

מבחן בקורס "מבוא כללי לתכנות ולמדעי המחשב" (הקבץ הסייבר)

סמסטר א' 6-2015

מועד א'

אמיר רובינשטיין, אורית מוסקוביץ'

משך הבחינה: שתיים.

חומר עזר מותר: שני דפי A4 דו צדדיים. אסור השימוש בכל חומר עזר אלקטרוני (מחשב, מחשבון, טלפון) או ביולוגי (חברים).

- יש לכתוב את כל התשובות בטופס בחינה זה. מחברות הטיוטה לא ייאספו.
- יש לכתוב את כל התשובות במקום המוקצב ובכתב קריא. חריגות משמעותיות מהמקום המוקצב, תשובות הכתובות בכתב קטן / לא ברור או תשובות שדורשות מאמצים רבים להבנתן עלולות לגרור הורדת ציון.
- במבחן 9 עמודים ו-8 שאלות – בידקו שכולם בידכם.
- מומלץ לא "להיתקע" על אף שאלה בודדת, אלא להמשיך לשאלות אחרות ולחזור לשאלה אח"כ.
- קיראו היטב את השאלות. הקפידו לענות בדיוק על מה שנשאלתם, ולנמק אם נדרשתם.

נא לרשום מספר תעודת זהות (ללא שם):

בהצלחה !

שאלה 1 (10 נק')

להלן בדיחה קצרה:

מתמטיקאית ומהנדס נמצאים על אי בודד. הם מוצאים שני עצי דקל, ועל כל אחד מהם קוקוס אחד. המהנדס מטפס על עץ מספר 1, מוריד את הקוקוס ואוכל אותו. המתמטיקאית עולה על עץ מספר 2, מורידה את הקוקוס, מטפסת על עץ מספר 1, שמה עליו את הקוקוס, יורדת מהעץ ומצהירה בשמחה: "הבעיה נפתרה!".

תזכורת: משתמשים ברדוקציה כדי לעבור מבעיה A לבעיה B. אם ידוע כי לבעיה B יש פתרון, נשתמש בו על מנת לפתור את בעיה A.

הסבירו את הבדיחה – מדוע המתמטיקאית המתוחכמת העבירה את הקוקוס מעץ אחד לשני? השתמשו במונח רדוקציה. פרטו כיצד נעשתה הרדוקציה, ובפרט, הסבירו מי הבעיה A, ומי הבעיה B.

שאלה 2 (10 נק')

בהינתן מספר שלם כלשהו n , רישמו פקודה בפייתון שמדפיסה את ספרת המאות שלו.

למשל עבור $n=1234$ יודפס 2. מותר להניח כי למספר לפחות 3 ספרות. אפשר לרשום שורה בודדת או פקודות בשתי שורות נפרדות, כרצונכם.

שאלה 3 (10 נק')

כידוע, בני אדם סופרים בד"כ בבסיס 10 (עשרוני), ואילו מחשבים בבסיס 2 (בינארי). בשאלה זו נתמקד במספרים בבסיס 3.

תזכורת: מספר בבסיס 2 (בינארי) מורכב מהספרות 0,1 בצורה הבאה :

$$\underbrace{1}_{2^4} \underbrace{0}_{2^3} \underbrace{0}_{2^2} \underbrace{1}_{2^1} \underbrace{1}_{2^0} = 2^0 \cdot 1 + 2^1 \cdot 1 + 2^2 \cdot 0 + 2^3 \cdot 0 + 2^4 \cdot 1 = 19$$

באופן דומה, נגזיר: מספר בבסיס 3 (טרינארי) מורכב מהספרות 0,1,2 בצורה הבאה :

$$\underbrace{2}_{3^4} \underbrace{0}_{3^3} \underbrace{1}_{3^2} \underbrace{2}_{3^1} \underbrace{1}_{3^0} = 3^0 \cdot 1 + 3^1 \cdot 2 + 3^2 \cdot 1 + 3^3 \cdot 0 + 3^4 \cdot 2 = 178$$

כלומר, כל ספרה (0,1 או 2) מקבלת משקל השווה לחזקה של 3, בהתאם למיקום שלה.

א) נתון מספר עשרוני המתחלק ב-3 ללא שארית. מה ניתן לומר על הספרה הימנית ביותר שלו, בכתיב טרינארי? הקיפו בעיגול.

1. חייבת להיות 0
2. חייבת להיות 1
3. חייבת להיות 2
4. יכול להיות 0 או 1 או 2, אי אפשר לדעת

ב) נתון מספר בכתיב טרינארי, ומוסיפים לו 0 מצד ימין. למשל אם המספר היה 210 אז לאחר התוספת מתקבל 2100. כיצד תוספת זו של 0 מצד ימין משנה את המספר? הקיפו בעיגול.

