מס' ת"ז_____ מס' מחברת_____

מבחן בקורס מבוא מורחב למדעי המחשב, CS1001.py
ביה"ס למדעי המחשב, אונ' תל אביב

סמסטר ב' 2014-15, מועד ב, 20/8/2015

מרצים: אמיר רובינשטיין, פרופ' בני שור

מתרגלות: יעל ברן, מיכל קליינבורט

משך הבחינה: 3 שעות.

חומר עזר מותר: 2 דפי עזר (דו צדדיים) בגודל A4 כ"א.

- במבחן 10 עמודים מודפסים – בידקו שכולם בידיכם. בנוסף, בסוף הבחינה ישנו דף נוסף, ריק, לשימוש ב"מקרה חירום" בלבד.
- יש לכתוב את כל התשובות בטופס הבחינה. המחברת תשמש כטיוטה בלבד ולא תיבדק.
- יש לענות על כל חמש השאלות.
- בכל השאלות, אלא אם נכתב במפורש אחרת:
  - אם עליכם לכתוב פונקציה, אין צורך לבדוק את תקינות הקלט שלה.
  - מותר להסתמך על סעיפים קודמים, גם אם לא עניתם עליהם.
  - ניתן לצטט טענות שנטענו בהרצאה או בשיעורי התרגול. במקרה זה יש לכתוב "בהרצאה/תרגול ראינו כי...".
- אנו ממליצים לא "להיתקע" על אף שאלה, אלא להמשיך לשאלות אחרות ולחזור לשאלה אח"כ.
- בכל סעיף ניתן לכתוב "איני יודע/ת" ולא לכתוב שום טקסט נוסף. במקרה זה יינתן 20% מציון הסעיף.
- יש לכתוב את כל התשובות במקום המוקצב ובכתב קריא. תשובות ובהן חריגות משמעותיות מהמקום המוקצב, או תשובות הכתובות בכתב קטן מדי, לא ייקראו ולא יקבלו ניקוד, או שיקבלו ניקוד חלקי בלבד. תשובות שדורשות מאמצים רבים להבנתן גם כן עלולות לגרור הורדת ציון.

בהצלחה !

טבלת ציונים: (לשימוש הבודקים)

שאלה	ערך	ניקוד
1	20	
2	30	
3	10	
4	15	
5	25	
סה"כ	100	

שאלה 1 (20 נק')

להלן מימוש חלקי למחלקה Binary המייצגת מספר טבעי (גדול או שווה 0) בכתיב בינארי. המספר ייוצג כמחרוזת.

לשם פשטות, לאורך כל השאלה נניח כי אין אפסים מובילים במספרים הבינאריים, למעט המספר "0". אין צורך לבדוק זאת במתודות שתכתבו או להתייחס למקרה כזה.

```
class Binary():
    def __init__(self, s):
        """ represent a binary number as a string """
        assert type(s)==str
        assert s.count("0") + s.count("1") == len(s)
        self.s = s

    def __repr__(self):
        return "0b" + self.s

    def length(self):
        return len(self.s)

    def __eq__(self, other):
        return self.s == other.s

    def __lt__(self, other):
        """ operator < """
        # this method is implemented for you; code omitted

    def __add__(self, other):
        """ operator + """
        # this method is implemented for you; code omitted
```

דוגמאות הרצה:

```
>>> Binary("0")
0b0
>>> Binary("1101") #13 in binary
0b1101
>>> Binary("1101") == Binary("1110") #calls __eq__
False
>>> Binary("1101") < Binary("1110") #calls __lt__
True
>>> Binary("100") + Binary("101") #calls __add__
0b1001
```

א. מיכל רוצה לדעת מה שארית החלוקה של מספר בינארי ב 3. היא התחילה לממש את המתודה div3 הבאה, שאמורה להחזיר את שארית החלוקה ב- 3 של מספר מהמחלקה Binary. השלימו את שתי השורות החסרות בפונקציה div3. עליכם להשתמש אך ורק במתודות שמופיעות בעמוד הקודם:

```
def div3(self):
    """ Returns remainder of self mod 3 """
    cnt = Binary("0")
    while cnt < self:
        cnt = cnt + Binary("11")
    div = 0
    if _____:
        div = 2
    if _____:
        div = 1
    return div
```

ב. בני מעוניין לחשב שארית מודולו 3 של מספרים בינאריים בני אלפי ביטים, וחושש שהמתודה מהסעיף הקודם לא תסתיים עד תחילת סמסטר הסתיו.

