

מבוא למדעי המחשב – שפת פייתון (234128)

חורף 2025

תרגיל בית 5

מועד הגשה: 04/02/2025 בשעה 20:00

זה היום האחרון של סמסטר חורף

הגשה רק בבודדים

חשוב לשמור את קוד
אישור ההגשה שמקבלים
מהמערכת כשמגישים!!

המתרגל האחראי על תרגיל זה: **הדר מטלו**

משרד: טאוב 710

E-mail: hadar.m@campus.technion.ac.il

שעת קבלה רגילה: יום ב' 14:00-15:00, בזום בקישור <https://technion.zoom.us/j/2269702002>,

שעות קבלה מיוחדות לשאלות על התרגיל:

יום ה' (23.01.2025) 15:00-16:00, בזום בקישור <https://technion.zoom.us/j/2269702002>

וביום ד' (29.01.2025) 13:00-14:00, טאוב 219

מילואים:

- **מילואים:** שיטת ההקלות של קורס פייתון מותאמת לסילבוס של הקורס. **אין להסתמך בשיטות בקורסים אחרים!**
- לטיפול בבקשות הארכה להגשה או ביטול הגשה או עזרה בחומר הקורס יש לפנות ל**ילנה** (enovbari@gmail.com), המתרגלת האחראית של הקורס.
- באופן רגיל למילואים לא ממושכים כל יום מילואים = יום דחייה. על מנת לקבל דחייה עבור ת"ב 0, יש לשלוח מייל בצירוף אישור מילואים (או מסמך שקול) למתרגלת אחראית ילנה.
- בונוס של 5 נקודות ניתן לסטודנטים/יות שמגישים אלקטרונית דרך אתר הקורס, בזמן עד למועד הגשה. לא ניתן טיפול ידני בבונוס (ללא קשר לנסיבות ואישורים).

הוראות הגשה:

- הגשה אלקטרונית **בבודדים**.
- תרגיל שיוגש בזמן (עד למועד ההגשה ולא בתקופת הארכה) יקבל אוטומטית **בונוס של 5 נקודות**.
- תאריך הגשת ערעורים יתפרסם עבור כל תרגיל הבית לחוד.
- **ערעורים על הדפסות שגויות בפלט שלכם לא יתקבלו.** עליך לבדוק את תוכנית שלך עם DiffMerge לפני ההגשה.

הנחיות:

- אנא קרא/י את השאלות בעיון לפני שתתחיל/י בפתרון.
- ניתן להניח כי הקלט תקין, אלא אם כן נאמר אחרת.
- יש לבדוק את אתר ה-FAQ של התרגיל לעיתים תכופות על מנת לקבל עדכונים על התרגיל.

- הגשה אלקטרונית **בבודדים** היא דרך אתר הקורס. הגיש קובץ **hw5.zip** (ולא **zip** פורמט אחר (!) כגון rar). **השם** של קובץ חייב להיות **באנגלית ללא רווחים**. אין דרישות נוספות לשם הקובץ, כלומר גם sharon.zip מקובל מבחינת המערכת. הקובץ יכול בתוכו **רק** את הקבצים הבאים (לא לעשות zip על התיקיה בה נמצאים הקבצים להגשה – במקרה כזה ההגשה לא תתקבל ע"י המערכת!):

- קובץ **students.txt** עם שמך באנגלית, מספר תעודת הזהות וכתובת האימייל שלך.
- קובץ פתרון **hw5q1.py** המכיל את הקוד לשאלה עם כל הסעיפים.
- קובץ pdf.12345678 המכיל את הקוד וההדפסות שלכם (ראו עמוד אחרון).
- **מאד חשוב ששם קובץ PDF יהיה ת"ז הסטודנט.**
- יש להקפיד להגיש את כל הקבצים בדיוק עם השמות שמופיעים לעיל. הגשה שלא תעמוד בתנאי זה לא תתקבל ע"י המערכת!

שימו לב- זהו תרגיל יבש:

- הבדיקה היא ידנית!
- הבדיקה תסתכל על הקוד שכתבתם ולא רק על הפלט.

