



מבוא למדעי המחשב מ' / ח' (234114 / 234117)

סמסטר אביב תשע"א - מבחן מועד ב'

סמסטר קיץ תשע"א - מבחן מועד א'

15.9.2011

--	--	--	--	--	--	--	--	--	--

שם פרטי

שם משפחה

מספר סטודנט

• רשום/ה לקורס: 234114 / 234117 סמסטר אביב / קיץ

• תואר ראשון / לימודי חוץ / אחר (לפרט): _____

- משך הבחינה – 3 שעות.
- בדקו שיש 18 עמודים (4 שאלות) במבחן, כולל עמוד זה.
- השימוש בחומר עזר כלשהו, כתוב או אלקטרוני, **אסור**.
- ניתן להשתמש בפונקציות קלט-פלט סטנדרטיות והקצאת זיכרון ב-C. שימוש בכל פונקציה אחרת, לרבות כזו שהוגדרה במהלך הקורס, אסור. אתם יכולים להגדיר פונקציות עזר כרצונכם. אין צורך להצהיר עליהן.
- כל זיכרון שאתם מקצים, אתם חייבים בשחרורו.
- אין צורך בהוכחות לחישובי הסיבוכיות.
- אין להניח חסם כלשהו על ערך int ב-C. יחד עם זאת, יש להניח שכל פעולה אריתמטית, השוואה, כתיבה וקריאה מהזיכרון של int לוקחות זמן קבוע, וכמך כן ש-int תופס מקום קבוע.
- ניתן לכתוב בעיפרון ולהשתמש במחק.

צוות הקורס :

סמסטר אביב תשע"א :

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

סמסטר קיץ תשע"א :

מרצה: פרופ' מיכאל אלעד

מתרגל: נתנאל רביב

שאלה	ערך	ציון	בדק
1	20		
2	30		
3	20		
4	30		
סה"כ	100		

בהצלחה !



שאלה 1 (20 נקודות) :

נתונה הפונקציה הבאה `naive_find_all_occurrences` (המשתמשת בפונקציית `naive_match_string`) המקבלת מחרוזת `text` ומחרוזת `word` ומדפיסה את כל המופעים של המילה `word` במחרוזת.

```
int naive_match_string(char* text, char* word);

int naive_find_all_occurrences(char *text, char *word)
{
    int i = 0;
    while (*(text + i) != '\0') {
        if (naive_match_string(text + i, word))
            printf("Found word at position %d\n", i);
        i = i + 1;
    }
}

int naive_match_string(char* text, char* word)
{
    while (*text != '\0' && *word != '\0') {
        if (*text != *word)
            return 0;
        ++text;
        ++word;
    }
    if (*text == '\0' && *word != '\0')
        return 0;
    return 1;
}
```

א. (8 נקודות)

מהי סיבוכיות זמן הריצה במקרה הגרוע של הפונקציה `naive_find_all_occurrences` כתלות ב- n (אורך `text`) וב- m (אורך `word`) ?

א.1 בהנחה ש- $n \geq m$ $\Theta(\text{_____})$

א.2 בהנחה ש- $n < m$ $\Theta(\text{_____})$



ב. (5 נקודות)

כיצד תשתנה התשובה בסעיף א. (אם בכלל) אם ידוע שהמחרוזת word נראית כך:
abb.....b (כלומר האות הראשונה היא a, ואחריה m-1 פעמים האות b).
יש להתייחס רק למקרה שבו $n \geq m$

Θ (_____)

ג. (5 נקודות)

כיצד תשתנה התשובה בסעיף א. (אם בכלל) אם ידוע שהמחרוזת word נראית כך:
b.....ba (כלומר m-1 פעמים האות b ולאחר מכן את אחרונה a).
יש להתייחס רק למקרה שבו $n \geq m$

Θ (_____)

ד. (2 נקודות)

מהי סיבוכיות המקום הנוסף של הפונקציה naive_find_all_occurrences
כתלות ב- n (אורך text) וב- m (אורך word) ?

Θ (_____)



שאלה 2 (30 נקודות) :

סעיף א :

ממשו את הפונקציה

```
void flip_halves(int* v, int n)
```

שעושה את הפעולה הבאה :

- אם n זוגי, היא מחליפה את $n/2$ האברים הראשונים עם $n/2$ האחרונים
- אם n אי-זוגי, היא מחליפה את $(n-1)/2$ האברים הראשונים עם $(n-1)/2$ האחרונים

לדוגמה, אם המערך v הוא הבא :

1	2	3	4	5	6
---	---	---	---	---	---

אז קריאה ל `flip_halves(v, 6)` תשנה אותו כדלקמן :

4	5	6	1	2	3
---	---	---	---	---	---

ואם המערך v הוא הבא :

1	2	3	4	5
---	---	---	---	---

אז קריאה ל `flip_halves(v, 5)` תשנה אותו כדלקמן :

4	5	3	1	2
---	---	---	---	---

סיבוכיות נדרשת : זמן $O(n)$, מקום נוסף $O(1)$.

```
void flip_halves(int* v, int n) {
```

[illegible]



הטכניון, מכון טכנולוגי לישראל
מבוא למדעי המחשב מ'/'ח'