הוא שם לב לתכונות הבאות:

- כאשר מוסיפים למספר בינארי 0 מימין (לדוגמה, משנים "1" ל "10"), אם שארית החלוקה שלו ב- 3 לפני ההוספה הייתה 1, אז שארית החלוקה שלו ב 3 אחרי ההוספה תהיה 2.
- כאשר מוסיפים למספר בינארי 1 מימין (לדוגמה, משנים "1" ל "11"), אם שארית החלוקה שלו ב- 3 לפני ההוספה הייתה 1, אז שארית החלוקה שלו ב 3 אחרי ההוספה תהיה 0.

בני שם לב כי באופן דומה, ניתן לדעת מה תהיה שארית החלוקה ב 3 של מספר בינארי לאחר הוספת 0 או 1 מצד ימין, גם כאשר שארית החלוקה שלו לפני ההוספה הייתה 0 או 2.

השלימו בטבלה הבאה את שאריות החלוקה לאחר הוספת 0 או 1 מימין למספר בינארי:

השארית של חלוקת המספר הנוכחי ב- 3	שארית החלוקה לאחר הוספת 0 מימין	שארית החלוקה לאחר הוספת 1 מימין
0		
1	2	0
2		

ג. השלימו בעמוד הבא את המימוש של div3_new (שמחזירה אותם ערכים כמו div3), תוך שימוש בחוקיות שבטבלה שלעיל.

```
def div3_new(self):
    """ Returns remainder of self mod 3 """
    next_0 = {0:___, 1:2, 2:___}
    next_1 = {0:___, 1:0, 2:___}
    remainder = 0
    for b in self.s:
        if b=="0":
            remainder = _____________________________
        else:
            remainder = _____________________________
    return remainder
```

ד. ציינו מהי סיבוכיות הזמן של המתודה div3_new החדשה הנ"ל, כתלות במספר הביטים n של self. תנו תשובה הדוקה ככל שתוכלו במונחי O(…).

O(_____)

ה. יעל טוענת כי הפתרון מסעיף א' הינו בעל סיבוכיות זמן ריצה טובה יותר מאשר זה של סעיף ג'. בני טוען כי יעל טועה. מי משניהם צודק? הסבירו בקצרה.

_____

_____

_____

_____

שאלה 2 (30 נק')

בשאלה זו נתאר מספר פונקציות שעוסקות במציאת אורכה של תת-המחרוזת המשותפת הארוכה ביותר (Longest Common Substring, מקוצר כ LCS) של שתי מחרוזות.

המחרוזת s היא LCS של המחרוזות s1 ו- s2 אם ורק אם מתקיימות התכונות הבאות:

1. המחרוזת s היא תת מחרוזת של s1.
2. המחרוזת s היא תת מחרוזת של s2.
3. אין מחרוזת שארוכה יותר מ s ומקיימת את תנאים 1 ו 2.

לדוגמה, ה LCS של "ababc" ושל "dbabca" היא "babc", וה LCS של "xxx" ושל "abcd" היא "" (המחרוזת הריקה). שימו לב שייתכן כי קיימת יותר מ LCS אחת לזוג מחרוזות נתון.

נסמן ב n1, n2 את אורכי המחרוזות s1, s2 בהתאמה.

א. 1. השלימו את הפונקציה הבאה, אשר מקבלת שתי מחרוזות ומחזירה את אורך ה LCS שלהן:

```
def lcs_length_1(s1,s2):

    if len(s1)==0 or len(s2)==0:
        return 0
    m = [[0]*len(s2) for i in range(len(s1))]
    for i in range(0,len(s1)):
        for j in range(0,len(s2)):

            if _____:
                m[i][j] = \
                1 + (m[i-1][j-1] if 0<i and 0<j else 0)

    return _____
```

2. תארו במלים מהו הערך אותו מכיל התא m[i][j] בתום הריצה. יש להשלים משפט בודד (אין צורך ביותר מ-20 מילים):

התא m[i][j] מכיל _____

_____

3. ציינו מהי סיבוכיות זמן הריצה של הפתרון כתלות באורכי המחרוזות n1 ו- n2. תנו תשובה הדוקה ככל שתוכלו במונחי $O(\ldots)$. הניחו כאן כי פעולות אריתמטיות מתבצעות בסיבוכיות זמן קבועה.

O(_____)

המשך השאלה מתייחס למימוש פתרון לבעיית ה- LCS בגישה שונה.

ב. 1. השלימו את הקוד הבא, אשר בודק האם קיימת תת-מחרוזת משותפת באורך k לשתי מחרוזות נתונות:

```
def has_common(s1,s2,k):

   if len(s1)<k or len(s2)<k:
       return False
   something = set()
   for i in _____:

       something.add(_____)

   for i in_____:

       if _____:

           return True

   return False
```

2. ציינו מהי סיבוכיות זמן הריצה הממוצעת של של has_common כתלות ב n1, n2 ו- k, בהנחה ש- min(n1,n2) > k, ושהערך שהפונקציה מחזירה הוא False. תנו תשובה הדוקה ככל שתוכלו במונחי O(…).

