

תרגיל 2 - מבנה נתונים

לא ניתן להגיש את התרגיל בזוגות

21 במרץ 2023

הערה

בכל מקום שרשום ניתוח מקום של אלגוריתם מסוים שאתם מציעים, הכוונה היא לניתוח מקום נוסף שהאלגוריתם מקצה על מנת לפתור את הבעיה כלומר בלי התחשבות במקום שהקלט והפלט תופסים בזיכרון.

הבהרה

כל מקום שרשום "הסבירו את נכונות האלגוריתם" ניתן להסביר במילים (בקצרה למה האלגוריתם עובד) או להוכיח באופן פורמלי.

משפטים

משפט האב

נניח ש $f: \mathbb{N} \rightarrow \mathbb{R}^+$ וגם $a \geq 1, b > 1$. נגדיר:

$$T(n) = a \cdot T\left(\frac{n}{b}\right) + f(n)$$

$$T(1) = \Theta(1)$$

1. אם $f(n) = O(n^{\log_b a - \varepsilon})$ עבור $\varepsilon > 0$ כלשהו אזי $T(n) = \Theta(n^{\log_b a})$.

2. אם $f(n) = \Theta(n^{\log_b a})$ אזי $T(n) = \Theta(n^{\log_b a} \log n)$.

3. אם $f(n) = \Omega(n^{\log_b a + \varepsilon})$ עבור $\varepsilon > 0$ כלשהו ואם עבור $n_0 \in \mathbb{N}$ וכל $c < 1$ ולכל $n > n_0$ מתקיים $a \cdot f\left(\frac{n}{b}\right) \leq c \cdot f(n)$ אזי $T(n) = \Theta(f(n))$.

חלק 1

שאלה 1 - Merge - Sort

סעיף א'

בשיעור ראיתם את אלגוריתם Merge - Sort.
להלן פסאודו קוד 1:

Algorithm 1: Merge-Sort

```

Merge(arr, left, right)

    if left > right
        | return
    end if
    mid =  $\frac{left+right}{2}$ 
    MergeSort(arr, left, mid)
    MergeSort(arr, mid + 1, right)
    Merge(arr, left, mid, right)

```

עליכם להציע אלגוריתם שבהינתן שתי רשימות ממוינות (בסדר עולה), $arr1, arr2$ (לא בהכרח בעלות אותו אורך), מכניס את המספרים למערך A בגודל $length(arr1) + length(arr2)$ כך שהמערך A ממוין (בסדר עולה).
על האלגוריתם לפעול בזמן ריצה של $O(length(arr1) + length(arr2))$ ולתפוס מקום של $O(length(arr1) + length(arr2))$. הסבירו את נכונות האלגוריתם, נתחו את זמן הריצה ואת המקום שהאלגוריתם דורש. אין צורך לכתוב פסאודו קוד, אפשר לתאר את האלגוריתם במילים.

סעיף ב'

נגדיר אלגוריתם $Uneven - Merge - Sort$ השונה מ-Merge - Sort שראינו בכך שבשלב החלוקה במקום לחלק את המערך לשני מערכים בגודל זהה, נחלק את המערך לשני מערכים בגדלים שונים, אחד מהם יהיה בגודל $\frac{p}{b}$ ואחד יהיה בגודל $\frac{q}{b}$ כאשר מתקיים $q + p = b$. נוסחאת הנסיגה על כן תהיה

$$T(n) = T\left(\frac{pn}{b}\right) + T\left(\frac{qn}{b}\right) + cn$$

הראו בעזרת שיטת עץ רקורסיה, כפי שראינו בתרגול, כי ניחוש הדוק עבור נוסחת הנסיגה הוא $T(n) = \theta(n \log n)$.

שאלה 2 - Master – Theorem

סעיף א'

נתונה פונקציה $T : \mathbb{N} \rightarrow \mathbb{R}^+$ המוגדרת להיות:

$$T(n) = 2T\left(\frac{n}{2}\right) + \frac{n}{\log n}$$

$$T(1) = 1$$

הסבירו (כלומר נמקו באופן פורמלי) מדוע פונקציה זו איננה עומדת בתנאים של משפט האב (כפי שנלמד בתרגול)

סעיף ב'

לפניכם הגרסה הפשוטה של משפט האב, שראינו בהרצאה.
תהי נוסחת נסיגה עבור הפרמטרים $a \geq 1, b > 1, c \geq 0$:

$$T(n) = a \cdot T\left(\frac{n}{b}\right) + n^c$$

$$T(1) = 1$$

(ניתן להניח ש n הוא חזקה שלמה של b). מתקיים:

1. אם $\frac{a}{b^c} < 1$ אז $T(n) = \Theta(n^c)$

2. אם $\frac{a}{b^c} = 1$ אז $T(n) = \Theta(n^c \log_b n)$

3. אם $\frac{a}{b^c} > 1$ אז $T(n) = \Theta(n^{\log_b a})$

הוכיחו שהגרסה שראיתם בתרגול גוררת גרסה זו של משפט האב.

חלק 2

שאלה 3 - Master – Theorem

לכל אחת מהפונקציות הבאות, מצאו חסם אסימפטוטי הדוק ככל האפשר :

$$1. S(n) = 8S\left(\frac{n}{2}\right) + n^3$$

$$2. R(n) = 32R\left(\frac{n}{8}\right) + n!$$

ניתן להשתמש בעובדה כי עבור $k, c \in \mathbb{N}$ מתקיים כי $c! \cdot k! < (ck)!$

$$3. Q(n) = 4Q\left(\frac{n}{16}\right) + \log n$$

$$4. T(2) = 1 \text{ כאשר } T(n) = 2T\left(\frac{n}{2}\right) + n \log^2 n$$

$$5. T(n) = 4\sqrt{n}T(\sqrt{n}) + n \cdot \log n \cdot \log(\log n)$$

רמז : ראינו כי ניתן לעשות החלפת משתנה על ידי $S(m) = T(2^m)$, איך נשנה את החלפת המשתנה כך שניתן יהיה להשתמש במשפט האב?

חלק 3

שאלה 4

נתון מערך בגודל n , A , של הספרות 0, 1, 2. עליכם למיין את המערך. כתבו אלגוריתם אשר מבצע את המשימה בזמן ריצה $O(n)$ וזמן ריצה $O(n)$ מקום. הסבירו את נכונות האלגוריתם וזמן הריצה שלו. כתבו פסאודו קוד של האלגוריתם. מערך לדוגמה באורך 7

$$A = [2, 0, 1, 0, 1, 2, 0]$$

יראה לאחר המיון כך

$$A = [0, 0, 0, 1, 1, 2, 2]$$

שאלה 5 (שאלת בונוס 5 נקודות)

תנאי הרגולריות במקרה 3 של משפט האב הוא: עבור $c < 1, n_0 \in \mathbb{N}$ ולכל $n > n_0$ מתקיים $a \cdot f\left(\frac{n}{b}\right) \leq c \cdot f(n)$. הביאו דוגמה לקבועים $a \geq 1, b > 1$ ופונקציה $f(n)$ העונה על כל התנאים שבמקרה 3 של משפט האב, פרט לתנאי הרגולריות.