[illegible]



סעיף ב :

מה עושה הפונקציה הבאה ומה סיבוכיות הזמן וסיבוכיות המקום הנוסף שלה ?

```
void foo(int* v, int n) {  
    if (n <= 1)  
        return;  
  
    flip_halves(v, n);  
    foo(v, n/2);  
    foo(v + n - n/2, n/2);  
}
```

הפונקציה מבצעת :

Θ (_____)

סיבוכיות זמן כפונקציה של n

Θ (_____)

סיבוכיות מקום נוסף כפונקציה של n



סעיף ג :

כתבו פונקציה foo2 שעושה בדיוק מה ש-foo עושה, ופועלת בסיבוכיות זמן $O(n)$ ובסיבוכיות מקום נוסף $O(1)$.

```
void foo2(int* v, int n) {
```




הטכניון, מכון טכנולוגי לישראל
מבוא למדעי המחשב מ'/ח'

[illegible]





הטכניון, מכון טכנולוגי לישראל
מבוא למדעי המחשב מ'/ח'

[illegible]



שאלה 4 (30 נקודות) :

מכונית חשמלית עומדת בנקודת התחלה (נקודת ה"אפס") לאורך כביש ורוצה להגיע לנקודת הסיום הממוקמת 100 ק"מ מנקודת האפס. לאורך הכביש n תחנות טעינה. התחנה ה- i נמצאת במרחק $dist[i]$ קילומטרים מנקודת האפס. כמו כן, נתון שהתחנה ה- i יכולה להטעין את סוללת המכונית עם תוספת אנרגיה המספיקה לנסיעה של $battery[i]$ קילומטרים. המטרה היא לתכנן רשימת עצירות טעינה שיאפשרו למכונית להגיע לנקודת הסיום.

עליכם לכתוב פונקציה :

```
void print_all_solutions(int *dist, int* battery, int n)
```

שיתדפיס את כל הפתרונות האפשריים. הנחות :

1. הסוללה של המכונית ריקה בתחילת הדרך, וכן $dist[0]=0$ (כלומר יש תחנה בתחילת הדרך).
2. אין מגבלה על כמות האנרגיה השמורה בסוללה.
3. עצירה בתחנה i משמעותה תוספת אנרגיה המתאימה ל- $battery[i]$ קילומטרים בדיוק, כלומר, לא ניתן לטעון באופן חלקי בתחנה. אם לדוגמה הרכב נכנס לתחנה i כשבסוללה נותרה אנרגיה ל- 10 ק"מ, אז ביציאה מהתחנה תהיה בסוללה אנרגיה ל- $10+battery[i]$ ק"מ.
4. לצורך הדפסה יפה של הפלט, הניחו שקיימת פונקציית שירות

```
void print_stops(int* stops, int n)
```

אשר מקבלת מערך $stops$ של n ערכי אמת (כלומר 0 או 1) ומדפיסה אותו כפיתרון. המשמעות של ערך 1 במקום ה- i במערך היא שהמכונית עוצרת בתחנה ה- i , וערך 0 משמעו שהיא אינה עוצרת שם. הפונקציה אינה בודקת את חוקיות הפיתרון. (אם ברצונכם לייצג פתרונות בצורה שונה, עליכם לממש פונקציית הדפסה סבירה בעצמכם).

דוגמה :

עבור המערכים הבאים בגודל $n=5$,

	0	1	2	3	4
dist	0	20	40	70	90
battery	60	40	60	10	0
solution A	1	0	1	0	0
solution B	1	0	0	1	0

פתרון חוקי הוא, לדוגמה, עצירות בתחנות 0 ו-2 (ראו solution A) מאחר שתחנה 0 מאפשרת לרכב לנסוע 60 ק"מ, וזה מספיק על מנת להגיע לתחנה 2 במרחק 40 ק"מ, ולהישאר עם עודף של אנרגיה לעוד 20 ק"מ. בתחנה 2 הרכב נטען בעוד אנרגיה ל- 60 ק"מ, ויוצא לדרך עם סה"כ אנרגיה ל- 80 ק"מ. זה מספיק על מנת להגיע לקו הסיום.

פתרון לא חוקי הוא, לדוגמה, עצירות בתחנות 0 ו-3 בלבד (ראו solution B), מאחר שהרכב ייתקע עם סוללה ריקה לפני שיגיע לתחנה 3 הנמצאת במרחק 70 ק"מ לאחר שטען רק אנרגיה ל- 60 ק"מ בתחנה 0.

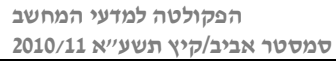


```
void print_all_solutions(int *dist, int* battery, int n) {
```



הטכניון, מכון טכנולוגי לישראל
מבוא למדעי המחשב מ'/ח'

[illegible]



הטכניון, מכון טכנולוגי לישראל
מבוא למדעי המחשב מ'/ח'

[illegible]

[illegible]



הטכניון, מכון טכנולוגי לישראל
מבוא למדעי המחשב מ'/ח'

[illegible]



הטכניון, מכון טכנולוגי לישראל
מבוא למדעי המחשב מ'/ח'

[illegible]