O(_____)

3. הציעו דרך לייעל את has_common, כך שסיבוכיות זמן הריצה שלה תהיה O(n1 + n2) תחת אותן הנחות. יש להסביר במילים בצורה ברורה. אין צורך לכתוב קוד.

ג. השלימו את מימושה של הפונקציה lcs_length_2. כמו הפונקציה lcs_length_1, היא מחזירה את אורכה של ה LCS בין שתי מחרוזות נתונות, אך היא משתמשת ב- has_common:

```
def lcs_length_2(s1,s2):
    if not has_common(s1,s2,1):
        return 0
    k = 1
    while has_common(s1,s2,k):
        k = k*2

    low = _____
    high = k-1
    while low < high-1:
        mid = (low+high)//2

        if _____:


            _____
        else:
            high = mid-1
    return high if has_common(s1,s2,high) else low
```

ד. נסמן ב- m את אורך תת המחרוזת המשותפת הארוכה ביותר של שתי מחרוזות נתונות s1 ו- s2.
כלומר m = | LCS(s1,s2) |. ציינו מהו מספר האיטרציות הכולל, בשתי הלולאות שמרכיבות את הפונקציה lcs_length_2, כתלות ב- m. אין צורך לתת ביטוי מדויק, אלא תשובה במונחי O(…), הדוקה ככל שתוכלו.

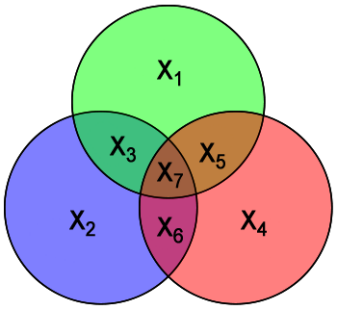
O(_____)

שאלה 3 (10 נק')

תזכורת לקוד Hamming:

אנו רוצים לשדר 4 ביטים של מידע x3, x5, x6, x7. ה- encoder מוסיף 3 ביטים x1, x2, x4 באופן הבא:

```
def hamming_encode(x3,x5,x6,x7):
    """ Hamming encoding of the 4 bits input """
    x1= (x3+x5+x7) % 2
    x2= (x3+x6+x7) % 2
    x4= (x5+x6+x7) % 2
    return (x1,x2,x3,x4,x5,x6,x7)
```



בכל אחד מהסעיפים הבאים, הקיפו בעיגול את את התשובה הנכונה, והסבירו.

א. סמנו מה מהבאים נכון לגבי המרחק בין זוגות מילות הקוד (החוקיות) בקוד האמינג:
1) המרחק הוא תמיד 3
2) חלק מהמרחקים הם 2, חלק הם 3, וחלק גדולים מ- 3.
3) חלק מהמרחקים הם 3, חלק הם 4, אבל אין מרחקים שגדולים מ- 4.
4) חלק מהמרחקים הם 3, חלק הם 4, וחלק הם 7.

ב. נניח כי נשלחה ההודעה 0000000, ולאחר מעבר בערוץ תקשורת רועש התקבלה ההודעה 1100000, כלומר נפלו שתי שגיאות, בביטים x1 ו- x2. במקרה זה:
1) ה- decoder יפרש זאת כשגיאה בודדת בביט 3
2) ה- decoder יפרש זאת כשגיאה בודדת בביט 4
3) ה- decoder יוכל לזהות שנפלו שתי שגיאות, אך לא יידע היכן (ולכן לא יידע לתקן)
4) ה- decoder יוכל לזהות שנפלו שתי שגיאות, ויידע היכן (כלומר יוכל לתקן).

שאלה 4 (15 נק')

יהי p מספר ראשוני כלשהו, באורך n ביטים.

עבור שני שלמים $1 \leq a, b \leq p-1$ נאמר כי b הוא <u>הופכי כפלי מודולו p</u> של a אם $a \cdot b = 1 \; mod \; p$.

