תרגיל 6 – רקורסיה מתקדמת וממוריזציה

תאריך פרסום: 28/11/2018

תאריך הגשה: 4/12 בשעה 23:59

מתרגל אחראי: ניר פרידמן

הנחיות כלליות:

- קראו את כל ההוראות לגבי הגשת תרגילי הבית באתר הקורס.
- קראו את כל העבודה לפני שתתחילו לפתור אותה.
- כתבו תיעוד (הערות) שמסביר את הקוד שלכם. אין לכתוב הערות בעברית.
- שאלות בנוגע לעבודה יישאלו ב-"פורום שאלות לתרגיל בית 6" במודל או בשעות הקבלה של המתרגל האחראי בלבד.
- כתבו את הקוד בקבצים בשם 'hw6_ques1.py' ו-'hw6_ques2.py' בלבד.
- השימוש בחבילות חיצוניות אסור בהחלט ויגרור ציון 0.
- על הפלטים להיות בדיוק כפי שמוגדר בתרגיל (ללא רווחים מיותרים בפלטי טקסט print).
- הקפידו להגדיר את חתימות הפונקציות באופן מדויק כפי שהוגדרו בתרגיל. מערכת הבדיקה קוראת לפונקציות בצורה אוטומטית וחתימה שגויה תגרור ציון 0.

שאלה 1:

בשאלה זו נרחיב את פתרון בעיית 'Inheritance Partition' שראינו בתרגול. נזכיר כי בבעיה זו, אנו מעוניינים לחלק רשימה של n נכסים (כאשר לכל נכס שווי מסויים), בין שני יורשים בשם יוסי וליאורה. לדוגמא רשימת הנכסים [10 ,5 , 20 , 10] מציינת כי יש לחלק 4 נכסים בשווי 10, 20, 5 ו-10. בשונה מהדוגמא בתרגול, כללי חלוקת הנכסים מוגדרים באופן הבא:

- שווי הנכסים יחולק ע"י קבלת משתנה מסוג tuple, שיגדיר את סך ערך הנכסים שיחולקו, כאשר המספר הראשון הוא החלק של ליאורה והשני של יוסי. לדוגמא, עבור (30,15) יש לחלק את הנכסים כך שליאורה תקבל נכסים בסך 30 דולר ויוסי בסך 15 דולר.
- עלינו לחלק את **כל הנכסים** בין שני היורשים, כל נכס חייב להיות משוייך לליאורה או יוסי.
- כל אחד מהיורשים רשאי לקבל עד (כולל) n-2 מהנכסים, כאשר n הוא מספר הנכסים הכולל ברשימת הקלט. לדוגמא, במידה ויש 10 נכסים, המספר המקס' של נכסים שאחד מהיורשים רשאי לקבל הוא 8.
- מאחר וליאורה הייתה הבת המועדפת, יש לבחור את הנכסים לליאורה, ורק לאחר-מכן ליוסי. *רמז: חישבו על סדר הקריאות.*
- שימו לב שעליכם לדווח את הנכסים שכל יורש יקבל. נכס ייוצג באמצעות המיקום שלו ברשימת הקלט.
- הניחו שהקלט תקין, כלומר רשימה של מספרים חיוביים, באורך גדול או שווה ל-2.
- **השתמשו ברקורסיה עם ממוריזציה בכדי לפתור את התרגיל, אין להשתמש בלולאות.**

עליכם לממש את הפונקציה partition (lst,values_lst), אשר מקבלת רשימת נכסים (lst), ואת ה-tuple (values_lst) שמגדיר את סך הנכסים שיחולקו ליורשים ליאורה ויוסי (בהתאמה). הפונקציה תחזיר רשימה באורך רשימת הקלט, שתכיל באינדקס i את שם היורש של נכס i ("Yossi" או "Liora"). במידה ואין חלוקה שעונה על הקריטריונים, יש להחזיר רשימה ריקה (באורך 0).

❖ **דוגמאות הרצה:**

```
inher_lst_1 = [50 , 20 , 10 , 15 , 5]
values_1 = (60 , 40)
print( partition(inher_lst_1 , values_1) )

inher_lst_2 = [5 , 40 , 20 , 20 , 20]
values_2 = (60 , 45)
print( partition(inher_lst_2 , values_2) )

inher_lst_3 = [30 , 20 , 20 , 15 , 5]
values_3 = (60 , 40)
print( partition(inher_lst_3 , values_3) )

inher_lst_4 = [60 , 20 , 15 , 5]
values_4 = (60 , 40)
print( partition(inher_lst_4 , inher_lst_4) )
```

