

הטכניון – הפקולטה למדעי המחשב מבוא למדעי המחשב מ' / ח' (234117 / 234114)

סמסטר אביב תשע"א
מבחן מועד א' - 3.7.2011

צוות הקורס :

מרצים: ד"ר ניר אילון (מרצה אחראי), ד"ר רועי אנגלברג
מתרגלים: ישראל גוטר (מתרגל אחראי), ניצן חריזמן, אלכסנדר נוס, מאור גרינברג,
נתנאל רביב, שי גרץ, חביאר טורק, אריאלה וולושין

משך הבחינה – 3 שעות.

השימוש בחומר עזר כלשהו, כתוב או אלקטרוני, **אסור**.

שאלה 1 (20 נקודות) :

בשאלה זו 2 סעיפים בלתי תלויים זה בזה.

סעיף א :

```
char g1(char *s, int n);  
char f1(char *s, int n)  
{  
    if (n == 1)  
        return *s;  
    return g1(s, n-1);  
}  
char g1(char *s, int n)  
{  
    int i;  
    char t;  
    for (i = 0; i < n/2; i++) {  
        t = *(s + i);  
        *(s+i) = s[n-1-i];  
        *(s+n-1-i) = t;  
    }  
    return f1(s, n);  
}
```

(הנחת קלט של f1 : n מספר חיובי)

מהי סיבוכיות הזמן של f1 במקרה הגרוע כתלות ב-n ? $\Theta(\text{_____})$

מהי סיבוכיות המקום הנוסף של f1 במקרה הגרוע כתלות ב-n ? $\Theta(\text{_____})$

מה ידפיס בלוק הקוד הבא?

```
{ char s[5] = {'4','5','2','6','9'};  
  printf("%c\n", f1(s, 5)); }
```

תשובה: _____

סעיף ב :

```
int f2(int n)  
{  
  int x, y, z=0, i;  
  for (x=n, i=0; i<n; i++, x*=n)  
  {  
    y=x;  
    while (y>1)  
      y/=3;  
    z+=y;  
  }  
  return z;  
}
```

(הנחת קלט של n מספר חיובי)

מהי סיבוכיות הזמן של $f2$ כתלות ב- n ? $\Theta(\text{_____})$

מהי סיבוכיות המקום הנוסף של $f2$ כתלות ב- n ? $\Theta(\text{_____})$

הציעו תנאי בדיקה בלולאת for אשר מתייחס ל- i (במקום התנאי הנוכחי $i < n$)

כך שסיבוכיות הזמן תהיה $\Theta(n \cdot \log n)$

התנאי המוצע יכול להיות כל ביטוי שבו, בין היתר, מוזכר גם i , למשל:

$i + n \neq 10$, $i * n \geq i + 1$ וכי'.

הערה : אם סיבוכיות זו מתקיימת גם עבור התנאי הקיים $i < n$, ניתן לתת את

התנאי הקיים כתשובה.

תשובה: _____

שאלה 2 (25 נקודות) :

כתבו פונקציה בעלת החתימה הבאה :

```
int handle_text(char *st)
```

הפונקציה מוחקת מהטקסט הנתון ב st רווחים מרובים סמוכים, ומשאירה רווח בודד בכל מקום שבו יש רווחים. הפונקציה מחזירה את מספר התווים שהוסרו ממחרוזת הקלט. **דוגמה**: הניחו שלפני הטיפול במחרוזת כלשהי s , קריאה ל- `printf(s)` מדפיסה:

		H	e	l	l	O				w	o	r	l	d		
--	--	---	---	---	---	---	--	--	--	---	---	---	---	---	--	--

או בלוק הקוד `{x = handle_text(s); printf(s);}` יגרום הדפסת:

	H	e	l	l	O		w	o	r	l	d	
--	---	---	---	---	---	--	---	---	---	---	---	--

ויקבע את הערך של המשתנה x להיות 4.

שימו לב:

1. כפי שרואים בדוגמה, אין להתייחס בצורה מיוחדת לרווחים המופיעים בהתחלה או בסוף המחרוזת, כלומר, גם רווחים סמוכים בתחילת מחרוזת הקלט או בסופה יהפכו לרווח אחד **בדיוק** בפלט.

2. יש לטפל רק בתו הרווח ' ', **ואין לטפל** בתווים אחרים הנראים כרווחים (לדוגמה `tab`).

