מטלה 2
מבוא לתכנות

סופר
עלמן
326156483

שאלה 1

א. נסביר מדוע סיבוכיות הזמן של האלגוריתם Partition היא $O(n)$, כאשר n הוא גודל המערך:

- הסבר: האלגוריתם Partition עובר על כל איבר במערך פעם אחת בלבד, ומבצע בכל איטרציה פעולות בסיסיות בזמן קבוע (השוואה, החלפה, עדכון אינדקסים). כלומר, כמות הפעולות שמבצעת Partition עלולה להגיע ל O - מעלה שווה כמות האיברים (כדי לכסות את המקרה שבו אין איברים שווים לציר). לכן, סיבוכיות הזמן היא $O(n)$.

ב. נחשב את סיבוכיות הזמן של האלגוריתם QuickSort:

$N = 2^x$, $x = \log N \rightarrow O(\log N)$ (N הוא גודל המערך)

$O(N) \cdot O(\log N)$: נקבל שסיבוכיות הזמן של QuickSort היא

$= O(N \cdot \log(N))$

הסבר: פונקציית QuickSort(int arr[], int low, int high) היא פונקציה רקורסיבית. אם low < high, אז:

```
int p = partition(arr, low, high);
QuickSort(arr, low, p-1);   // T(i-1)
QuickSort(arr, p+1, high);  // T(n-i)
```

כאשר i/n הוא הסיכוי שהציר יהיה במקום ה i (לפי מערך)

מתקיים כי: $T(n) = \underbrace{(n-1)}_{O(n)} + T(n-i) + T(i-1)$ (partition)

$$= (n-1) + \frac{1}{n} \cdot \sum_{i=0}^{n-1} \left(\underbrace{T(n-i)}_{\text{מערך}} + \underbrace{T(i-1)}_{\text{מערך}} \right)$$

$$= (n-1) + \frac{1}{n} \cdot 2 \sum_{i=0}^{n-1} T(i) \quad / \cdot n$$

$$n \cdot T(n) = 2 \sum_{i=0}^{n-1} T(i) + n(n-1) \quad (1)$$

$$(n-1) T(n-1) = 2 \sum_{i=0}^{n-2} T(i) + (n-1)(n-2) \quad (2)$$

(1) − (2)

Scanné avec CamScanner

$$(1)-(2) =: nT(n) - (n-1)T(n-1) = 2T(n-1) + 2(n-1)$$

$$\Leftrightarrow nT(n) - (n+1)T(n-1) = 2(n-2) \quad \Big) / n(n+1)$$

$$\Leftrightarrow \underbrace{\frac{T(n)}{n+1}}_{f(n)} = \underbrace{\frac{T(n-1)}{n}}_{f(n-1)} + \frac{2(n-1)}{n(n+1)}$$

$$f(n) = f(n-1) + \frac{2(n-1)}{n(n+1)}$$

$$f(n) = f(n-1) + \frac{4}{n+1} - \frac{2}{n}$$

$$f(n-1) = f(n-2) + \frac{4}{n} - \frac{2}{n-1}$$

$$f(n-2) = f(n-3) + \frac{4}{n-1} - \frac{2}{n-2}$$

$$\vdots$$

$$f(n) = \underbrace{f(1)}_{a_1} + \sum_{i=2}^{n} \frac{4}{i+1} - \sum_{i=2}^{n} \frac{2}{i}$$

$$= \frac{4}{n+1} + 2\sum_{i=3}^{n+1} \frac{1}{i}$$

$$= \frac{4}{n+1} + 2\left(\ln n - \left(1 + \frac{1}{2}\right)\right)$$

$$= \frac{4}{n+1} + 2\ln n - 3$$

$$T(n) = 2(n+1)\cdot \ln n - 4n$$

$$\ln n \to \log(n)$$

$$= 2(n+1)\cdot \log(n) - 4n$$

$$= O(n \cdot \log(n))$$

P.S.N

:דוגמה

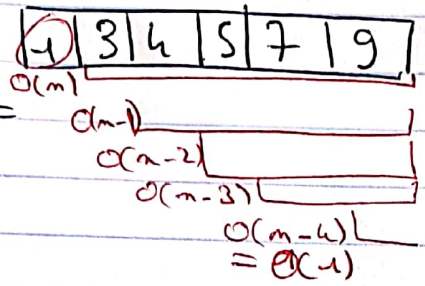
12	7	5	18	3	--	6

Pivot

מערכת של מספרים.

Scanné avec CamScanner

זוהי $O(m^2)$ ולכן הסיבוכיות של האלגוריתם היא לכל היותר $O(m^2)$.



לכן נסיק כי סיבוכיות הזמן של האלגוריתם היא $O(m^2)$.

---

2) בקוד הזה קיבלנו את ה-backtracking בזכות שהשתמשנו בפונקציה. הפונקציה הזו מוגדרת בתוך הפונקציה הראשית ולכן היא מקבלת את כל המשתנים שהוגדרו בפונקציה הראשית ולכן אין צורך להעביר אותם כפרמטרים. בנוסף, הפונקציה הזו משתמשת ב-nonlocal כדי לשנות את המשתנים שהוגדרו בפונקציה הראשית. בנוסף, הפונקציה הזו משתמשת ב-high כדי לדעת מתי לעצור את הלולאה ולא לנסות לבנות מספרים גדולים יותר מהנדרש.

הסיבוכיות של האלגוריתם היא $O(2^m)$ כאשר m הוא מספר הספרות בקלט n. הסיבוכיות נובעת מכך שבכל קריאה רקורסיבית אנחנו מנסים להוסיף ספרה חדשה למספר הנוכחי, ולכן יש לנו 2^m אפשרויות שונות לבנות מספרים באורך m או פחות.

שאלה 2

א) מאוצא מספר החילופים (קריאות לפונקציה swap) מבלבס) על מערך בגודל m+1 הוא: סכום המאוצא הזה על הקלטים האפשריים (מאורך 1+m) הוא: אחרי לוקחים את האפשרויות של המיקום i וכל פעם מחשבים כמה פעמים i מופיע בתור המיקום.

מקבלים:

$$\sum_{i=1}^{m+1} O(i) = \frac{O(m+1)\cdot(m+2)/2}{(m+1)}$$

$$= \underline{O(m)} =$$

ב) ה- מחזיר את האיבר האמצעי ולכן לאחר קריאה אחת לפונקציה הוא יחזיר את האיבר במאוצא ה- הקלט האפשרי כי יש לו מקום שם בין הקריאות swap ; אחת מהאחרות הללו בטוח שהאיבר שלנו יחזור אחרי ה- שלו ושניהם יהיו במיקומים.

ג) בהינתן קלט שונה מאחד שלפנינו, הזמן ריצה של הפונקציה יהיה $O(m)$ וכמות הקריאות ל- swap תהיה שונה מאחת ל- כי ה- swap של i יבוצע אחרי שנעבור על מערך באורך i ולכן מספר הקריאות ל- swap יהיה שונה.

אלגוריתם מבלבס על מערך בגודל m+1 יעבור על כל האיברים במערך ולכן זמן הריצה יהיה $O(m)$.