1. מגדילה את המספר ב-1
2. מגדילה את המספר ב-3
3. מכפילה את המספר פי 2
4. מכפילה את המספר פי 3

שאלה 4 (15 נק')

סדרת פיבונאצ'י היא סדרת מספרים המוגדרת באופן הבא: המספר הראשון וגם המספר השני בסדרה שווים 1, ולאחר מכן כל מספר שווה לסכום שני קודמיו. למשל המספר השלישי שווה $1+1=2$, ואילו הרביעי שווה $1+2=3$. להלן רשימה של כמה מהאיברים הראשונים בסדרה:

1, 1, 2, 3, 5, 8, 13, ...

השלימו את הפונקציה הבאה, שמקבלת רשימה, ובודקת האם הרשימה מכילה התחלה של סדרת פיבונאצ'י. לשם פשטות, הניחו כי הרשימה באורך של לפחות 3. דוגמאות הרצה:

```
>>> fib([1,1,2,3,5,8])
```

True

```
>>> fib([1,1,4,5,8])
```

False

```
>>> fib([3,5,8])    #לא התחלה של הסדרה
```

False

הערה: באפשרותכן לא להשתמש בתבנית הבאה, ובמקום זאת לכתוב את הפונקציה מאפס בעמוד האחרון של המבחן. במקרה זה סמנו X בולט על הפונקציה בעמוד זה.

```
def fib(L):  
    # נבדוק האם שני האיברים הראשונים לא שווים ל-1  
    if _____:  
        return False  
    # כעת נבדוק את יתר האיברים  
    i = _____  
    while i < _____:  
        if _____:  
            return False  
        i = i+1  
    return _____
```

שאלה 5 (20 נקודות)

השאלה עוסקת בקוד לגילוי שגיאות, שפועל באופן הבא :
בהנתן הודעה באורך 3 ביטים, נוסף לה שני ביטים נוספים (השידור יהיה באורך 5 ביטים סה"כ). שני הביטים הנוספים יכילו את מספר האפסים שיש ב- 3 הביטים הראשונים, מספר זה מיוצג בבינארית. לדוגמא (הביטים שנוספו להודעה מסומנים בקו תחתון) :

010 → 01010

111 → 11100

הסבר : בהודעה 010 יש פעמיים 0, ולכן הוספנו 10 בסוף (2 בכתוב בינארי). בהודעה 111 אין כלל 0, ולכן הוספנו 00 בסוף.

א. רישמו את השידורים החוקיים שיתקבלו משתי ההודעות הבאות :

000 → _____

101 → _____

ב. כמה הודעות חוקיות יש בסה"כ? _____

ג. כמה הודעות בלתי חוקיות יש בסה"כ? _____

אורית רוצה לשלוח לאמיר את ההודעה 010 (כלומר, השידור החוקי עבור הודעה זו הוא 01010).

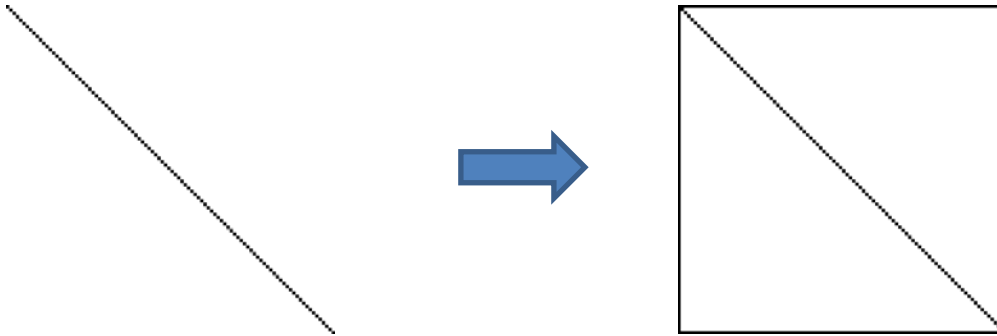
ד. אם נפלה בשידור שגיאה אחת, האם אמיר תמיד יוכל לגלות שהשידור בלתי חוקי? אם לדעתך כן – הסבירו בקצרה. אחרת תנו דוגמה לשגיאה בהודעה הנ"ל שלא יאפשרו להבחין כי השידור בלתי חוקי.

ה. נניח כי בשידור שהתקבל אצל אמיר נפלו 2 שגיאות בדיוק. הראו דוגמא לשידור כזה, שבו אמיר יוכל לזהות שהשידור בלתי חוקי. אין צורך להסביר.

ו. נניח כי בשידור שהתקבל אצל אמיר נפלו 2 שגיאות בדיוק. הראו דוגמא לשידור כזה, שבו אמיר לא יוכל לזהות שהשידור בלתי חוקי. אין צורך להסביר.

שאלה 6 (10 נק')

אמיר פתח חנות מסגרות. בכל פעם שמביאים לו תמונה הוא ממסגר אותה במסגרת שחורה. למשל, עבור התמונה השמאלית, שמכילה אלכסון שחור על רקע לבן, תתקבל התמונה הממוסגרת מצד ימין, שבה שורת הפיקסלים העליונה והתחתונה, וכן עמודת הפיקסלים השמאלית והימנית שחורות.