שימו לב לאתר ה- FAQ של התרגיל, אשר מכיל עכדונים ותשובות לשאלות נפוצות, ויעודכן מפעם לפעם. חשוב להתעדכן לעיתים תכופות – זה יכול לחסוך הרבה בעיות, ונקודות שירדו בבדיקה שכשלה. בפרט, לפני פנייה למתרגל האחראי בדקו את הכתוב על מנת לחסוך לעצמכם זמן המתנה לתשובה, שבדרך כלל תהיה הפניה ל- FAQ בכל מקרה. ניתן למצוא את ה-FAQ בלינק.

https://docs.google.com/document/d/1sulv0p9_ZQRvHc4E-d3qi0bnZavgBG0ENsV7S9mo-1o/edit?usp=sharing

! חובה להתעדכן בקובץ השאלות הנפוצות !

https://docs.google.com/document/d/1sulv0p9_ZQRvHc4E-d3qi0bnZavgBG0ENsV7S9mo-1o/edit?usp=sharing

ת"ב 5 מבוסס על תרגולים 9,10 והרצאות 9,10.

אין להשתמש בחומר שנלמד אחריהם.

תרגיל בית 5

אורי חובב ספורט מנסה להכין אתר עם כל מיני נתונים על תחרויות ספורט אהובות עליו. עזרו לאורי לכתוב פונקציות נחוצות לאתר.

למטרת התרגיל, סיפקנו לכם קובץ csv בפורמט של הקבצים שאיתם האתר יעבוד:

`sports_events.csv` - עליו תצרכו לעשות ההרצה הסופית לצורך ההגשה (ראו ההסבר בסוף).

ניתן להניח שבכל קבצי csv שאיתם אורי עובד, אין מחרוזת עם יותר מ20 תווים.

לאורך כל התרגיל אפשר ומומלץ להשתמש בפונקציות שמימשותם בסעיפים קודמים.

בנוסף – אפשר להניח בכל הסעיפים ששמות התחרויות ושמות האתלטים אינם עולים על 20 אותיות כל אחד.

חשוב מאוד!!! - לנוחותכם סיפקנו לכם קובץ ג'ופיטר `HW5_structure.ipynb` שמכיל בדיוק את המבנה של ההגשה הסופית! כל מה שאתם צריך לעשות, זה למלא את הפונקציות עם המימושים שלכם, ואז להריץ הכל. **חייבים להשתמש במבנה זה להימנע מהגשה שגויה.**

סעיף 1

הקובץ `sports_events.csv` מכיל את מאגר התחרויות. כשלב ראשוני, כתבו פונקציה

```
def read_event_data(file_path)
```

שמקבלת נתיב לקובץ CSV `file_path` (ניתן להניח שהמסלול חוקי), טוענת אותו ו**מחזירה** אותו (לא להדפיס!!)

סיבוכיות זמן: $O(1)$

יש להניח שסיבוכיות הזמן של פונקציית `read_csv` של pandas היא קבועה.

הערה: תטענו ותדפיסו את 5 השורות הראשונות בשביל עצמכם, ותנסו להבין את המבנה של המאגר וכיצד שמורים בו פרטי התחרויות לפני שממשיכים בפתרון. יש למחוק הדפסה זו לפני ההגשה.

תשתמשו בקטע קוד זה לבדיקה עצמית (תמחקו לפני ההגשה):

```
file_path = 'sports_event.csv'
events = read_event_data(file_path)
print(events.head(10))
```

ותקבלו את ההדפסה הבאה:

	Event	Date	Athlete	Country	Score
0	Shot Put	2024-03-07	Liam Taylor	IND	4.79
1	Shot Put	2024-03-11	John Doe	USA	3.90
2	Long Jump	2024-03-05	Michael Brown	GBR	7.35
3	Long Jump	2024-03-09	Jane Smith	JAM	8.15
4	Shot Put	2024-03-10	Jane Smith	JAM	5.06
5	Long Jump	2024-03-05	Anna Martinez	GER	4.34
6	Discus Throw	2024-03-10	Michael Brown	GBR	9.70
7	Discus Throw	2024-03-03	John Doe	USA	5.61
8	Discus Throw	2024-03-02	David Kim	CHN	8.63
9	Shot Put	2024-03-03	James Wilson	KEN	2.58

סעיף 2

ממשו את הפונקציה:

```
def top_athlete(events, event_name)
```

שמקבלת קובץ CSV טעון (כלומר הפלט של הסעיף הקודם) events, ושם של תחרות, event_name, ומחזירה (לא להדפיס!) את השם ואת הscore של האתלט עם הscore הכי גבוהה בתחרות הנתונה.

