המכללה האקדמית להנדסה
אורט בראודה
ORT BRAUDE COLLEGE

המחלקה להנדסת חשמל ואלקטרוניקה

מבני נתונים ואלגוריתמים 31632

מבחן סופי, מועד ב', סמסטר א' תשע"ד, 24/02/2014

<u>הוראות לנבחן:</u> משך הבחינה שלוש שעות. חומרי העזר המותרים הם השקפים של הקורס שנמסרו באתר הקורס בלבד. שימוש במחשבונים ומכשירים אלקטרוניים אסור בהחלט. רשום את תשובותיך במחברת המצורפת לבחינה. ציין בבירור את מספר השאלה במחברת והשתדל להיות <u>קצר וענייני</u> (תשובות ארוכות מדי או נסיונות לתת מספר תשובות אפשריות <u>ייפסלו</u> את תשובתך!). הקפד על כתב יד ברור ומסודר, ומחק את כל חומר הטיוטא. השימוש בשפה האנגלית מותר. השאלון מכיל 6 שאלות, ונפרש על פני 3 עמודים.
בהצלחה !

<u>שאלה 1</u> [20%]

להלן מיפרט ADT של מבנה נתונים בשם Points. זהו מבנה נתונים המורכב מאוסף של נקודות במישור הגאומטרי הדו-מימדי. כל נקודה במישור תיוצג על ידי זוג סדור (x,y), כאשר x,y הם מספרים ממשיים (float):

```
p = Points()
   Create a new empty Points object p
p.add(x,y)
   Add a new point (x,y)
p.delete(x,y)
   Remove the point (x,y) if it is in p, otherwise do nothing
p.contains(x,y)
   Return True if p contains the point (x,y), otherwise returns False
p.xsection(c)
   Return list of all points (x,y) in p, such that x==c
p.ysection(c)
   Return list of all points (x,y) in p, such that y==c
```

א. כתוב מחלקה בשם Points בשפת Python אשר מממשת את המיפרט (ADT) הנ"ל באמצעות מבני הנתונים הבסיסיים של Python בלבד (כגון: list, set, tuple, dictionary).

ב. מהי הסיבוכיות זמן של פעולות xsection ו-ysection שקיבלת במימוש שלך?

ג. האם ניתן לממש את מבנה הנתונים Points באופן שונה כך שסיבוכיות פעולות xsection ו-ysection תהיה O(1)? אם כן, הצע רעיון כיצד זה ניתן לביצוע. אם לא, הסבר מדוע זה בלתי אפשרי?

<u>שאלה 2</u> [6%]

נתונה הרשימה הבאה למיון:

L = [3, 2, 1, 6, 5, 4, 9, 8, 7, 12, 11, 10, …, 303, 302, 301]

כמה סבבים של האלגוריתם bubble_sort יידרשו בכדי למיין את L? כמה פעולות החלפת איברים יידרשו?

בצד שמאל רשום האלגוריתם bubble_sort הדרוש לביצוע המיון של הרשימה L. אין צורך לנמק. נדרשות תשובות סופיות בלבד.

<u>הערה:</u> סבב מוגדר כפעולה מלאה של גוף הלולאה while של האלגוריתם.

```
def bubble_sort(L):
    N = len(L)
    while True:
        sorted = True
        for i in range(0,N-1):
            if L[i+1] < L[i]:
                sorted = False
                L[i], L[i+1] = L[i+1], L[i]
        if sorted:
            return
```

שאלה 3 [32%]

בכל האלגוריתמים של שאלה זו, מותר להשתמש במתודות של list בלבד! (אין להשתמש ב- set או dict!)

א. רשום אלגוריתם יעיל ככל האפשר בשם zero_indices(L) בשפת Python המקבל רשימת מספרים שלמים L, ומחזיר זוג אינדקסים (i,j) שעבורם L[i]+L[j]==0 אם ישנם כאלה, אחרת יחזיר None.
לדוגמא:

```
L1 = [7, -2, 5, -6, 1, 2]
L2 = [2, 1, 9, -5]
zero_indices(L1) = (1, 5)
zero_indices(L2) = None
```

ב. הערך מהי סיבוכיות הזמן של האלגוריתם שמצאת בסעיף הקודם?

ג. האם האלגוריתם שמצאת הוא היעיל ביותר שייתכן? אם אתה סבור שלא, נסה להציע רעיון לאלגוריתם יעיל יותר.

ד. בנה אלגוריתם יעיל ככל האפשר בשם lequal(A,B) המקבל שתי רשימות של מספרים שלמים A, B, ומחזיר ערך בוליאני True אם הרשימות מכילות את אותן איברים (לא בהכרח באותו סדר או מספר כפילויות). במידה ולא, האלגוריתם יחזיר False. לדוגמא:

```
A = [8, 3, -5, 17, 3, -5, 3, 8, 3]
B = [3, 17, -5, 17, 8, 17, 17]
C = [17, -5, 1, 3, 8]
lequal(A,B) = True
lequal(A,C) = False
```

ה. הערך מהי סיבוכיות הזמן של האלגוריתם שמצאת בסעיף הקודם?

