

## מבני נתונים סמסטר א' תש"ף – תרגיל בית להגשה מס' 1

**תאריך הגשה אחרון: 28/11/19 ב- 23:55**

**מתרגל אחראי: אבישי קריף.**

### הנחיות:

- קראו היטב את השאלות.
- ניתן להגיש את העבודה בזוגות. לא ניתן להגיש את העבודה בקבוצה מעל שני אנשים.
- יש להגיש את העבודה בקובץ zip בתוכו יהיו הקבצים:  
`ex1.pdf`, `WordLadder.cpp`, `WordLadder.h`
- **שם הקובץ שיוגש למערכת ההגשה יהיה מורכב מת"ז של המגיש/ים.** לדוגמה:  
עבור הגשה ביחיד - `11111111.zip`  
עבור הגשה בזוג - `11111111_22222222.zip`
- במקרה של הגשה בזוגות, רק אחד מבני הזוג יגיש את העבודה במודל.
- גם בתחילת הקובץ בעבודה יש לרשום את שמות ומספרי ת.ז. של המגישים.
- איחור במועד ההגשה יגרור הורדה של ציון, 5 נק' לכל יום איחור או חלק ממנו.  
בכל מקרה לא יהיה ניתן להגיש מעבר ל-3 ימי איחור ממועד ההגשה המקורי.  
במקרים חריגים בלבד יש לפנות למרצה כדי לקבל אישור על הגשה באיחור.
- יש לתת הוכחות פורמליות עבור חסמים אסימפטומטיים בדומה לדוגמאות שהועברו בכיתה.
- שאלות לגבי העבודה יש לשאול בפורום באתר הקורס ("מודלי") או בשעות קבלה של המתרגל האחראי **בלבד**. אין לשלוח שאלות במייל המתרגלים או המרצה.
- **להזכירכם: יש לשמור על הגינות אקדמית!**

## חלק תיאורטי

### שאלה 1:

סדרו את הפונקציות הבאות לפי סדר אסימפטומטי  $O(\dots)$  מן ה"קטנה" ל"גדולה".  
אם שתי פונקציות בסידור מקיימות  $f_i = \Theta(f_k)$  אז ציינו והוכיחו זאת. תנו הוכחה פורמלית  
לתשובתיכם לגבי כל זוג עוקב של פונקציות בסידור וכל שתי פונקציות עבורן  $f_i = \Theta(f_k)$ .

$$f_1(n) = 2017, \quad f_2(n) = 2^{\log \sqrt{2} n}, \quad f_3(n) = 2^{\sqrt{n}}, \quad f_4(n) = \frac{1}{n}, \quad f_5(n) = 3^n,$$

$$f_6(n) = 2^{3^n}, \quad f_7(n) = n^n, \quad f_8(n) = 3^{2^n}, \quad f_9(n) = \log(\sqrt{n}), \quad f_{10}(n) = \log(2^n n^2),$$

$$f_{11}(n) = \log(n^{10}), \quad f_{12}(n) = n^2 + \log(n) + n$$

### שאלה 2:

הוכיחו את הטענות הבאות:

$$1. \quad n^3 - 5n = \Omega(n^3)$$

$$2. \quad \sqrt{n^2 + 9n} = O(n)$$

### שאלה 3:

נתון האלגוריתם הבא:

```
for i ← 1 to n
    k ← 1
    while (k < i)
        k ← k * 2
```

מהו זמן הריצה שלו במונחים אסימפטוטיים? יש להסביר ולהיצמד להגדרה.

## שאלה 4:

הוכיחו או הפריכו את הטענה הבאה :

יהיו  $f(n)$  ו- $g(n)$  פונקציות כך ש- $f(n), g(n) \geq 1$  לכל  $n$ .

מכאן :  $f(n) + g(n) = O(f(n) \cdot g(n))$

## חלק מעשי – הגשה בקובץ cpp בלבד (תכנות ב- cpp ולא c)

במשימה זו תתרגלו רשימות, מחסניות ותורים.

Word-ladder (סולם מילים) הוא רשימה של מילים בני אותו אורך שההבדל בין כל שתי מילים **צמודות** ברשימה הוא בדיוק של אות אחת. למשל :

stare

start

smart

במשימה זו תצטרכו למצוא סולמות מילים. על התוכנית שתכתבו לקבל מילת התחלה, מילת סיום ומילון של מילים ממנו עליה לבנות סולם-מילים ממילת ההתחלה למילת הסיום.

יש דרכים שונות לבצע משימה זו. עליכם יהיה לפעול לפי האלגוריתם (הנקרא חיפוש-לרוחב) הבא בלבד :

- טענו מקובץ את כל המילים של המילון לרשימה מקושרת.
- קבלו כקלט מילת התחלה ומילת סיום.
- הגדירו מחסנית ודחפו לתוכה את מילת ההתחלה.
- הגדירו תור והוסיפו אליו את **המחסנית**.
- חיזרו על התהליך הבא עד שתימצאו סולם מתאים או עד שהתור יתרוקן.
  - הוציאו את המחסנית שבראש התור והשוו את המילה שבראש המחסנית עם מילת הסיום.
  - אם המילים שוות – המחסנית מכילה סולם מילים רצוי.
  - אחרת חפשו ברשימת המילים את **כל המילים** ששונות באות אחת בדיוק מהמילה שבראש המחסנית ושאינן נמצאות כבר במחסנית. (למה?)
  - עבור כל מילה כזו שכפלו את המחסנית ודחפו את המילה למחסנית המשוכפלת, ואת המחסנית המשוכפלת הוסיפו לתור.

לכתיבת התוכנית עצמה עקבו אחרי הפרטים שלהלן :

הגדירו מחלקה בשם WordLadder שבתוכה תממשו את האלגוריתם. על ה-main לקרוא את המילון מתוך קובץ טקסט ואת מילת ההתחלה והסיום – כל אלו ישלחו כארגומנטים מתוך שורת הפקודה – command line. (שם הקובץ שמכיל מילון, מילת התחלה, מילת סיום). לאחר מכן, יופעל האלגוריתם. בסיום התוכנית תדפיס את הסולם שנמצא – כל מילה בשורה נפרדת. במידה ולא נמצא סולם מילים, יש להדפיס הודעה "No word-ladder found".

## הערות:

- הניחו שכל מילה בקובץ המילון יושבת בשורה נפרדת (אין צורך לבדוק זאת).
- הניחו שאורך כל המילים של הקלט (מילת ההתחלה, מילת הסיום ואוסף המילים במילון) שווה (אין צורך לבדוק זאת).
- ניתן להשתמש בתיקיות סטנדרטיות כדי לעבוד עם רשימה, תור ומחסנית.

## להלן דוגמת הרצה:

נניח שקובץ המילון מכיל את המילים הבאות:

```
dict2.txt - Notepad
File Edit Format View Help
POON
PLEE
SAME
POIE
PLEA
PLIE
POIN
```

נרצה לבדוק האם יש סולם מילים חוקי בין המילה **TOON** למילה **PLEA**.

להלן שורת הפקודה שנקליד כדי להפעיל את האלגוריתם:



להלן הפלט הנדרש שיתקבל **בדיוק**:

```
The ladder:
ladder size: 7
TOON
POON
POIN
POIE
PLIE
PLEE
PLEA
```

וזאת מכיוון שאכן קיים סולם מילים בין המילה TOON לבין המילה PLEA:

TOON - POON - POIN - POIE - PLIE - PLEE - PLEA

מצורף דוגמא לקובץ מילון.

## בהצלחה!