

תרגיל 4 - רקורסיה

תאריך פרסום:	12/12
תאריך הגשה:	יפורסם בהמשך
מתרגל אחראית:	עומרי אביטל
משקל תרגיל:	5 נקודות
מטרות העבודה:	עבודה עם פונקציות מעטפת, הרכבת אלגוריתמים רקורסיביים.
דגשים לתרגיל 4:	

- כל שימוש בפקודות השמורות `for` ו-`while` (בכל צורה שהיא – למשל `list` `sorted` או `comprehension`) יגרור את פסילת הפתרון. פרט לשאלות בהן מצוין אחרת.
- ניתן לממש פונקציות מעטפת ופונקציות עזר.
- שם פונקציית מעטפת צריך להיות תואם לשם הפונקציה שאתם מתבקשים לממש בסעיף. אין מגבלה על שם הפונקציה הרקורסיבית שנקראת מפונקציית המעטפת.
- אין להשתמש בפרמטרי ברירת מחדל בהגדרה של פונקציה.
- ניתן להניח שהקלט תקין.
- שימו לב, חלק מהשאלות ידרשו מכם לשלוח עותקים של רשימות.

שאלה 1

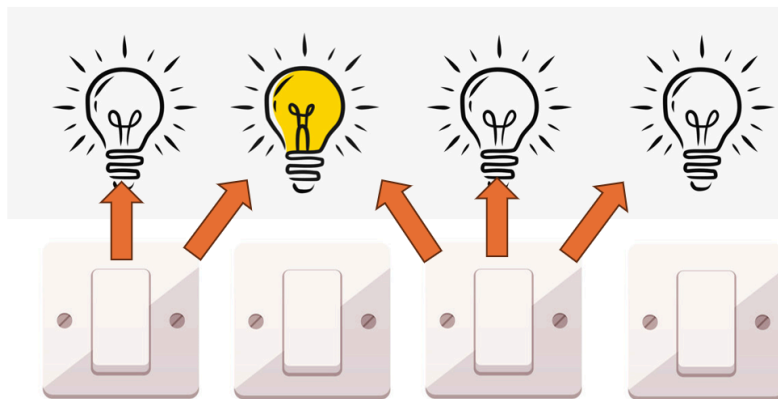
בעיית הנורות:

נציג שורה של נורות באמצעות רשימה המכילה ערכים בוליאניים כאשר:

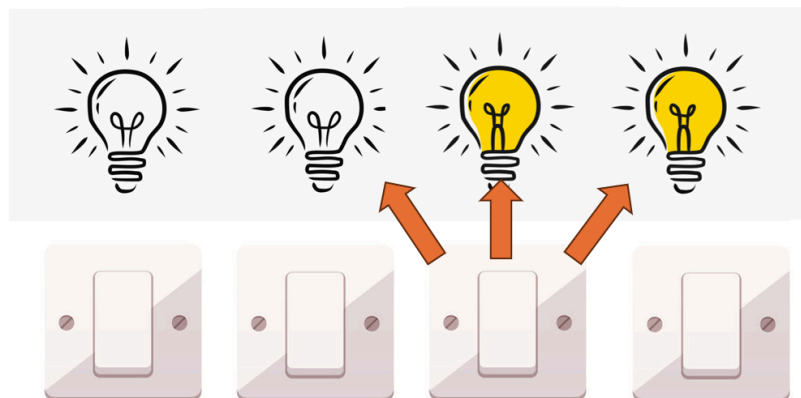
- True – מייצג נורה דולקת
- False – מייצג נורה כבויה

מול כל נורה נמצא מתג.

המתג יהפוך את מצבה של הנורה מולה הוא ניצב ושל הנורות הסמוכות לה כמתואר באיור.



לדוגמה לחיצה על המתג הימני עליו החצים נמצאים באיור הקודם, תגרום לשינוי הבא:



סעיף א':

ממשו את הפונקציה `lightbulb_solver(lightbulb_array, target_array)`

קלט:

הפונקציה מקבלת שתי רשימות באותו האורך:

- המשתנה `lightbulb_array` מייצג את המצב הנתון של הנורות כמערך.
- המשתנה `target_array` מייצג את מצב המטרה אליו נרצה להביא את הנורות.

פלט:

על הפונקציה להחזיר `True` במידה וניתן באמצעות לחיצה על מתגים כמתואר בהגדרות השאלה להפוך את `lightbulb_array` ל- `target_array` ו- `False` אחרת.