ו. האם האלגוריתם שמצאת עבור lequal(A,B) היעיל ביותר שייתכן? אם אתה סבור שלא, נסה להציע רעיון לאלגוריתם יעיל יותר.

הערה: בכל האלגוריתמים של שאלה זו, מותר להשתמש במתודות של list בלבד! (אין להשתמש במבני נתונים set או dict!). זוג האינדקסים (i,j) אינו חייב להיות על פי סדר כלשהו וגם אין הכרח ש-i≠j.

שאלה 4 [12%]

נתון גרף לא-מכוון g אשר קודקודיו הם המספרים [1,2,3,4,5,6,7] וצלעותיו נתונים על ידי טבלת שכנות משמאל.

Vertex	Adjacent Vertices
1	2, 3, 4
2	1, 4, 7
3	1, 4
4	1, 2, 3
5	7
6	7
7	2, 5, 6

א. שרטט את הגרף g באופן ברור במחברת.

ב. בצע אלגוריתם DFS שמתחיל בצומת 1 ובכל שלב שבו האלגוריתם צריך להתקדם לצומת הבא מתוך רשימת צמתים אפשריים, בחר את הצומת הראשון על פי הסדר הטבעי של המספרים. שרטט באופן ברור את העץ הפורש המתקבל כתוצאה מכך.

ג. רשום את רשימת הקודקודים על פי סדר הביקור (visit) שהאלגוריתם DFS ביצע (אין צורך להסביר).

שאלה 5 [15%]

להלן מיפרט ADT (חלקי) של מבנה נתונים מסוג עץ שנקרא SimpleTree בשפת Python.

```
t = SimpleTree()
    Create a new empty SimpleTree object
t.get_root()
    Return the root position of the tree (or None if tree is empty)
t.get_parent(p)
    Return the position of p's parent (or None if p is root)
t.get_children(p)
    Return the position of p's children
t.num_children(p)
    Return the number of children of position p
t.add_root(e)
    Place element e at the root of an empty tree
    Raise ValueError if tree nonempty
t.add_child(p, e)
    add a new element e at the end of the children of p
    Return the position of the new node
```

הנחת עבודה: המחלקה SimpleTree מומשה במלואה (בצרוף מתודות נוספות שאינן רלבנטיות כאן), וסיבוכיות הזמן של כל המתודות היא O(1).

ענה על השאלות הבאות:

א. בנה אלגוריתם יעיל ככל האפשר בשם bignodes(t) בשפת Python המחשב את מספר כל המפתחות בעץ שיש להם יותר משני ילדים. עליך להשתמש במתודות הנ"ל בלבד.

ב. עץ t נקרא אחיד (uniform) אם לכל שני אחים (שני צמתים בעלי אבא משותף) יש אותו מספר צאצאים (descendants). בנה אלגוריתם יעיל ככל האפשר is_uniform(t) בשפת Python המקבל עץ t (מסוג SimpleTree) ומחזיר True אם t אחיד, False אם t אינו אחיד. עליך להשתמש במתודות הנ"ל בלבד.
מהי סיבוכיות הזמן של האלגוריתם שבנית?

הערה: מומלץ להשתמש בפונקציות עזר לשם פישוט הקוד.

שאלה 6 [15%]

א. רשום אלגוריתם find_cycle3(g) בשפת Python אשר מקבל גרף מכוון g ומחפש מחזור (cycle) באורך 3. אם נמצא מחזור כזה בגרף g, האלגוריתם יחזיר את המחזור הראשון שימצא (כרשימת קודקודים), אחרת האלגוריתם יחזיר None. בכתיבת האלגוריתם תוכל להשתמש בכל המתודות של המחלקה Graph המוצגת בשקפים של הקורס.

ב. נתון גרף מכוון g (directed graph) שבו לכל קודקוד יש לכל היותר 5 קודקודים שכנים (יוצאים ונכנסים). מה תהיה סיבוכיות הזמן (worst case) של האלגוריתם במקרה כזה? התבסס על התכונות של מבנה הנתונים של גרף שבנינו בקורס (עליך להכיר את סיבוכיות הזמן של כל מתודה של גרף).

הערה: מחזור בגרף מכוון היא רשימת קודקודים באורך 2 או יותר [v0, v1, …, vn] שבה עבור כל i<n יוצאת קשת מקודקוד vi לקודקוד vi+1, ובסוף יוצאת קשת מקודקוד vn לקודקוד v0.