תרגיל 1- בינה מלאכותית- אושר אלחדד 318969748, מייל oshereli2@walla.com

חלק 1 – מבוא והקדמה *(10 נקודות)*

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*.

תשובה-

1. כל שורה בקובץ db/Israel.csv הינה צומת (בכביש) שכל צומת מורכבת למעשה ממספר פרמטרים (מספר עמודות בכל צומת-שורה):
ID- מזהה הצומת
Lat (latitude)- קו רוחב
Lon (longitude)- קו אורך
רשימה של מחלפים (link) שכל מחלף מורכב למעשה ממספר פרמטרים (שמופרדים ע"י @ ביניהם):
   - צומת יעד (מיוצג ע"י ID)
   - מרחק
   - סוג הכביש
   - (צומת המקור הינה השורה שבה המחלף נמצא- המייצגת את הצומת עם ID)

כפי שניתן לראות גם בקובץ graph.py:

```
# define class Link
Link = namedtuple('Link',
        ['source', 'target',  #  int (junction indices)
         'distance',          #  float
         'highway_type',      #  int < len(road_info.ROAD_TYPES)
         'link_params',       # a tuple of 2 floats.
        ])


# define class Junction
Junction = namedtuple('Junction',
            ['index',        #  int
             'lat', 'lon',   #  floats: latitude/longitude
             'links',        #  list of Link
            ])
```

8. הסבירו בדו"ח איזו פונקציה יוריסטית בחרתם ומדוע היוריסטיקה אכן קבילה.

אנו בחרנו בתור פונקציה יוריסטית את הפונקציה הבאה- פונקציה שמחשבת זמן נסיעה משוערך מצומת אחת לצומת המטרה ע״י חישוב המרחק בין מיקומי הצמתים מבחינת קו אורך וקו רוחב (ללא התחשבות במרחק הכביש עצמו ובעיקולים האפשריים בו) וחלוקה במהירות הנסיעה המקסימלית (110) מבין כל הכבישים במפה שלנו (אנו עובדים עם יחידות של קמ״ש וק״מ ולכן נקבל זמן נסיעה בשעות). בחרנו את פונקציית היוריסטיקה הנ״ל מכיוון שברוב המקרים הכבישים בין 2 צמתים הינם די ישרים ודי קרובים למרחקים המדויקים בין 2 צמתים, ולכן זו פונקציית יוריסטיקה שמעריכה באופן די טוב את זמן הנסיעה המשוערך (ככל ששני הצמתים יותר רחוקים ויש ביניהם יותר כבישים זמן הנסיעה בפועל יהיה יותר ויותר גדול מזמן הנסיעה המשוערך באמצעות פונקציה זו), וזה בעצם מה שאנחנו צריכים מפונקציית היוריסטיקה- שתעריך באופן הכי טוב את זמן הנסיעה שעוד נותר לנו כדי למצוא מסלול אופטימלי. כמו כן פונקציית היוריסטיקה שלנו קבילה וזה עוד סיבה לכך שבחרנו בפונקציה זו מכיוון שזה מאפשר לנו אופטימליות ופיתוח פחות בנים בכל צעד באלגוריתם. נוכיח כי אכן פונקציית היוריסטיקה שלנו קבילה, כלומר לכל צומת t מתקיים $0 \leq h(t) \leq h^*(t)$ כאשר h זו הפונקציה היוריסטית שלנו ו-h^* זו הפונקציה היוריסטית המושלמת, כלומר מחזירה לנו את זמן הנסיעה האמיתי והמדויק לצומת המטרה במסלול האופטימלי. תהי צומת t כלשהי במפה שלנו, נראה שאכן מתקיים $0 \leq h(t)$: מכיוון ש-$h(t)$ מחושב ע״י מרחק בין מיקומי 2 הצמתים (המרחק המינימלי הוא 0 כאשר שני הצמתים הם אותה צומת עם אותו המיקום) חלקי מהירות הנסיעה המקסימלית מבין כל הכבישים במפה (110), ואכן נקבל כי זמן הנסיעה המינימלי הוא כאשר המרחק בין 2 הצמתים הוא המינימלי (מכיוון שהמכנה שלנו הוא קבוע) ונקבל כי זמן זה אכן שווה ל0 (לא קיים מרחק שלילי בין 2 צמתים) כפי שרצינו ולכן $0 \leq h(t)$.