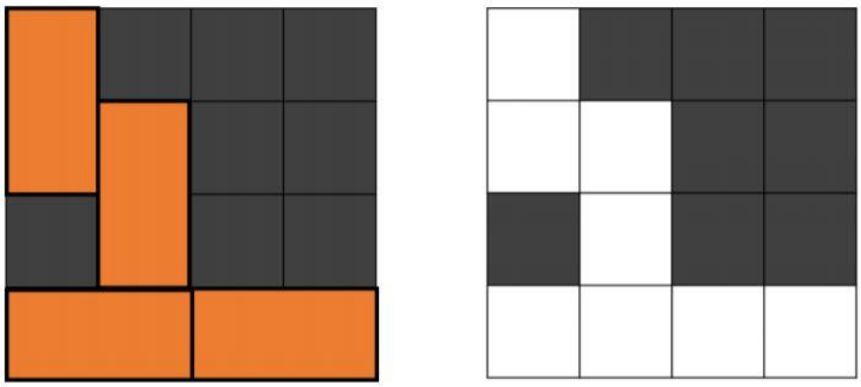
```
['Liora', 'Yossi', 'Liora', 'Yossi', 'Yossi']
['Yossi', 'Liora', 'Liora', 'Yossi', 'Yossi']
[]
[]
```

שאלה 2:

בשאלה זו נממש פונקציה רקורסיבית אשר מוצאת פתרון למשחק. לוח המשחק הוא טבלה דו-מימדית, ובו חלק מהתאים ריקים (לבן באיור מימין) וחלק חסומים אשר לא ניתן לשנות אותם (מסומנים בשחור באיור מימין).

מטרת המשחק הינה לרצף את **כל** המשבצות הריקות בלוח בקוביות בגודל 2x1 (מסומנות בכתום באיור משמאל). כלומר, הצלחה משמעותה ריצוף מלא של הלוח בסוף תהליך השיבוץ.

לא קיים פתרון חוקי לכל לוח. על הפונקציה להחזיר ריצוף חוקי של הלוח במידה וקיים, אחרת תחזיר כי אין פתרון (פרטים בהמשך). לדוגמה, עבור הלוח התקין מימין מוצע פתרון חוקי משמאל:



כתבו פונקציה (domino_problemboard המקבלת כקלט לוח משחק - רשימה דו-מימדית בגודל NxM (list באורך N, כאשר כל איבר הוא list בגודל M), N מייצג את מספר השורות ו-M את מספר העמודות.
על התוכנית להחזיר פתרון חוקי (במידה וקיים) כאשר כל קוביה תיוצג באמצעות שני תווי "B", תא ריק יסומן בתו "*" ותא חסום יסומן בתו "#".. במידה ולא קיים פיתרון, על התוכנית להחזיר את המחרוזת "Broken board".

לנוחותיכם, אנו מספקים את הפונקציה (print_board(board, אשר מקבלת לוח ומדפיסה אותו.
ניתן להניח כי:

- הקלט תקין מבחינת טיפוסים וגדלים של כל הרשימות והערכים שבהן.
- כל הערכים בלוח הקלט הן "*" או "#".
- $1 \leq M, N \leq 10$

<u>הנחיה</u>
מותר ומומלץ להשתמש בפונקציית עזר רקורסיבית. *<u>רמז- עבור כל תא בלוח, חישבו מה האפשרויות לשיבוץ קוביה שתכסה את התא.</u>*

<u>אין להשתמש בלולאות, שימוש בלולאה יגרור ציון 0 לתרגיל.</u>

<u>דוגמאות הרצה:</u>

```
board_1 = [['*' , '#'],
           ['*' , '#']
          ]

print_board( domino_problem(board_1) )
```

```
B  #

B  #
```

```
board_2 = [['*' , '#' , '#' , '#' , '*' , '#'],
           ['*' , '*' , '#' , '#' , '*' , '*'],
           ['#' , '*' , '#' , '#' , '#' , '*'],
           ['*' , '*' , '*' , '*' , '#' , '#']
          ]
print_board( domino_problem(board_2) )
```

```
B  #  #  #  B  #

B  B  #  #  B  B

#  B  #  #  #  B

B  B  B  B  #  #
```

```
board_3 = [['*' , '#' , '#' , '#'],
           ['#' , '*' , '#' , '#'],
           ['#' , '*' , '#' , '#'],
           ['*' , '*' , '*' , '*']
          ]
print( domino_problem(board_3) )
```

```
Broken board
```

הנחיות הגשה:

- יש להגיש את העבודה למערכת ההגשה כפי שמתואר בהנחיות ההגשה במודל.
- כתבו טסטים לקוד שלכם, מעבר כל הבדיקות במערכת ההגשה **אינו מבטיח ציון 100!**
- יש להגיש קובץ מכווץ אחד בלבד בשם 'hw6.zip' (לא rar. או שום סיומת אחרת), כאשר בתוכו נמצאת תיקייה בשם hw6 ובתוכה הקבצים 'hw6_ques1.py' ו-'hw6_ques2.py'. כל מבנה, שם אחר או סיומת אחרת ייגררו ציון 0.

בהצלחה! ☺