



## מבני נתונים: תרגיל תכנות מס' 2

דרישות	Python 3.6/7/8 (מומלץ דרך anaconda) Numpy - 1.18.5 * Pandas - 1.0.5 *
אמצעי הגשה	Moodle
יצירת קשר	<a href="mailto:Dorbar1@mail.tau.ac.il">Dorbar1@mail.tau.ac.il</a> <a href="mailto:liorweiner@mail.tau.ac.il">liorweiner@mail.tau.ac.il</a>
תאריך אחרון להגשה	23:55 07.06.25

### הנחיות כלליות:

- קראו בעיון את השאלות והקפידו שהתכניות שלכם פועלות בהתאם לנדרש.
- יש להגיש את כל קבצי הפתרונות לשאלות q1\_XY.py, q2\_XY.py, q3\_XY.py, לאחר החלפת XY במספר הקבוצה שלכם (מופיע במודל). במידה ומספר קבוצתכם חד ספרתי, יש לכתוב רק ספרה אחת. לדוגמא עבור קבוצה 5 בהגשת שאלה 1, **לא לכתוב** q1\_05 אלא q1\_5. אי עמידה בהנחיה זו תביא להורדת ציון.
- את השאלות הפתוחות יש להגיש בקובץ pdf נפרד, בקובץ זה חייבים להיות תעודות זהות של הסטודנטים של כל חברי הקבוצה, השמות ומספר הקבוצה בכתרת העמוד.
- חשוב: על כל קבוצה לבחור נציג/ה יחיד/ה שיגיש את הקובץ.
- יש להשתמש בפונקציות והמשתנים שכבר מופיעים בקבצי השלד של התרגיל **ללא שינוי** השמות המקוריים אי עמידה בהנחיה זו תביא להורדת ציון, ניתן ורצוי להוסיף פונקציות נוספות.
- יש לתעד את הקוד **באנגלית**, קוד ללא תיעוד יביא להורדת ציון.
- מותר ואף רצוי להשתמש בחבילות של python, החבילות המצוינות למעלה חייבות להיות בגרסה המצוינת! (אחרת הקוד עלול לא לרוץ) במידה והשתמשתם בחבילה נוספת אנא ציינו זאת בקובץ המצורף שלכם לצד הגרסה שבה השתמשתם.
- בדיקה עצמית: כדי לוודא את נכונות התוכניות במקרי קצה ובעבור קלטים שגויים, בכל שאלה מומלץ להריץ בדיקה עם מגוון קלטים שונים.
- כל שאלה צריכה להיות בקובץ נפרד – השתמשו בקבצים המצורפים. שימו לב לתיעוד אשר מרמז על סדר הפעולות.
- יש להשתמש בפורום הייעודי בלבד עבור תרגיל תכנות 2 על מנת לשאול שאלות הנוגעות לתרגיל תכנות זה.

### טיפים לתרגיל:

- בעת ניתוח סיבוכיות יש להסביר את הדרך ולא רק להציג את הפתרון הסופי.
- שאלה 1 חלק ב' שימו לב כי הרשימה A לא בהכרח ממוינת לפי סדר הגעת ההודעות. בנוסף הבקשה 'ABA' והבקשה 'aba' נחשבות לאותה בקשה. יש להתאים את הקוד לכך (אין הבדל בין אותיות קטנות לגדולות).
- שאלה 2 חלק ב' הוא חלק לא קל, אני ממליץ לכם לקרוא קצת חומרים שלא קשורים לקורס בנושא Huffman Coding על מנת לקבל רעיונות לפתרון הבעיה הזאת.
- בכל המקומות בהם מופיע pass עליכם למלא את הקוד שלכם וניתן למחוק אותם מהשלד.



## שאלה 1 – תור (35 נקודות)

א. ממשו מחלקת תור לפי הפסודו קוד שלמדתם בכיתה (10 נק') עליכם לממש את התור על ידי שימוש ב**רשימה מקושרת** באופן הבא:

Attributes:

head – האיבר בראש התור

tail – האיבר האחרון בתור

q\_size – אורך התור

Methods:

front() – מחזיר את האיבר הראשון בתור

empty() – מחזיר TRUE אם התור ריק וFALSE אחרת

enqueue(x) – מכניס איבר חדש x לסוף התור

dequeue() – מוחק את האיבר הראשון בתור ומחזיר את הערך שלו.

ב. זיהוי התקפה:

השרת של חברת מג'יק עובד בצורת תור. הוא מקבל בקשות ומכניס אותן לתור ולאחר מכן מטפל בהן לפי סדר הגעתן. "התקפה" מוגדרת כ-n בקשות מאותו סוג שהגיעו ברציפות כאשר כל n הבקשות הגיעו בתוך חלון זמן של t שניות.

