

מבוא לבינה מלאכותית (89-570)

תרגיל בית 1: בעיות חיפוש

#### מטרות המשימה

- התנסות בייצוג בעיות אקטואליות ומציאותיות כמרחבי מצבים.
- התנסות באלגוריתמי חיפוש.
- תכנון ווריאציה אלגוריתמית בהתאם לבעיה.

#### הערות:

- תאריך הגשה: 1.12.19
- המטלה להגשה **ביחידים בלבד!**



## תיאור הבעיה

תוכנת MyWay מוצאת את המסלול בעל זמן הנסיעה הצפוי הקצר ביותר, כאשר זמן הנסיעה בכל קטע משוערך על ידי המהירות הממוצעת באותו קטע.

הנכם יזמים שהקימו את חברת BetterWay המנסה ליצר אלטרנטיבה לאפליקציית MyWay הותיקה.

התוכנה תעבוד עם קובץ נתונים המייצג את רשת הכבישים של ישראל. אנו ביצענו הורדה של מפת ישראל מאתר [www.openstreetmap.org](http://www.openstreetmap.org) והמרה לפורמט שיהיה נוח לעבודה עם python ושיכלול

רק את המידע הרלוונטי מתוך שלל הנתונים המקוריים, זהו הקובץ `israel.csv`.

## חלק 1 – מבוא והקדמה

1. **שאלה יבשה:** פיתחו את הקבצים `db/israel.csv` ותארו את המבנה שלהם במדויק: מה מייצגת כל שורה ומה הפרמטרים בה. לצורך כך, עיינו בקוד של השגרה `load_map_from_csv` בקובץ הקוד המסופק `ways/graph.py`.
2. **שאלה רטובה:** מלאו את תוכן שגרת העזר `map_statistics` המחשבת פרמטרים המאפיינים את המפה הנתונה.

3.

יש לאפשר להריץ את הקוד עבור הסעיף דרך שורת הפקודה

```
$ python stats.py
```

הפלט צריך להיות תוצאת הדפסה פשוטה של המילון אותו מחזירה הפונקציה.

צרו 100 בעיות חיפוש אקראיות הבעיות צריכות להכיל את צומת ההתחלה וצומת הסיום.

כתבו את צומת ההתחלה וצומת הסיום בקובץ `problems.csv` בפורמט הבא

s1, t1

s2, t2

s3, t3

...

s20, t20

הערה חשובה: חלקים מהמפה אינם קשירים, לכן על אחריותכם לסנן בעיות חיפוש שאינן פתירות.

## חלק 2 – UCS

בחלק זה נממש את אלגוריתם UCS כפי שנלמד בכיתה.

לפני מענה על השאלות בחלק הזה מומלץ לעבור על כל הקוד המצורף עם התרגיל.

4. כתבו בפייתון פונקציה `find_ucs_rout` המקבלת שלושה פרמטרים - צומת התחלה, צומת סיום ופונקציית מחיר ומחזירה את המסלול המהיר ביותר מנקודת המוצא ליעד בעזרת אלגוריתם UCS (פונקציית המחיר ממומשת בקוד תחת `tools.compute_distance`).
5. השתמשו באותן 100 בעיות חיפוש שיצרתם בסעיף 3 והריצו עליהן חיפוש UCS. עבור כל אחת מהבעיות, פלטו לקובץ `results/UCSRuns.txt` את זמני הנסיעה.

```
הקובץ main.py מרכז את הממשק אל שורת הפקודה; יש לכתוב בו מעט ככל הניתן.  
יש לאפשר להריץ את הקוד עבור הסעיף דרך שורת הפקודה.  
למשל, אם נקודת המוצא היא 30 ונקודת היעד 55:  
$ python main.py ucs 30 55  
על הפלט להיות רשימה פשוטה של מספרי צמתים, כולל קצוות:  
30 21 44 73 55
```

## חלק 3 - A\*

בחלק זה נממש את A\* כמו שנלמד בכיתה.

6. כתבו בפייתון פונקציה `find_astar_route` המקבלת ארבעה פרמטרים - צומת התחלה, צומת סיום, פונקציית מחיר ופונקציה יוריסטית, ומחזירה את המסלול המהיר ביותר מנקודת המוצא ליעד בעזרת אלגוריתם A\*.
7. נרצה להפעיל את מתודת החיפוש הנ"ל עם פונקציית מחיר של זמן נסיעה ויוריסטיקה קבילה ומונוטונית מתאימה. פונקציית המחיר ממומשת בקוד, עליכם לממש יוריסטיקה מתאימה.
8. הסבירו בדו"ח את מבנה הקלט והפלט של שתי הפונקציות הנ"ל בקוד ובנוסף איזו פונקציה יוריסטית בחרתם ומדוע היוריסטיקה אכן קבילה.
9. השתמשו באותן 100 בעיות חיפוש שיצרתם בסעיף 3 והריצו עליהן חיפוש. עבור כל אחת מהבעיות, פלטו לקובץ `results/AStarRuns.txt` את הזמן המשוערך ע"י היוריסטיקה מהמוצא ליעד ואת זמן הנסיעה בפועל. הציגו בדו"ח גרף ובו נקודה לכל אחת מההרצות הנ"ל. מה ניתן ללמוד מהגרף על הקשר בין המשתנים?

```
הקובץ main.py מרכז את הממשק אל שורת הפקודה; יש לכתוב בו מעט ככל הניתן.  
למשל, אם נקודת המוצא היא 30 ונקודת היעד 55:  
$ python main.py astar 30 55  
על הפלט להיות רשימה פשוטה של מספרי צמתים, כולל קצוות:  
30 21 44 73 55
```

10. בהנחה שהרצתם את אלגוריתם  $A^*$  בדיוק לפני שהעומס בכביש התחיל, האם בהכרח המסלול המתקבל יהיה אופטימלי? הסבירו את קביעתכם.