שימו לב שהפונקציה מחזירה את שני הערכים, ולא מדפיסה כלום.

ניתן להניח שהevent_name הוא שם של event תקין שקיים במאגר.

ניתן להניח שבדיוק אתלט אחד השיג את הציון הכי גבוהה.

סיבוכיות זמן: $O(n)$ כאשר n הוא מספר ה-events בקובץ csv.

תשתמשו בקוד הבא לבדיקה עצמית (למחוק לפני הגשה!):

```
file_path = 'sports_event.csv'
events = read_event_data(file_path)
print(top_athlete(events, "Long Jump"))
```

תקבלו את ההדפסה הבאה:

```
('David Kim', 9.67)
```

סעיף 3

ממשו את הפונקציה:

```
def event_scores(events, event_name)
```

שמקבלת קובץ CSV טעון (כלומר הפלט של סעיף 1) events, ושם של תחרות (event_name) ומחזירה (לא להדפיס!) רשימה של כל הscores של אותה event.

אורך הרשימה המוחזרת צריכה להיות שווה למספר המופעים של אותה event ב-csv.

סיבוכיות זמן: $O(n)$ כאשר n הוא מספר ה-events בקובץ csv.

תשתמשו בקוד הבא לבדיקה עצמית (למחוק לפני הגשה!):

```
file_path = 'sports_event.csv'
events = read_event_data(file_path)
print(sorted(event_scores(events, "Long Jump")))
```

ותקבלו את ההדפסה:

[0.35, 0.42, 1.4, 2.14, 2.65, 4.34, 6.48, 7.35, 8.15, 8.97, 9.67]

סעיף 4

ממשו את הפונקציה

```
def unique_events(events)
```

שמקבלת קובץ CSV טעון (כלומר הפלט של סעיף 1), ומחזירה (לא להדפיס!) רשימה עם סוגי הevents השונים.

סיבוכיות זמן: $O(nm)$ כאשר n הוא מספר ה-events בקובץ csv ו- m הוא מספר סוגי הevents השונים.

תשתמשו בקוד הבא לבדיקה עצמית (למחוק לפני הגשה!):

```
file_path = 'sports_event.csv'
events = read_event_data(file_path)
print(sorted(unique_events(events)))
```

ותקבלו את ההדפסה:

['Discus Throw', 'High Jump', 'Long Jump', 'Shot Put']

סעיף 5

ממשו את הפונקציה

```
def count_events(events)
```

שמקבלת קובץ CSV טעון (כלומר הפלט של סעיף 1), ומחזירה (לא להדפיס!) שתי רשימות – רשימה של התחרויות השונות במאגר, ורשימה שמכילה את מספר התחרויות שהתרחשו בכל תחרות ברשימה הראשונה. סדר ההופעות של התחרויות ברשימה הראשונה לא משנה אך הערכים ברשימה השנייה חייבים להיות בהתאמה לרשימה הראשונה.

השתמשו בפונקציה unique_events שמימשתם בסעיף הקודם.

סיבוכיות זמן: $O(nm)$ כאשר n הוא מספר ה-events בקובץ csv ו-m הוא מספר סוגי ה-events השונים.

תשתמשו בקוד הבא לבדיקה עצמית (למחוק לפני הגשה!):

```
file_path = 'sports_event.csv'
events = read_event_data(file_path)
print(count_events(events))
```

ותקבלו את ההדפסה (לא בהכרח באותו סדר):

(['Shot Put', 'Long Jump', 'Discus Throw', 'High Jump'], [18, 11, 8, 13])

בדוגמא לעיל אנו רואים שה'Shot Put' event הופיע 18 פעמים במאגר, ה'Long Jump' event הופיע 11 פעמים במאגר, ה'Discus Throw' event הופיע 8 פעמים במאגר וה'High Jump' event הופיע 13 פעמים במאגר.