עליכם לממש אלגוריתם שבהינתן קלט מסוים של רשימה A (נלקח לאחר יום עבודה) כאשר איבריה רשימות באורך 2 (האינדקס ה-0 בכל תת רשימה מייצג את סוג הבקשה (string) והאינדקס השני מייצג את השנייה שבה התקבלה הבקשה (float)), n המייצג את מספר הבקשות מאותו סוג וt שמייצג את חלון הזמן, האלגוריתם אחראי על הקפצת הודעת התקפה בליווי השנייה שבה הגיעה ההודעה הראשונה של ההתקפה. מבנה הפלט יהיה list באורך 3 כאשר האינדקס הראשון יהיה מסוג string של הודעת ההתקפה עם השנייה שבה ההודעה הראשונה של ההתקפה הגיעה. האינדקס השני יהיה הערך של השנייה שבה ההודעה הראשונה של ההתקפה הגיעה (float), האינדקס השלישי יהיה סוג הבקשה (string) **המקורי**. במידה ולא הייתה התקפה, יש להחזיר את הערך 0.

לדוגמא: בהינתן הקלט

```
A=[['Shelf',1.1], ['Shelf',1.2], ['shelf',1.3], ['shelf',1.5], ['Shelf',0.01], ['NEW USER',2.5], ['Rating',5.9]]
n=5,t=5
```

הפלט יהיה:

```
['There was an attack on second 0.01', 0.01, 'Shelf']
```

עבור הקלט  $n=5, t=20$ ,  $A=[['shelf',10], ['shelf',5], ['flower',1.76]]$ ,

הפלט יהיה 0 כי לא הייתה התקפה באותו יום.

1. כתבו אלגוריתם המייצר פתרון לבעיה זו. האלגוריתם חייב להשתמש בתכונות התור שבניתם וכן במתודות הרלוונטיות עבורו. בהינתן קלט מסוג list החזירו את הפלטים בהתאם להסבר מעלה.

**שימו לב:** אסור להשתמש במשתנים מסוג "list" בפתרון (חוץ מהקלט שלכם), כלומר, אין להגדיר משתני עזר חדשים מסוג "list". פתרון הבעיה עם משתנה/משתנים מסוג list יגרור הורדת נקודות משמעותית. הפתרון צריך להתבסס על אובייקט התור שבניתם בסעיף הקודם. אתם רשאים למיין את הרשימה שקיבלתם (אפשר להניח סיבוכיות  $O(n \log n)$ ), ולהשתמש בה כדי להכניס את איבריה לתור כרצונכם. מלבד זאת, אין להיעזר ברשימה בפתרון התרגיל. (20 נקודות)

2. מהו סיבוכיות האלגוריתם שכתבתם? (5 נקודות)



## שאלה 2 – ערימה (35 נקודות + 3 נקודות בonus)

- א. ממשו מחלקת ערמת מינימום לפי הפסודו קוד שלמדתם בכיתה (10 נק') עליכם לממש את הערימה על ידי שימוש במערך באופן הבא:

Attributes:

t – המערך שמחזיק את סדר הערימה

size – גודל הערימה

Methods:

insert(x) – מכניס איבר חדש x לערימה

delete\_min() – מוחק את האיבר המינימלי בערימה ומחזיר אותו.

heapify(i) – מבצע את פרוצדורת heapify

ב. בעיית איחוד החברות:

- בבעיה זו נתונות חברות שאותם רוצה לקנות איל הון ובכך להגדיל את הונו. מחיר החברות ידוע מראש, בנוסף למחיר הנקוב נדרש לשלם מס עבור כל קניה. ברגע שאיל ההון קונה חברה עליו לשלם מס פרופורציונלי לשווי האיחוד. מטרתו לאחד את כל החברות המוצעות למכירה לכדי חברה אחת גדולה באופן כזה שסך המס אותו ישלם יהיה הנמוך ביותר.
- נניח בשאלה זו, שהמס הפרופורציונלי  $= 100\%$  משווי החברות המאוחדות, כלומר אם מאחדים חברה בשווי x עם חברה בשווי y, המס על האיחוד יהיה  $x + y$
  - דוגמא מספרית לבעיה:

מחיר החברות				
חברה	א	ב	ג	ד
מחיר	2	4	7	9

- נשאלת השאלה מה הוא הסדר האופטימלי לאיחוד?

- אופציה א – לסדר איחוד:

	שלב א	שלב ב	שלב ג	סה"כ תשלום מס
החברות המאוחדות	א+ב	ג+ד	תוצר שלב א+ תוצר שלב ב	
גובה המס	6	16	22	$6 + 16 + 22 = 44$

- אופציה ב – לסדר איחוד:

	שלב א	שלב ב	שלב ג	סה"כ תשלום מס
החברות המאוחדות	א+ב	א+חברה ג	תוצר שלב ב+ חברה ד	
גובה המס	6	13	22	$6 + 13 + 22 = 41$