נשים לב כי זמן הנסיעה האמיתי והמדויק מצומת t לצומת המטרה מחושב ע״י כמה וכמה חישובי זמני נסיעה בפועל בין 2 צמתים שכנים שחישוב זה מורכב מאורך הכביש בין 2 צמתים אלו בפועל (בק״מ) חלקי מהירות הנסיעה המקסימלית בכביש זה (בקמ״ש) וכך נקבל את זמן הנסיעה בפועל בכביש זה. נשים לב בנוסף כי אם אנו רוצים כביש באורך מינימלי בין צומת t לבין צומת המטרה אנו צריכים כביש שהוא בדיוק קו ישר בין 2 הצמתים (לפי אי שוויון המשולש בהכרח המרחק בין 2 הצמתים הוא יותר נמוך מכל דרך אחרת בין 2 צמתים אלו) כלומר המרחק שאנו מחשבים בפונקציה היוריסטית שלנו ולכן בהכרח המרחק בפועל של המסלול האופטימלי נסמנו בw (כך ש-$w = \sum_{i=1}^{n} w_i$ כלומר סכום אורכי הכבישים בדרך) הינו גדול או שווה למרחק בין שני הצמתים (מבחינת קווי אורך ורוחב על מפת כדור הארץ), נסמנו בd. כמו כן נשים לב כי המהירות בפועל במסלול האופטימלי בכל כביש (נסמנה ב-v_i, לכל כביש i בין 2 צמתים במסלול האופטימלי) הינה קטנה או שווה למהירות איתה אנו משתמשים בפונקציה היוריסטית שהיא המהירות המקסימלית מבין כל הכבישים במפה (110). ולכן סה״כ נקבל כי-

$$h(t) = \frac{d}{110}$$

$$h^*(t) = \sum_{i=1}^{n} \frac{w_i}{v_i}$$

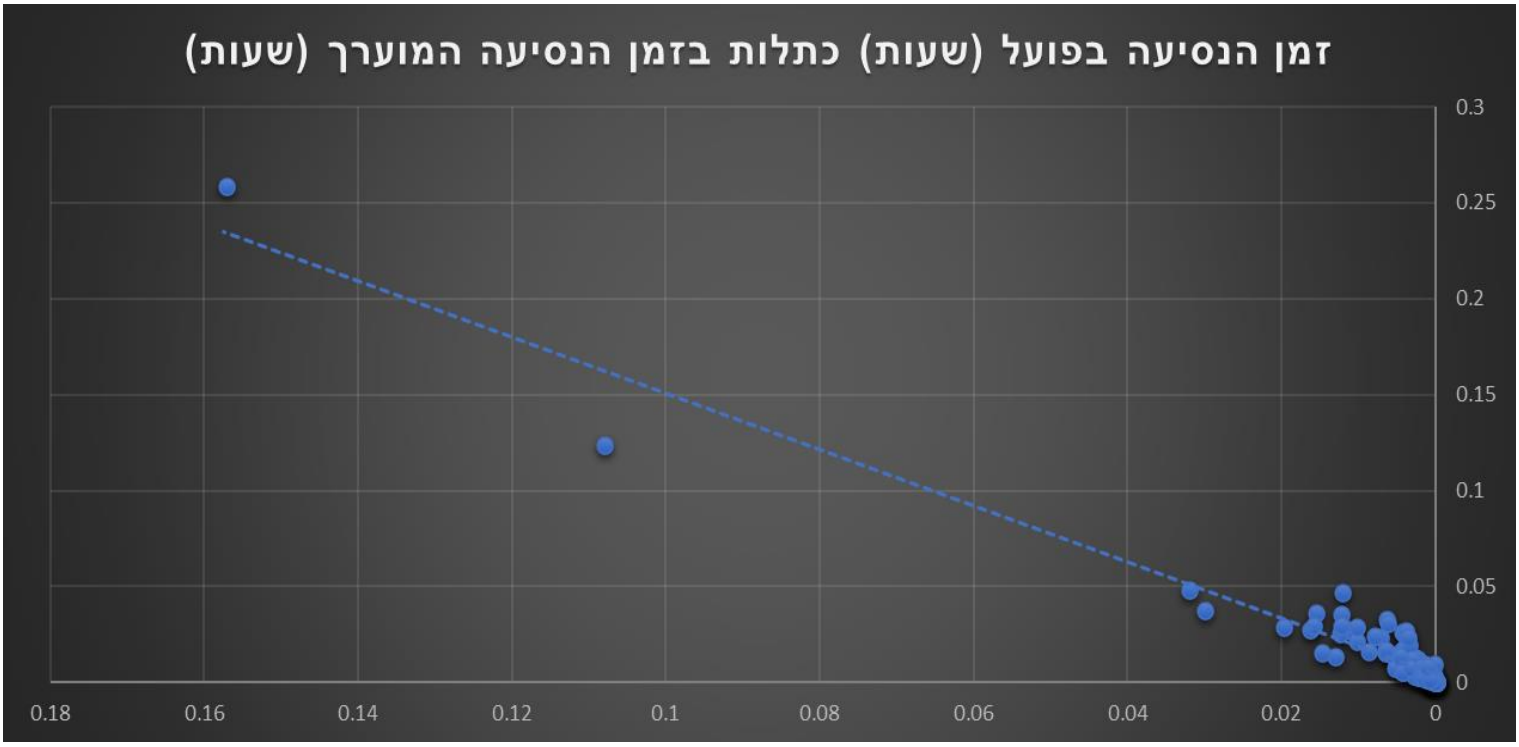
$$h(t) = \frac{d}{110} \leq \frac{1}{110}\sum_{i=1}^{n} w_i \leq \sum_{i=1}^{n} \frac{w_i}{110} \leq \sum_{i=1}^{n} \frac{w_i}{v_i} = h^*(t)$$

והוכחנו כי לכל צומת t אכן מתקיים $0 \leq h(t) \leq h^*(t)$ ולכן הפונקציה היוריסטית שלנו אכן קבילה כדרוש.