חלק יבש - ענו על השאלות הבאות

- 1) הסבירו בקצרה מדוע המימוש שלכם עומד בסיבוכיות הנדרשת?
- 2) איך סיבוכיות הזמן תשתנה במידה ולא היה נתון ששם של תחרות לא עולה על 20 אותיות?

סעיף 6

ידוע שבכל התחרויות שבכל התחרויות score המקסימלי הוא 10.

ממשו את הפונקציה

```
def analyze_scores(events, event_name)
```

שמקבלת קובץ CSV טעון (כלומר הפלט של סעיף 1) (events) ושם של תחרות (event_name) הפונקציה מחזירה (לא להדפיס!) היסטוגרמה המייצגת כמה פעמים השיגו score בכל טווח של scores מ 0 ועד 10 בקפוצות של אחד.

השתמשו בפונקציה event_scores שמימשתם בסעיף 3.

לדוגמא:

עבור המאגר עם עשר שורות:

	Event	Date	Athlete	Country	Score
0	Shot Put	2024-03-07	Liam Taylor	IND	4.79
1	Shot Put	2024-03-11	John Doe	USA	3.90
2	Long Jump	2024-03-05	Michael Brown	GBR	7.35
3	Long Jump	2024-03-09	Jane Smith	JAM	8.15
4	Shot Put	2024-03-10	Jane Smith	JAM	5.06
5	Long Jump	2024-03-05	Anna Martinez	GER	4.34
6	Discus Throw	2024-03-10	Michael Brown	GBR	9.70
7	Discus Throw	2024-03-03	John Doe	USA	5.61
8	Discus Throw	2024-03-02	David Kim	CHN	8.63
9	Shot Put	2024-03-03	James Wilson	KEN	2.58

ועבור הקריאה לפונקציה

`analyze_scores(events, "Long Jump")`

נקבל את הפלט:

`[0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0]`

כאשר התא הראשון ברשימה מייצג את מספר התחרויות באותו event שקיבלו score בין 0 ל-1 (לא כולל) התא השני ברשימה מייצג את מספר התחרויות באותו event שקיבלו score בין 1 ל-2 (לא כולל) וכו'. כלומר הערכים ברשימה מתאימים לscores:

`[0-1, 1-2, 2-3, 3-4, 4-5, 5-6, 6-7, 7-8, 8-9, 9-10, 10]`

בדוגמא לעיל

'Long Jump' מופיע 3 פעמים במאגר. פעם אחת עם ציון בין 4-5, פעם אחת עם ציון בין 7-8 ופעם אחת עם ציון בין 8-9. לכן הסטנוגרמה המוחזרת היא:

`[0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0]`

סיבוכיות זמן: $O(n)$ כאשר n הוא מספר ה-events בקובץ csv.

תשתמשו בקוד הבא לבדיקה עצמית (למחוק לפני הגשה!):

```
file_path = 'sports_event.csv'
events = read_event_data(file_path)
print(analyze_scores(events, "Long Jump"))
```

ותקבלו את ההדפסה:

`[2, 1, 2, 0, 1, 0, 1, 1, 2, 1, 0]`

חלק יבש - ענו על השאלות הבאות

- 1) הסבירו בקצרה מדוע המימוש שלכם עומד בסיבוכיות הנדרשת?
- 2) הסבירו את ההבדל בסיבוכיות של הפונקציה הזאת לאומת הפונקציה בסעיף הקודם. למה ניתן להשיג סיבוכיות יותר טובה בפונקציה הזאת?
- 3) בהנחה שסיבוכיות הזמן של הפונקציה event_scores נשארת $O(n)$, האם הסיבוכיות של הפונקציה analyze_scores תושפע אם לא היה נתון שאורך שם של תחרות/ שם של אתלט לא יכול לעלות על 20 תו?

סעיף 7

ממשו את הפונקציה

```
def analyze_event(file_path, event_name)
```

המקבלת מחרוזת של מסלול לקובץ CSV (ניתן להניח שהמסלול חוקי), ושם של event. הפונקציה מבצעת את הפעולות הבאות (ראו גם את פורמט הפלט בדוגמה):

- טוענת את קובץ ה-CSV שנמצא ב-file_path (ניתן להניח שמסמך חוקי).
- מדפיסה למסך את שם האתלט שהשיג את הscoren הכי גבוהה ב-events הנתון. ניתן להניח שבדיוק אתלט אחד השיג את הציון הכי גבוהה.
- משרטטת למסך את היסטוגרמת scoresn של אותה event.