- הפתרון האופטימלי בדוגמא זו הוא 41

- ממשו אלגוריתם בעזרת שימוש בערימה, המקבל כקלט את רשימת החברות ושוויון לפני האיחוד, ומחזיר את ערך הפתרון האופטימלי כפלט (סך המס שישולם לאחר שכל החברות יאוחדו לחברה אחת). (15 נקודות)
- הפעילו את המימוש על קובץ הנתונים (data\_2.xlsx) והחזירו כפלט את ערך הפתרון האופטימלי, (פרמטר הגודל המקסימלי לערימה = 200). (2 נק')
- תארו באופן מילולי את האלגוריתם שמימשתם והסבירו מדוע הוא מביא לפתרון אופטימלי של הבעיה. (הסבר אינטואיטיבי, אין צורך בהוכחה פורמלית, אך הקפידו על הסבר ברור והגיוני, ניתן ללוות את ההסבר בדוגמאות). (4 נקודות)
- מהי סיבוכיות האלגוריתם שהצעתם? (4 נקודות)
- בנוס (3 נקודות): Huffman Coding היא שיטה לקידוד סימנים, אך שימושי מאוד לבעיות נוספות. הסבירו בקצרה מהו Huffman Coding ומהם השימושים שלו, הביאו דוגמה לכך והסבירו אותה.



### שאלה 3 – טבלאות גיבוב (30 נקודות)

בחלק זה תממשו סוגים שונים של Hash Tables (טבלאות גיבוב), אשר נבדלים בפונקציית הגיבוב ובהתמודדות עם התנגשויות. בקוד המצורף ישנו מימוש חלקי של מחלקה אשר לפי הגדרת הפרמטרים באתחולה בונה כל סוג של טבלת גיבוב – עליכם לעבוד לפי הסעיפים כדי להשלים את הקוד.

הקדמה – שיטת גיבוב הכפלה: (ניתן לקרוא על השיטה עוד בספר הקורס, פרק 12, עמ' 209)

$$h(k) = \lfloor m(kA - \lfloor kA \rfloor) \rfloor$$

כאשר:

k – המפתח אותו מכניסים לטבלת הגיבוב.

m – פרמטר נרמול לגודל טבלת הגיבוב.

A – מספר בין 0-1 אשר מוכפל במפתח.

לדוגמא:

$$A=0.6180339887, m=1000, k=123456$$

$$h(k) = \lfloor 1000(123456 \cdot 0.6180339887 - \lfloor 123456 \cdot 0.6180339887 \rfloor) \rfloor = 4$$

עבור מימוש כל מחלקה Hash Table, נדרוש:

Attributes:

size – גודל המערך

m – פרמטר לפונקציית הגיבוב (משפיע גם על גודל המערך).

A – פרמטר לפונקציית הגיבוב.

Methods:

member(key, value) – בודק האם איבר X בטבלה, מחזיר True/False ואת מספר הבדיקות שנדרשו כדי לקבל את התשובה.

insert(key) – מוסיף את X לטבלה ומחזיר את מספר הבדיקות שנדרשו כדי לבצע את הפעולה.

delete(key) – מוחק את X מהטבלה ומחזיר את מספר הבדיקות שנדרשו כדי לבצע את הפעולה.

א. השלימו את החסר במתודות hash\_function\_2 ו hash\_function. (5 נקודות)

ב. במתודה insert השלימו את המקרה "OA\_Quadratic\_Probin". collision\_handling (5 נקודות)

ג. במתודה delete השלימו את המקרה "OA\_Double\_Hashing". collision\_handling (5 נקודות)



ד. בקובץ data.xlsx סטים של נתונים.

בנו קוד המכניס כל סט נתונים  $(X,Y,Z)$  לכל אחת מה- Hash Tables הבאות: (5 נקודות)

	פונקציית גיבוב	התמודדות עם התנגשויות	פרמטרים
1	חלוקה	שרשור	$m=149$
2	חלוקה	מיעון פתוח - בדיקה ריבועית	$m=149$
3	חלוקה	מיעון פתוח - גיבוב כפול	$m_1=149, m_2=97$
4	כפל	שרשור	$m=149, A=0.589$
5	כפל	מיעון פתוח - בדיקה ריבועית	$m=149, A=0.589$
6	כפל	מיעון פתוח - גיבוב כפול	$m_1=149, m_2=97, A_1=0.589, A_2=0.405$

ה. עבור כל צמד Hash Tables וסט נתונים, צרו את "מדד היעילות"  $\left(\frac{\text{מספר הפעולות}}{\text{מספר הרשומות}}\right)$  וחוו את

דעתכם על התוצאות (התייחסו בתשובתכם לאופי הנתונים ולשוני בין טבלאות הגיבוב). (5 נקודות)

ו. לכל סט נתונים, בחרו באחת מפונקציות הגיבוב והציעו פרמטר גיבוב חדש כך שמדד היעילות הטוב ביותר שחישבתם בסעיף ה' ישתפר. הסבירו את העיקרון שליווה אתכם בבחירת השיטה והפרמטרים. (5 נקודות)

הערה: (בסעיפים א', ב', וג' השלימו את הקוד במקום replace men)