למשל עבור הצמידים:

```
lightbulb_array = [False, False, False, False, False]
```

```
target_array = [False, False, False, False, True]
```

```
lightbulb_array = [True, True]
```

```
target_array = [False, True]
```

יוחזר הערך `False`

ועבור הצמידים:

```
lightbulb_array = [True, False, False, False, False, False]
```

```
target_array = [False, True, False, True, True, True]
```

```
lightbulb_array = [True, False, False]
```

```
target_array = [True, False, True]
```

יוחזר הערך `True`

סעיף ב':

ממשו את הפונקציה (lightbulb_array, target_array) lightbulb_solver_with_steps

קלט:

הפונקציה מקבלת שתי רשימות באותו האורך:

- המשתנה lightbulb_array מייצג את המצב הנתון של הנורות כמערך.
- המשתנה target_array מייצג את מצב המטרה אליו נרצה להביא הנורות את.

פלט:

במידה וניתן להגיע מ lightbulb_array ל- target_array, על הפונקציה להחזיר, **בסדר עולה**, את האינדקסים של המתגים בהם נעשה שימוש.

אחרת יש להחזיר את הרשימה [-1].

למשל עבור:

```
lightbulb_array = [True, False, True]
```

```
target_array = [True, True, True]
```

יוחזר [0,1,2]

עבור:

```
lightbulb_array = [True,False, False, False, False, False]
```

```
target_array = [False, True, False, True, True, True]
```

יוחזר [0,4]

עבור:

```
lightbulb_array = [False, False, False, False, False]
```

```
target_array = [False, False, False, False, True]
```

יוחזר [-1]

שאלה 2

שאלת הפנקייקים המורחבת:

לפתרון השאלה ניתן להשתמש בלולאה אחת בלבד.

ניתן להשתמש בפונקציות עזר.

בתרגיל 2 חילקנו פנקייקים בין סועדים והגבלנו את כמות הפנקייקים לסועד ל-2 פנקייקים בלבד. כעת, נבצע את החלוקה כשהגבלה נתונה רק על גודל הפנקייקים.

חוקי חלוקת הפנקייקים החדשים כדלהלן:

- סכומי הגדלים של כל הסועדים יהיו שווים.

ממשו את הפונקציה: `divide_pancakes_extended(pancakes, total_size)`

קלט:

- `pancakes` – רשימה של מספרים טבעיים גדולים מ-0 המייצגת את גדלי הפנקייקים.
- `total_size` – סכום הגדלים הנדרש עבור כל סועד.

פלט:

- על הפונקציה להחזיר את כמות הסועדים המקסימלית להם אפשר לחלק פנקייקים בהתאם לחוקים.

למשל עבור:

`pancakes = [1, 1, 4, 2, 3, 3, 2, 5]`

`total_size = 6`

יוחזר הערך 3.

עבור הקומבינציות: `[1,5]`, `[1,2,3]`, `[2,4]` לדוגמה.
במקרה הזה קיימות קומבינציות נוספות שיובילו לתוצאה 3.

שאלה 3

בונים מילים:

בשאלה זו תממשו משחק בו יש להרכיב מילה המזכה בניקוד מקסימאלי תוך שימוש בקלפי אותיות.

נניצג את הקלפים באמצעות רשימה של מחרוזות המורכבות מאותיות קטנות (lower case).

- ניתן להשתמש בכל קלף פעם אחת בלבד.
- המלים שמזכות בניקוד נתונות כמילון בו המפתח הוא מילה והערך הוא הניקוד שלה.

ממשו את הפונקציה `create_word(cards, words)`

קלט:

- `cards` – רשימה המכילה תווים (מחרוזות באורך 1 של אותיות קטנות).
- `words` – מילון המכיל מילים חוקיות המזכות בנקודות.

פלט:

- יש להחזיר את המילה המזכה בכמות הנקודות הגבוהה ביותר הניתנת להרכב באמצעות הקלפים שיש לנו. אם לא ניתן להרכיב אף מילה שמזכה בנקודת תוחזר מחרוזת ריקה.

למשל עבור הארגומנטים:

```
words = {'hi': 1, 'hello': 40, 'world': 10, 'this': 4, 'a': 6, 'test': 7}
```

```
cards = ['h', 'e', 'l', 'o', 't', 'i', 's', 'w', 'r', 'd']
```

תוחזר המילה `world`, עבורה הניקוד המתקבל הוא 10.

שימו לב שבעוד והמילה `hello` מזכה ביותר נקודות היא לא ניתנת להרכבה מקלפים שכן האות `l` מופיעה פעם אחת בלבד. (L קטנה)