מבחן מגן / מבוא לאלגוריתמים תשפ"ד

עני על 4 מתוך 5 שאלות.

1. מערך "מחולק" A באורך n הינו מערך שמחולק לרצפים ממוינים באופן הבא: האיבר הראשון ממוין, 2 האיברים הבאים אחריו ממוינים, 4 האיברים הבאים אחר כך ממוינים וכו' עד ל $\left\lceil\frac{n}{2}\right\rceil$ האיברים האחרונים שממוינים. כל טווח ממוין במערך זר לכל טווח אחר ממוין במערך.
לדוג': [7,2,4,100,111,112,116,12,13,14,15,17,20,21,22]
לשם פשטות ניתן להניח כי אורך המערך הוא מהצורה 2^i-1 כאשר i מספר שלם חיובי.
נתבונן בתהליך הבא הנועד לחפש מספר K במערך "מחולק":

```
FIND_NUM(A,n,K)
start ←0
end ←0
while end < n do
        if K>= A[start] And K<=A[end]
                return BinarySearch(A,start,end,K)
        range ←end-start+1
        start ←end+1
        end ←end+range*2
return NOT_FOUND
```

א. עבור סדרה של n חיפושים במערך כאשר ידוע כי בסה"כ מכל טווח ממוין במערך כמות החיפושים היא כגודל הטווח [אבל לא ידוע סדר החיפושים, שהרי אם היה ידוע ניתן היה לגשת מיד למקום המתאים במערך ורק שם לבצע חיפוש בינארי].
מהו **זמן הריצה לשיעורין** לתהליך החיפוש במקרה הגרוע?

ב. עבור מערך "מחולק" שבו ידוע כי כל האיברים הינם שלמים בתחום [0...k]
הוחלט ליצור מערך עזר help בגודל k+1 שמאותחל כולו בערך 1-. כאשר מחפשים איבר קיים במערך [באמצעות FIND_NUM], האינדקס שבו הוא נמצא במערך ה"מחולק" נשמר במערך העזר.
לדוג': אם מחפשים את האיבר 15 במערך ומוצאים אותו באינדקס 18 אז משנים את help[15] ל18.
כמו כן, לפני ביצוע תהליך החיפוש FIND_NUM של איבר K במערך, בודקים האם help[K]!=-1 ואם כן מחזירים מיד help[K] [בלי צורך בחיפוש].
עבור מערך "מחולק" בגודל n וסדרה של n^2 חיפושים של איברים קיימים במערך, מהו **זמן הריצה לשיעורין** של חיפוש בודד במקרה הגרוע?

2. נתון מערך לא ממוין באורך n המכיל מספרים המייצגים אורכי צלעות.

א. כתבי פונקציה [בפסאודוקוד או בכל שפת תכנות הנוחה לך] המקבלת את המערך ומוצאת 3 צלעות במערך המרכיבות משולש עם היקף **מקסימלי** ככל האפשר.
תזכורת: משולש תקני אפשרי רק אם האורך הכולל של כל 2 צלעות שלו גדול ממש מהצלע השלישית.
לדוגמה:
עבור המערך [5,2,1,3,10,4] הפונקציה תחזיר 3,4,5

ב. ציינו באיזו שיטה השתמשת, הסבירי את נכונות האלגוריתם ונתחי את זמן ריצתו.

בס"ד

.3

שנוכל מעט מה להיות ממוין, לא שם להצבה מה שמניתי

מערך A עם n איברים נקרא **כמעט ממויין k** ($k<n$) אם $A[j] \geq A[i]$, לכל i,j המקיימים $j-i>k$. במילים אחרות, המערך לא חייב להיות ממויין, אבל כל שני איברים הנמצאים בסדר הפוך, לא יכולים להיות רחוקים זה מזה יותר מ-k מקומות.

k לא קבוע קטן.

<u>סעיף א (8 נקודות):</u>

1. (2 נקודות) תן דוגמה למערך כמעט ממויין 3 בעל 8 תאים, אשר **איננו** מערך כמעט ממויין 2.

2. (2 נקודות) מערך כמעט ממויין 1 הינו מערך ממוין.

12345

(סמן התשובה הנכונה). נכון / לא נכון

3. (2 נקודות) בהינתן האלגוריתם הבא:

```
1. for(i=0;i<n; i = i+k){
2. sort (A,i,i+k-1)
3. }
```

אשר ממיין k-איברים עוקבים ב-A בצורה סדרתית.

<u>לדוגמא:</u> עבור המערך מסעיף א', האלגוריתם ימיין את האיברים 0,1,2, ולאחר מכן את האיברים 3,4,5 וכן הלאה.

האם האלגוריתם הופך מערך כמעט ממויין k לממויין? אם כן – הוכח, אם לא ספק דוגמא נגדית.

4. (2 נקודות) מהו זמן הריצה של האלגוריתם, בהנחה שהפונקציה sort פועלת על m איברים בזמן $O(m\log m)$?

k לא קבוע קטן

<u>סעיף ב (11 נקודות):</u>

הצע אלגוריתם המבוסס על מיון מהיר אשר מקבל מערך והופך אותו לכמעט ממויין k.

האלגוריתם החדש צריך להיות יעיל יותר מהאלגוריתם המקורי. מהו זמן הריצה האסימפטוטי של האלגוריתם החדש במקרה הטוב ביותר ?

רמז: איזו שורה לשנות במיון מהיר על מנת שיבצע את הנדרש?

***רשות**: האם יש שינוי בזמן הריצה של המיון החדש? אם כן פרטי והסבירי.*

mod = 0

4. נתון מערך עם מספרים שלמים. יש למצוא את גודל תת הקבוצה הגדולה ביותר המכילה איברים כך שעבור כל 2 איברים X ו-Y בקבוצה מתקיים $X \bmod Y = 0$ או $Y \bmod X = 0$

לדוגמה במערך: [1,5,3,4,10,20,100,45]

תת הקבוצה הגדולה ביותר הינה: ,1,5,10,20,100 שגודלה 5.

א. כתבי פונקציה בפסאודוקוד או בכל שפת תכנות הנוחה לך, המקבלת את המערך ומחזירה את גודל תת הקבוצה המקסימלי האפשרי.

ב. ציני באיזו שיטה השתמשת, הסבירי ונתחי את סיבוכיות זמן הריצה של האלגוריתם.

צריך למיין אותם.

5. נתון האלגוריתם הבא:

```
PartialSums(A,ℓ,r)
n ← r- ℓ +1
if r<l then
        return 0
sum ← 0
for i ← ℓ to r do
        sum ← sum + A[i]
print(sum)
i ← ℓ
while i+⌊√n⌋ − 1 <= r do
        PartialSums(A,i,i+⌊√n⌋ − 1)
        i ← i+ ⌊√n⌋
```

כתבי נוסחת נסיגה לחישוב זמן הריצה של האלגוריתם ופתרי אותה.

רשות*: בהינתן מערך שרוצים לבצע עליו שאילתות רבות לבדיקת "סכום חלקי" מ- i עד j במערך. האם ניתן לבצע עיבוד מקדים כך שזמן הריצה של השאילתה יהיה $O(1)$ [יותר יעיל ממה שנעשה באלגוריתם כאן] ?*

בהצלחה!!!