דוגמה ופורמט הפלט

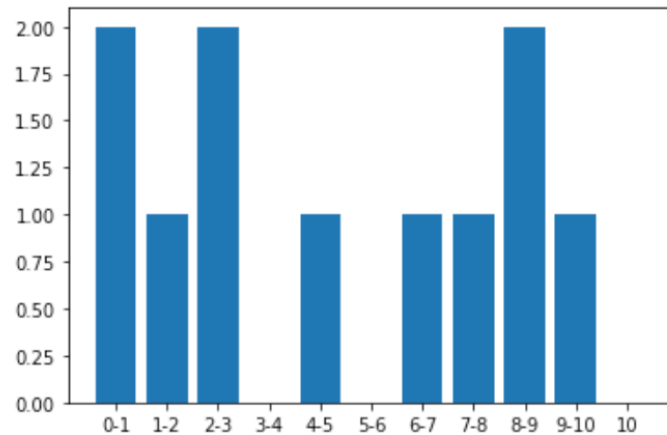
למשל עבור sports_event.csv שמסופק עם התרגיל:

ההרצה של הקוד:

```
file_path = 'sports_event.csv'
analyze_event(file_path, "Long Jump")
```

תדפיס למסך את הפלט בפורמט הבא:

Top athlete in Long Jump is David Kim with score: 9.67



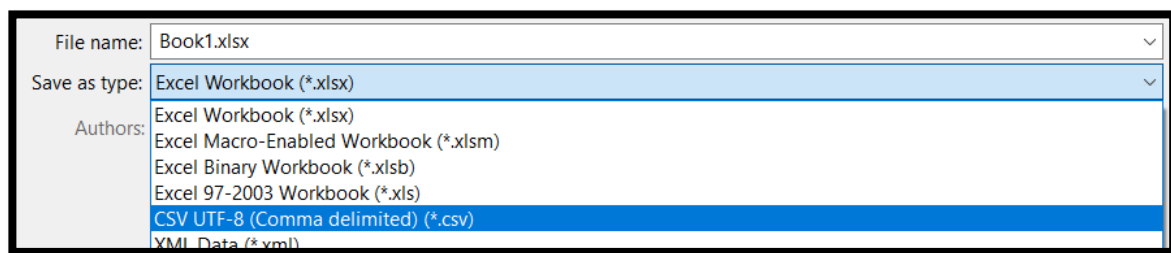
הפונקציה רק מדפיסה פלטים למסך והיא לא מחזירה שום ערך.

הרצות

אחרי שמימשתם את הפונקציות בכל הסעיפים, והשתמשתם בקטעי קוד שנתנו לבדיקה עצמית, אתם נדרשים להגיש את הקוד שלכם, עם הרצות על המאגר שסיפקנו – `sports_event.csv`.

בנוסף, אתם מתבקשים בסעיף האחרון לבנות דוגמה משלכם – לייצור קובץ מאגר חדש בשם `my_events.csv`, ולהריץ את הפונקציה `analyze_event` עליו.

כדי להכין קובץ `csv` משלכם, תיצרו קובץ `excel`, תמלאו אותו, ותשמרו אותו עם סיומת `csv`.



חשוב – לשים את קבצי ה-`csv` באותה תיקיה של הקוד לפני ההרצות.

לאחר המימוש וההרצות אתם יכולים להמיר את המחברת לקובץ `pdf`. ראו הסבר למטה.

חשוב מאוד!!! - לנוחיותכם סיפקנו לכם קובץ ג'ופיטר `HW5_structure.ipynb` שמכיל בדיוק את המבנה של ההגשה הסופית! כל מה שאתם צריך לעשות, זה למלא הפונקציות עם המימושים שלכם, ואז להריץ הכל. חייבים להשתמש במבנה זה להימנע מהגשה שגויה.