#### חלק 4 - $IDA^*$

11. חזרו על סעיפים 6-10 עם שימוש באלגוריתם  $IDA^*$  הממומש תחת המתודה `find_idastar_route` כ- `TreeSearch` במקום  $A^*$

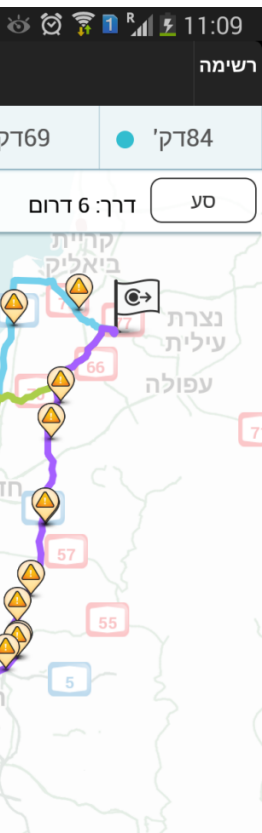
#### חלק 5 - סיכום

12. השתמשו במתודה `draw.plot_path` בכדי ליצר מפה של כל אחד מהפתרונות שיצרתם. צרפו את התמונות תחת תקייה `solutions_img`.

13. לכל אחד מהאלגוריתמים ( $IDA^*$ ,  $A^*$ ,  $UCS$ ), כתבו את זמן הריצה הממוצע וסטיית התקן\*. מי מהאלגוריתם רץ בזמן הקצר ביותר? מדוע?

\*להגדרה של סטיית תקן, אפשר להסתכל [כאן](#)

14. **שאלת רשות:** בתכנת Waze, בהינתן נקודת מוצא ונקודת יעד יש אפשרות לקבל הצעות לכמה מסלולים אלטרנטיביים. אתם מתבקשים להציע רעיון אלגוריתמי לסיפוק אפשרות שכזו. ציינו אילו שיקולים מעניינים אתכם בבחירת המסלולים ולמה הם עשויים להיות חשובים למשתמש. תארו את השיטה מילולית או באמצעות פסאודוקוד ברמת אבסטרקציה גבוהה והסבירו מדוע השיטה שבחרתם תספק מענה טוב לשיקולים שציננתם לגבי המסלולים. דוגמה לפלט סביר היא למשל שלושת המסלולים הבאים מרמת ישי לתל אביב. **בנוסף ינתן לרעיונות יצרתיים במיוחד.**



## דו"ח מחקר

- זהו חלק חשוב מאוד בעבודה ויהווה גורם מרכזי בקביעת ציונכם.
- אין צורך להדפיס את הדו"ח. הגישו אלקטרונית בלבד כקובץ *PDF*.
- הקפידו לכתוב את האימיילים שלכם גם בראש הדו"ח לצורך קבלת המשוב.
- נמקו היטב את כל תשובותיכם, הסבירו את המוטיבציה ליורסטיקה, הסבירו תוצאות מעניינות שרואים בגרפים או בכל מידע אחר שאספתם ונראה לכם רלוונטי.

## הקוד

- עליכם להגיש כל קוד שנכתב לצורך ביצוע המטלה.
- הנחיות לגבי קובצי קוד בפיתוח מופיעות כאן: [www.python.org/dev/peps/pep-0008/](http://www.python.org/dev/peps/pep-0008/)
- מומלץ לעבוד לפיהן אך אי עמידה בהן לא תפגע בציון.
- תיעוד למבנה התיקיות מופיע בתיקיה *docs*.
- הקוד שסופק לכם יכול להיות מוכל בפרויקט המוגש, מלבד קובץ המפה בשל גודלו.
- במידה ואתם חושבים שיש צורך **חיוני** בשינוי כלשהו בקבצי הקוד המסופקים - אתם מוזמנים לפנות לאמיר במייל. שימו לב שבפיתוח ניתן לשנות אובייקטים רבים באופן דינמי.
- טרם הגשה:
  - עדכנו את הקובץ *docs/details.txt*. עם הפרטים שלכם.
  - שימו לב שכל הפנייה למיקום קובץ/תיקיה כלשהי בקוד תהיה רלטיבית ולא אבסולוטית (*relative path*), כך שהקוד יעבוד כפי שהוא על כל מחשב בכל מיקום שנבחר לתיקית הפרויקט. הקפידו לבדוק זאת לפני ההגשה!
  - נא לא להשתמש בחבילות נוספות על מה שכבר קיים בקוד שניתן לכם. אם יש צורך חריג, נא לבדוק עם המתרגלים.
  - שימוש בכל קוד חיצוני מצריך הצהרה מפורשת על המקור שלו. במקרה של שימוש בספריה מסודרת, יש להוסיף את שמה לקובץ *docs/dependencies.txt*.
  - הסירו את קובץ המפה מהתיקיה *db/israel.csv*.
  - כווצו את התיקיה לארכיון *ZIP/RAR*.
- את הארכיון שיצרתם יש להגיש אלקטרונית דרך ה *submit*
- את הדו"ח יש להגיש ב *google classroom*..

**בהצלחה ... ותיהנו!**