון: תרגיל כפל מטריצות A_1, \dots, A_n כאשר A_i במימדים $p_{i-1} \times p_i$.

מטרה: למצוא סדר לביצוע התרגיל הממזער את מספר כפלים האריתמטיים.

(הערה: כפל של $A \cdot B$ כשא A במימדים $p \times q$ ו- B במימדים $q \times r$ דורש pqr מכפלות).

דוגמה:



שאלה:

כמה אפשרויות יש לביצוע תרגיל באורך n ?

נסמן ב- $p(n)$ את מס' האפשרויות למקם סוגריים בתרגיל באורך n .

$$p(n) \triangleq \begin{cases} 1 & n = 1 \\ \sum_{k=1}^{n-1} p(k) \cdot p(n-k) & n \neq 1 \end{cases}$$

הפתרון הוא C_{n-1}
Cathalan's number

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

⇐ לא ניתן לעבור באופן יעיל על כל האפשרויות.
נגדיר תת-בעיה לכל תת-סדרה של התרגיל, כלומר לכל $1 \leq i \leq j \leq n$: A_i, A_{i+1}, \dots, A_j

הגדרה

נסמן ב- $M(i, j)$ את המספר הכפלים האריתמטיים ברטון ביותר שצריך על מנת לפתור את $A_i \cdot A_{i+1} \cdots A_j$.

$$M(i, j) \triangleq \begin{cases} 0 & i = j \\ \min_{i \leq k \leq j} \{ \underbrace{M(i, j)}_{\text{dimensions } p_{i-1} \times p_k} + \underbrace{M(k+1, j)}_{\text{dimensions } p_k \times p_j} + p_{i-1} \cdot p_k \cdot p_j \} & i < j \end{cases}$$

טענה

$M(i, j)$ מקיים את נוסחת הרקורסיה.

שאלה:

כיצד לחשב במהירות את נוסחת הרקורסיה ?

$$M = \begin{array}{c|cccccc} & 1 & 2 & 3 & \dots & n-1 & n \\ \hline 1 & 0 & & & & & ?(\text{פלט}) \\ 2 & & 0 & & & & \\ 3 & & & 0 & & & \\ \vdots & & & & \ddots & & \\ n-1 & & & & & 0 & \\ n & & & & & & 0 \end{array}$$

נשים לב שאם נייצג את M כמטריצה מתקיים:

- ★ תנאי העצירה נותן את ערכי האלכסון.
- ★ מעוניינים רק במשולש העליון של המטריצה.
- ★ לפי נוסחת הרקורסיה $M(i, j)$ תלוי בתאי המטריצה משמאלו ומתחתיו.

⇐

קיימים הרבה סדרים שבהם ניתן לחשב את איברי המטריצה M כך שברגע שמחשבים את $M(i, j)$ כל התאים במטריצה שהוא תלוי בהם כבר חושבו. בכל סדר חישוב שכזה, חישוב תא $M(i, j)$ לוקח זמן של $O(j-i)$.
 ⇐ במקרה הגרוע זה $O(n)$.
 ⇐ זמן הריצה הוא $O(n^3)$.