לפני זמן רב (מאוד) כשבני למד לתואר ראשון במתמטיקה, נודע לו כי לכל $1 \leq a \leq p-1$ קיים הופכי כפלי מודולו p. נלהב מכך שנזכר בעובדה זו, בני הציע את האלגוריתם הבא לחישוב ההופכי הכפלי מודולו p של a נתון $(1 \leq a \leq p-1)$:

- לכל $b = 1,2,3,\ldots, p\text{-}1$:
  - נחשב $(a \cdot b) \; mod \; p$
  - אם התוצאה היא 1, נחזיר את b ונסיים

א. עבור $p=7$, מהו ההופכי הכפלי מודולו p של $a=2$? _____

ב. מהו מספר **המכפלות** מודולו p הנדרש ע"י האלגוריתם של בני במקרה הטוב ביותר? ומהו במקרה הגרוע ביותר? בטאו את התשובות כפונקציה של n, תוך שימוש בסימון O.

מספר המכפלות במקרה הטוב ביותר O(_____)

מספר המכפלות במקרה הגרוע O(_____)

ג. אמיר טוען כי האלגוריתם שבני הציע מוכיח כי בני לא הפנים את החומר בקורס "מבוא מורחב למדעי המחשב", ובפרט לא הבין את המשפט הקטן של פרמה:

$$a^{p-1} = 1 \; mod \; p \quad \text{לכל} \quad 1 \leq a \leq p-1.$$

לטענת אמיר, זהות זו מאפשרת למצוא הופכי כפלי מודולו p באופן הרבה יותר יעיל.
כיתבו ביטוי מתמטי מפורש עבור ההופכי הכפלי של a מודולו p , בהתבסס על המשפט הקטן של פרמה.

ד. ציינו שמו של אלגוריתם שנלמד בקורס, המאפשר חישוב יעיל של ההופכי הנ"ל, בהתבסס על ההביטוי שרשמתם.

ה. מהו מספר המכפלות מודולו p שהאלגוריתם שציינתם בסעיף הקודם יבצע, כתלות ב n, במקרה הגרוע? כיתבו ביטוי במונחי $O(\ldots)$ ונמקו בקצרה.

O(_____)

שאלה 5 (25 נק')

נגדיר סדרת מספרים ע"י הנוסחה הרקורסיבית הבאה:

$$C(n) = \sum_{i=0}^{n-1} C(i)C(n-i-1)$$

ועם תנאי הבסיס $C(0) = 1$.

(הערה: איברי הסדרה מוכרים כמספרי קטלאן, אך עובדה זו איננה חשובה להמשך השאלה).

א. ממשו פונקציה catalan1(n), המקבלת את n ומחזירה את $C(n)$. הפונקציה תפעל באופן איטרטיבי (כלומר באמצעות לולאות, וללא שימוש בקריאות רקורסיביות).

```
def catalan1(n):
    cat = [0]*(n+1)
```

ב. ציינו מהי סיבוכיות הזמן של catalan1 שכתבתם, כפונקציה של n, במונחים של $O(\ldots)$.
תנו תשובה הדוקה ככל שתוכלו.

O(_____)

ג. ממשו בעמוד הבא פונקציה המקבלת את n ומחזירה את $C(n)$. הפונקציה תפעל באופן רקורסיבי, ותהווה מימוש ישיר של הנוסחה לעיל. בנוסף, הפונקציה תפעיל מנגנון של ממואיזציה לצורך ייעול. כלומר היא תימנע מלחשב מחדש ערך של הפונקציה שכבר חושב.
הפונקציה catalan2 היא פונקצית מעטפת ואין לשנותה; השלימו את הקוד של catalan_rec בלבד.

בנוסף להשלמת הקוד בעמוד הבא:

ציינו מהו עומק הרקורסיה של catalan2, כפונקציה של n, במונחים של $O(\ldots)$:

O(_____)

ציינו מהי סיבוכיות הזמן של catalan2, כפונקציה של n, במונחים של $O(\ldots)$:

O(_____)

בשני המקרים תנו תשובה הדוקה ככל שתוכלו.
הניחו כי פעולות אריתמטיות (כפל וחיבור) רצות בזמן קבוע – $O(1)$.

```
def catalan2(n):
    d = dict()
    return catalan_rec(n, d)

def catalan_rec(n, d):
    if n == 0:
        result = 1
    else:
```

ד. ידועה הנוסחה הסגורה הבאה לחישוב $C(n)$:

$$C(n) = \prod_{i=2}^{n} \frac{n+i}{i}$$

להלן מימוש של catalan3 ע"פ הנוסחה הנ"ל:

```
def catalan3(n):
    res = 1
    for i in range(2,n+1):
        res *= ((n+i)/i)
    return round(res)
```

התבוננו בהרצות הבאות:

```
>>> catalan1(35) #Assuming your code from (a) is correct
3116285494907301262 #This is the correct result
>>> catalan3(35)
3116285494907300864
```

מהו המקור להבדל בין שתי התוצאות?

דף נוסף למקרה הצורך