הוראות הגשה

אתם מתבקשים להגיש קובץ `ZIP` עם הקבצים הבאים:

- קובץ `students.txt` עם שמך באנגלית, מספר תעודת הזהות וכתובת האימייל שלך.
- קובץ פתרון `hw5.py` המכיל את הפונקציות הנדרשות.
- קובץ `pdf.12345678` המכיל את הקוד, ההדפסות ותשובות לשאלות המילוליות (שם הקובץ הוא מספר תעודת הזהות).

מסמך PDF

במסמך ה-PDF שאתם מגישים צריכים להיות הדברים הבאים:

1. הפונקציות שנדרשתם לכתוב וההרצות הדרושות:

- זה קובץ `HW5_structure.ipynb`, אחרי שהשלמתם את המקומות החסרים עם מימושי הפונקציות והרצתם הכל.

- מותר לכם להוסיף תאים לצורך מימוש פונקציות עזר משלכם בכל הסעיפים.

את המחברת הזאת אתם ממירים לקובץ `pdf`. **הוראות ליצירת PDF מהjupyter notebook מופיעים בהמשך.**

2. תשובות לשאלות היבשות יכולות להיות רשומות באחת מהדרכים הבאות:

- מסמך WORD או TEXT כלשהו שהומר ל-PDF וחובר למסמך עם הפונקציות וההרצות
- תא נפרד ב-jupyter notebook שם כתובות התשובות (ראו מבנה לדוגמא - `HW5_structure`).
- סריקה של התשובות הכתובות על דף בכתב קריא וברור

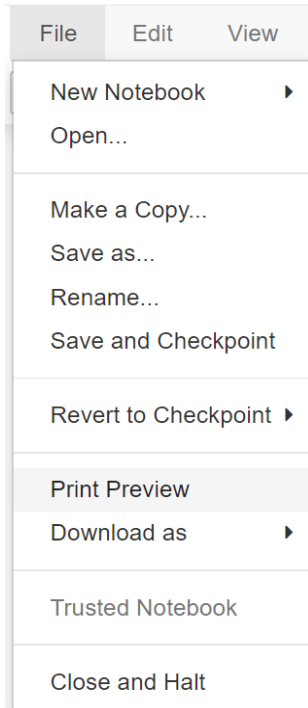
שימו לב! אתם מגישים קובץ PDF אחד בלבד!

ישנם המון אתרים חינוכיים המאפשרים איחוד של קבצי PDF. אתר לדוגמה:

https://www.ilovepdf.com/merge_pdf

יצירת PDF מ-Jupyter notebook:

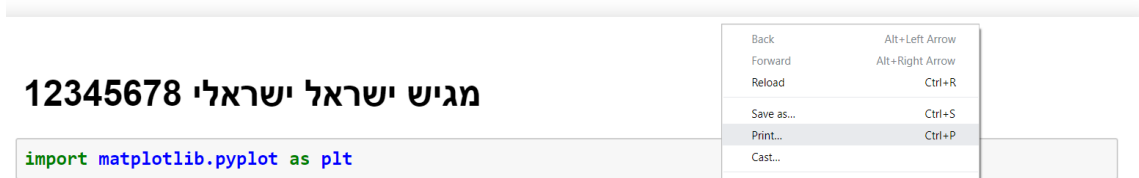
לחצו על File:



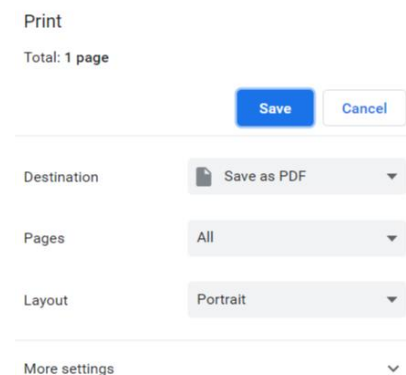
ואז בחרו ב- Print Preview כדי לראות את איך תראה המחברת ג'ופיטר שלכם כשהיא תודפס (או כשתישמר כקובץ PDF)

כשאתם מוכנים, לחצו על הכפתור הימני בעכבר ואז בחרו ב- print או לחצו `ctrl+P`:

ניתן



להעביר את העבודה לpdf ע"י בחירה באפשרות - Save as PDF:



מאד חשוב ששם קובץ PDF יהיה ת"ז הסטודנט.

בהצלחה! 😊