3. אין לדאוג לניקוי או שחרור זיכרון עודף שנותר כתוצאה מקיצור המחרוזת.

דרישות:

סיבוכיות זמן $O(n)$ (n – אורך המחרוזת).

סיבוכיות מקום נוסף $O(1)$.

שאלה 3 (30 נקודות):

נתון מערך a של מספרים שלמים, באורך n . ניתן להניח ש- n הוא חזקה שלמה חיובית של 2, כלומר $n=2^k$ עבור $k>0$ כלשהו.

המערך ייקרא ממוין בממוצעים אם מתקיים:

1. המערך הוא באורך 2 וגם $a[0] \leq a[1]$.

או:

2. המערך הוא באורך 2^k עבור $k>1$, וגם:

א. החצי השמאלי של המערך כשלעצמו הוא ממוין בממוצעים.

ב. החצי הימני של המערך כשלעצמו הוא ממוין בממוצעים.

ג. הממוצע של החצי השמאלי אינו גדול מהממוצע של החצי הימני.

דוגמא: הסדרה הבאה:

14	21	20	30
----	----	----	----

ממוינת **בממוצעים**. לחצי השמאלי יש ממוצע 17.5. לחצי הימני יש ממוצע 25. מתקיים $17.5 \leq 25$, כדרוש. בנוסף, החצי השמאלי כשלעצמו ממוין בממוצעים כי מתקיים $14 \leq 21$, החצי הימני כשלעצמו ממוין בממוצעים כי $20 \leq 30$. לעומת זאת, הסדרה הבאה:

90	110	40	50
----	-----	----	----

אינה **ממוינת בממוצעים** כי הממוצע של החצי השמאלי הוא 100 בעוד הממוצע של הימני 45, ומתקיים $100 > 45$.

כתבו פונקציה שחתימתה:

```
int check_sorted_in_averages(int a[], int n)
```

הפונקציה מקבלת מערך a באורך n , כאשר n חזקה שלמה חיובית של 2, ומחזירה 1 אם הוא ממוין בממוצעים, אחרת 0. אין לשנות את אברי מערך הקלט בחזרה מהפונקציה.

דרישות: סיבוכיות זמן $O(n)$, מקום $O(\log n)$.

הערה: פתרון נכון בסיבוכיות זמן $O(n \log n)$ ומקום $O(\log n)$ או בסיבוכיות זמן $O(n)$

ומקום $O(n)$ יזכה ברוב הנקודות המוקצות לשאלה זו.

שאלה 4 (25 נקודות):

משורר נודע הסובל מחוסר השראה מעוניין לכתוב שיר חדש. כידוע, החלק הקשה בכתיבת שיר הוא מציאת המילים האחרונות המתחרזות של כל שורה (להלן: "המילים המתחרזות" או endwords). עליכם לכתוב פונקציה `find_endwords` בעלת החתימה הבאה:

```
void find_endwords(  
    Word dictionary[], int dictionary_size, int length)  
שתדפיס את כל האפשרויות לסדרה של length מילים מתחרזות של שיר, כאשר המילים באות ממילון נתון dictionary המכיל dictionary_size מילים. הניחו כי length זוגי. הטיפוס Word מייצג מילה בודדת בתרגיל זה. התנאים הבאים הינם הכרחיים ומספיקים עבור סדרת מילים סופיות של שיר חוקי:
```

1. המילה הראשונה בסדרה מתחרזת עם השניה, השלישית עם הרביעית, החמישית עם השישית וכן הלאה. בנוסף, המילה השניה אינה מתחרזת עם השלישית, הרביעית אינה מתחרזת עם החמישית, השישית אינה מתחרזת עם השביעית וכן הלאה. לרשותכם עומדת פונקציה:

```
int is_rhyme(Word w1, Word w2)
```

המחזירה 1 אם שתי המילים הנתונות מתחרזות, אחרת 0.

2. מספר המילים המתארות אהבה הוא לפחות שניים אך לא יותר מארבעה. עומדת לרשותכם פונקציה:

```
int is_love(Word w)
```

המחזירה 1 אם מילה w מתארת אהבה, אחרת 0.

לצורך הפלט, הניחו שעומדת לרשותכם הפונקציה:

```
void print_words(Word words[], int length)
```

המדפיסה סדרה words של length מילים.

יש להשתמש ב – backtracking תוך שימוש בכל כללי המילים המתחרזות של שיר חוקי כפי שהוגדר לעיל.