

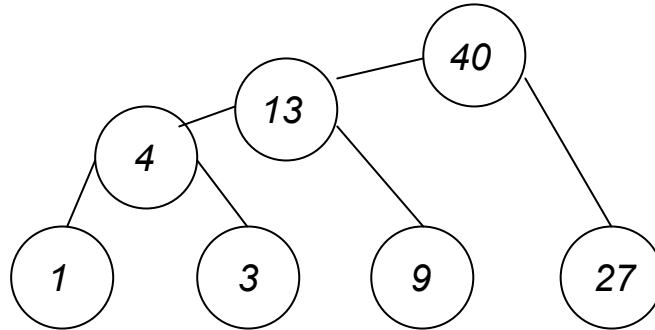
# פתרון חלקי למבחן בקורס מבוא מורחב למדעי המחשב,

סמסטר ב' 2013

## מועד א

### שאלה 1 (20 נק')

א. (6 נק')



ב. (7 נק')

$$1 + 2 + 3 + \dots + (m - 1) + (m - 1) =$$

$$\frac{m(m - 1)}{2} + m - 1 = \frac{m^2 + m - 2}{2}$$

ג. (7 נק')  $k=9$ .

### שאלה 2 (20 נק')

א. (4 נק')  $2^4=16$

ב. (4 נק') 4

ג. (12 נק')

מספר 1-ים	יש שגיאה בקוד המינג (7 ביטים ראשונים)	מספר השגיאות	ניתן לתיקון? (כן/לא)	באיזה חלק יהיה התיקון?
זוגי	יש	2	לא	
זוגי	אין	0		
איזוגי	יש	1	כן	ב-7 ביטים ראשונים
איזוגי	אין	1	כן	בביט זוגיות

הערה: הניקוד שהופחת לטעויות בטבלה איננו אחיד. למשל, אם תשובה איננה עקבית עם סעיף ב', או אם יש סתירה פנימית בתוך אותה שורה – ירדו יותר נקודות.

### שאלה 3 (20 נק')

א. (8 נק')

```
def horiz_edge(A, k=1):
    n,m = A.dim()
    edge = Matrix(n,m,255) #a new nXm matrix with value 255 everywhere
    for i in range(1, n - 1):
        for j in range(k, m - k):
            summ = 0
            for t in range(-k, k+1):
                summ += abs(A[i+1, j+t] - A[i-1, j+t])
            summ /= 2 * k + 1
            Edge[i,j] = 0 if summ > 80

return edge
```

ב. (4 נק')  $O(mn)$ .

ג. (8 נק')

```
def pixel_count(edgelist, n):
```

```
    rows = [0] * n
    for pixel in edgelist:
        rows[pixel[0]] += 1

    return rows
```

### שאלה 4 (20 נק')

א. השלימו את הגדרת רקורסיבית של הפונקציה acc :

```
def acc(f, v, lst):
    if ____len(lst) == 0 ____:
        return ____v____
    else:
        return ____acc(f, f(v, lst[0]), lst[1:])____
```

ב. השלימו את הגדרת הפונקציה myProd אשר מקבלת רשימת מספרים ומחשבת את מכפלתם :

```
def myProd(lst):
    return acc(lambda x,y: x * y, 1, lst)
```

```
def g( func1, func2 ):
    return lambda x: func1(func2(x))
```

```
def h( x ):
    return x
```

הערה : במהלך המבחן נמסר תיקון לשאלה, ולאחר מכן התיקון בוטל (בניסוח השאלה לא היתה כל שגיאה). עקב זאת, ניתן ניקוד מלא לפתרונות של סעיף אי שמבצעים את החישוב בסדר לא נכון.

### **שאלה 5 (20 נקודות)**

- א.  $O(n^2)$  – כל המספרים באותו דלי ובחירת pivot תמיד מיני או מקסי  
 ב.  $O(n)$  – כל דלי מכיל מספר אחד בלבד  
 ג.

```
def bucket_sort(lst, a=100):
    n = len(lst)
    # init buckets
    buckets = [ [] for i in range(n) ]
    # scatter
    for num in lst:
        buckets[ n * num // a ].append(num)
    # sort and gather
    sorted_list = []
    for b in buckets:
        if len(b) > 1:
            # quicksort(lst) returns a sorted copy of lst
            b = quicksort(b)
        sorted_list.extend(b)
    return sorted list
```