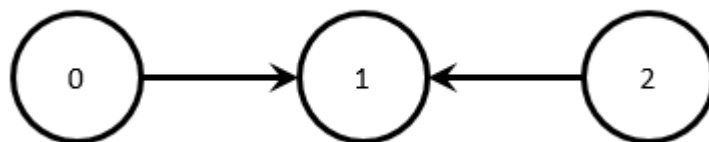
השלימו את הפונקציה `frame` הבאה, כדי שתיצור את אפקט המסגור הנ"ל בדיוק. הניחו כי הפונקציה `frame` מקבלת כקלט תמונה ריבועית (בגודל $n \times n$). יש להשלים שורה אחת בלבד.
הערה: נזכיר כי `im.size` מחזיר `w,h` (רוחב וגובה) של התמונה. כיוון שהתמונה ריבועית ושני הערכים זהים, נוכל להסתפק בראשון מביניהם.

```
def frame(im):  
  
    n = im.size[0] # ראו הערה למעלה  
  
    mat = im.load() # טעינת המטריצה שמייצגת את התמונה  
  
    for x in range(n):  
        for y in range(n):  
            if _____  
                mat[x,y] = 0  
  
    return im
```

שאלה 7 (10 נק')

תזכורת: מסלול אוילר בגרף הוא מסלול שעובר בכל קשת פעם אחת בדיוק. מעגל אוילר הוא מסלול אוילר שמתחיל ונגמר באותו צומת.

נבנה את הגרף המכונן הבא:



(א) האם קיים בגרף מסלול אוילר? כן / לא (הקיפו בעיגול)

(ב) השלימו את המטריצה המייצגת את הגרף:

$L = \begin{bmatrix} & & \\ & & \\ & & \end{bmatrix}$,

$\begin{bmatrix} & & \\ & & \\ & & \end{bmatrix}$,

$\begin{bmatrix} & & \\ & & \\ & & \end{bmatrix}$

(ג) כזכור, הפונקציה euler שראינו בכיתה (מצורפת בעמוד האחרון לנוחיותכם), מקבלת כקלט רשימה דו מימדית (מטריצה) המייצגת גרף לא מכונן וקשיר, ובודקת האם קיים בו מסלול מעגל אוילר.

נריץ את הפונקציה euler על הגרף המכונן L מסעיף ב.

אמיר טוען כי הפונקציה תחזיר תשובה נכונה, אף על פי שמעבירים לה גרף מכונן. כלומר:

- אם בסעיף א קבעתם שבגרף L קיים מסלול אוילר, הפונקציה תחזיר True.

- אם בסעיף א קבעתם שבגרף L אין מסלול אוילר, הפונקציה תחזיר False.

אורית טוענת כי הפונקציה תחזיר תשובה שגויה עבור הגרף הזה. מי מהם צודק?

הסבירו בקצרה מדוע הפונקציה מחזירה את התשובה שקבעתם.

שאלה 8 (15 נק')

בכיתה מימשנו צופן החלפה (substitution cipher) באמצעות הפונקציות encrypt המצפינה טקסט באמצעות מילון, ו-decrypt המפענחת טקסט מוצפן באמצעות אותו המילון (הפונקציות מצורפות בעמוד האחרון לנוחיותכם).

נבנה את המילון הבא :

```
>>> D = {'a':'l', 'l':'a', 't':'g', 'u':'u', 'r':'l', 'i':'n',  
'n':'a', 'g':'g'}
```

(א) מהו הפלט של הפקודה הבאה? רישמו במקום המיועד :

```
>>> e = encrypt("alan turing", D)  
>>> e
```

(ב) תזכורת: אם נצפין הודעה m בעזרת מילון D ונקבל את הטקסט המוצפן e, ולאחר מכן נפענח את e באמצעות אותו מילון D, נצפה לקבל את ההודעה המקורית m.

עבור המשתנה e שחושב בסעיף א, הסבירו מדוע הפקודה הבאה לא תחזיר את הערך הרצוי :

```
>>> decrypt(e, D)
```



```

def euler(G):
    ''' find if G has Euler path/cycle.
        Assume G is undirected, connected and has no self edges !
    '''
    n = len(G) #number of nodes in the graph
    odd = 0

    for i in range(n):
        deg = sum(G[i])#how many edges 'touch' G[i]
        if deg%2 == 1:
            odd += 1

    if odd == 0:
        print("Euler cycle")
    elif odd == 2:
        print("Euler path")
    else:
        print("No Euler path")

```

```

import random

#####
## Encryption
#####
def encrypt(text, enc_dict):
    ''' converts text using enc_dict as substitution key '''
    cipher = ""
    for ch in text:
        if ch in enc_dict:
            cipher += enc_dict[ch]
        else:
            cipher += ch #characters not in enc_dict untouched
    return cipher

#####
## Decryption, when we have the substitution cipher
#####
def reverse_dict(enc_dict):
    dec_dict = {}
    for source in enc_dict:
        target = enc_dict[source]
        dec_dict[target] = source
    return dec_dict

def decrypt(cipher, enc_dict):
    dec_dict = reverse_dict(enc_dict)
    return encrypt(cipher, dec_dict)

```

!٩١٥