מבני נתונים - תרגיל 4

תאריך הגשה: 04.05.2023

תרגיל זה עוסק באלגוריתמי המיון: bucket sort, radix sort, counting sort ו-quicksort שנלמדו בביתה, וּוריאציות עליהם.

חלק א (37 נקודות)

Counting Sort

להלן פסאודו-קוד של Counting-Sort כפי שראינו בביתה:

אלגוריתם 1 Counting Sort

Input: arr: unsorted array of integers$\in \{0\} \cup [k]$, k: largest number in the array.
Output: sorted_arr: sorted arr.

```
1  Function Counting-Sort(arr, k):
2      C = Allocate[0...k]
3      for i = 0 to k do
4          C[i] = 0
5      for j = 0 to length(arr)−1 do
6          C[arr[j]] += 1
7      for i = 1 to k do
8          C[i] += C[i − 1]
9      sorted_arr = Allocate[0...n − 1]
10     for j = length(arr)−1 downto 0 do
11         sorted_arr[C[arr[j]] −1] = arr[j]
12         C[arr[j]] −= 1
```

א. [8 נק'] האם Counting Sort הוא מיון יציב? אם כן, נמקו בקצרה. אם לא, הראו דוגמה נגדית. (יציב - כפי שהוגדר בסיכום התרגול. כלומר הסדר היחסי בין שני איברים זהים ישמר לאחר המיון).

ב. נניח שנשנה את סדר הריצה בלולאה האחרונה של Counting-Sort מ - `for` j=`length(arr)-1 downto 0` ל-`for` j=`0 to length(arr)-1`. (שורה 10).

1. [5 נק'] הסבירו מדוע האלגוריתם עדיין יעבוד כמצופה (כלומר יחזיר מערך ממוין).
2. [10 נק'] האם אלגוריתם המיון החדש יציב? אם כן, הסבירו. אם לא, הראו דוגמה נגדית.

ג. [14 נק'] תארו אלגוריתם שמקבל כקלט מערך באורך n עם שלמים (integers) בתחום $[0,\ldots,k]$, עושה עיבוד מקדים על הקלט ולכל $1 \le a \le b \le k$ מוצא את מספר האיברים במערך שבטווח $[a,\ldots,b]$. סיבוכיות הריצה של האלגוריתם עצמו צריכה להיות $O\left(1\right)$ וזמן העיבוד המקדים שלו צריך להיות $O\left(n+k\right)$.

חלק ב (25 נקודות)

Quick Sort

א. [10 נק'] האם Quick Sort כפי שראינו הוא אלגוריתם יציב? אם כן, הסבירו. אם לא, הסבירו והראו דוגמה נגדית.

ב. [15 נק'] נניח שבכל שלב ב-Quick Sort חלוקת המערך מתבצעת ביחס של $(1-\alpha)$ ל-α (כלומר כל ציר pivot שנבחר מחלק את המערך ליחס הנ"ל), כאשר $0 < \alpha < \frac{1}{2}$ קבוע. הראו שהעומק המינימאלי של עלה בעץ הרקורסיה הוא $\Theta\left(-\frac{\log(n)}{\log(\alpha)}\right)$, והעומק המקסימאלי הינו $\Theta\left(-\frac{\log(n)}{\log(1-\alpha)}\right)$. (אין צורך לטפל בעיגול לשלמים, ניתן להניח שהחלוקה מסתדרת במובן הזה).

חלק ג (38 נקודות)

שמורת הלולאה ו-Radix Sort+Bucket Sort

א. [6 נק'] בתרגול הוכחנו את נכונות Radix Sort באמצעות שמורת הלולאה. מטרת סעיף זה היא להראות לכם ששמורת הלולאה אינה זרה לכם ודומה מאוד לאינדוקציה.
הוכיחו באינדוקציה את הנכונות של Radix Sort. היכן בהוכחה אתם משתמשים בטענה שהמיון על כל ספרה הוא יציב? הסבירו והראו דוגמה בה שימוש במיון לא יציב היה שולל את נכונות Radix Sort.

ב. [8 נק'] בספרו של מאיר שלו "הצלחת שמתחת" מתוארת הבעיה הבאה:
"יום אחד, כשאפרת פתחה את המגירה של הצלחות, היא שמעה מתוכה לחישה. מישהו אמר: "אותי... אותי... תיקחי אותי בבקשה..." "מי זה מדבר?" שאלה אפרת. "זאת אני," אמר הקול, "הצלחת שהכי למטה, הצלחת שמתחת, המסכנה והמקופחת, תיקחי אותי בבקשה אל השולחן... תמיד לוקחים את הצלחות שלמעלה ואני נשארת כאן."
לאחר דאגה רבה אפרת חושבת על האלגוריתם הבא - בכל ארוחה נוציא את הצלחת התחתונה ביותר ובסיום נחזיר אותה לתחילת הערימה. למרות התלונות של הצלחת העליונה אפרת מממשת את האלגוריתם.
בהנחה שיש n צלחות חדשות ומוציאים בכל פעם צלחת אחת, הוכיחו באמצעות שמורת הלולאה שלאחר n ארוחות יש שוויון בין הצלחות ואף אחת לא קופחה.

ג. [8 נק'] הוכיחו באמצעות שמורת הלולאה את נכונות Binary Search.

אלגוריתם 2 Binary Search

```
Input: arr, key
Output: key index.
Function BinarySearch(arr, key):
    min ← 0, max ← length(arr) − 1, mid ← 0
    while min ≤ max do
        mid = (min+max)/2
        if arr[mid] < key then
            min = mid + 1
        else if arr[mid] > key then
            max = mid − 1
        else
            return mid
    return error
```

ד. [16 נק'] ניתנים לכם n מספרים הנדגמים מ-$\left\{2^0, 2^1, \ldots, 2^k\right\}$ באופן אחיד כאשר $k = \Theta\left(n^2\right)$. תארו אלגוריתם למיון המספרים בזמן **ממוצע** של $O\left(n\right)$.

הערה: המשמעות של מספר הנדגם באופן אחיד מ-$S = \left\{2^0, 2^1, \ldots, 2^k\right\}$ היא שההסתברות שהוא שווה לכל מספר ב-S היא $\frac{1}{k+1}$.