

חישוב חזקה

תיאור הבעיה:

נרצה לחשב חזקה בצורה היעילה ביותר. כלומר לחשב את a^n .
בראייה ראשונית אפשר לראות כי סיבוכיות זמן הריצה עבור $a^n = a \cdot a \cdot a \cdot \dots \cdot a$ הוא $O(n)$.

ישנה אפשרות לחשב זאת ע"י בשיטה אינדוקטיבית וגם ע"י שיטה רקורסיבית, במקרה זה השיטה האינדוקטיבית עדיפה מכיוון שהשיטה הרקורסיבית מבצעת פעולות נוספות של הקצאות זיכרון עבור כל קריאה.
בשתי השיטות הסיבוכיות היא $O(n)$.

האם ניתן לספק פתרון נוסף ביעילות טובה יותר מ- $O(n)$?

פתרון:

התשובה היא שניתן, על מנת לחשב חזקה בסיבוכיות $O(n \log n)$ נעביר את החזקה להצגה בינארית.
בהתאם למיקום ולספרה 1 במס' הבינארי נדע באיזו חזק להשתמש, נצטרך לשמור בכל שלב את החזקה של X , אך לבצע כפל רק כאשר בהצגה בינארית של המספר $m_i \neq 0$.
צריך לעלות מספר ממשי X בחזקת n . נעביר את מספר n להצגה בינארית:

$$m_i = \{0,1\}, i = 0, \dots, k \text{ , כאשר } n = m_k 2^k + m_{k-1} 2^{k-1} + m_{k-2} 2^{k-2} + \dots + m_1 2 + m_0$$

לדוגמה:

$$X^7 = X^1 \cdot X^2 \cdot X^4 \leftarrow 7 = 2^0 + 2^1 + 2^2 \leftarrow n_2 = 111 \leftarrow n_{10} = 7$$

ניתן לממש את פתרון זה בשתי צורות, בעזרת לולאה ובמימוש רקורסיבי.