9. השתמשו באותן 100 בעיות חיפוש שיצרתם בסעיף 3 והריצו עליהן חיפוש A^*. עבור כל אחת מהבעיות, פלטו לקובץ *results/AStarRuns.txt* את מסלול הנסיעה (כולל קצוות), זמניהם והזמן המשוערך ע"י היוריסטיקה מהמוצא ליעד.

(כל בעית חיפוש בשורה כאשר רשימת הצמתים תחילה ולאחריהם ' - ' זמן הנסיעה ' - ' הזמן המשוערך.)

הציגו בדו"ח גרף ובו נקודה לכל אחת מההרצות הנ"ל. (ציר ה X מייצג את זמן הנסיעה היוריסטי, וציר ה Y מייצג את זמן הנסיעה בפועל) מה ניתן ללמוד מהגרף על הקשר בין המשתנים?



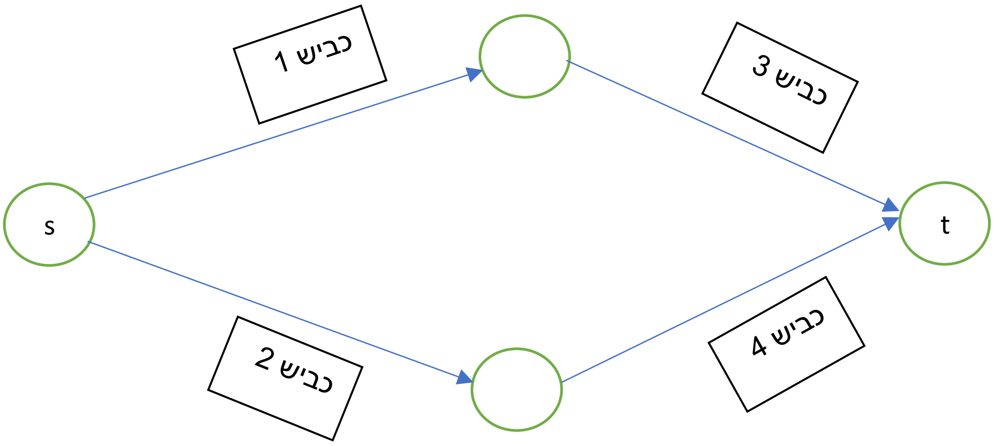
תשובה-

נשים לב כי העברנו קו מגמה וכי אכן יש תלות בין המשתנים- ככל שזמן הנסיעה המוערך יותר גדול כך גם גדל זמן הנסיעה בפועל, כמו כן נשים לב כי בגרף ניתן לראות כי אכן מתקיים שהיוריסטיקה קבילה מכיוון שעבור הערך של היוריסטיקה (הזמן המוערך) בהכרח זמן אותה הנסיעה בפועל הינו יותר גדול. מטיב הקשר ביניהם ניתן לראות גם את טיב היוריסטיקה ושאכן היא יודע לנבא באיזה מסלול זמן הנסיעה יהיה נמוך יותר גם בפועל ובכך מאפשרת לאלגוריתם A^* לבחור בסיכוי גבוה כל פעם את הקודקוד שהעלות הכוללת שלו (מה שעבר והיוריסטיקה-מה שעוד נותר לו) הינה הנמוכה ביותר כלומר את המסלול הקצר ביותר. נשים לב כמו כן כי רוב הבעיות שהגרלנו הינן בעלות עלות די דומה (זמן נסיעה מוערך 40 בערך, וזמן הנסיעה בפועל 80 בערך) וישנן כמה בעיות בודדות שזמן הנסיעה (המוערך ובפועל) יותר גדול משמעותית.

10. בהנחה שהרצתם את אלגוריתם A^* בדיוק לפני שהעומס בכביש התחיל, האם בהכרח המסלול המתקבל יהיה אופטימלי? הסבירו את קביעתכם. (התשובה לשאלה זו אינה תכנותית)

תשובה-

נשים לב כי במידה והרצנו את אלגוריתם A^* בדיוק לפני שהעומס בכביש התחיל אזי אנו יכולים לקבל מסלול שאינו אופטימלי (כלומר האלגוריתם יחזיר מסלול שהוא מגדיר אופטימלי אך הוא אינו אופטימלי) מכיוון שבחישוב העלות (זמן הנסיעה) שלנו אחנו מתחשבים במרחק בין 2 הצמתים (שאנחנו נוסעים מאחת לשנייה) ובמהירות המקסימלית לסוג הכביש אך לא מתחשבים באופציה שיתכן ויהיו פקקים (כמו כן גם אם היו מתחשבים בפקקים יכול להיות שבאמת מסלול הוא אופטימלי והצעד הנוכחי שאנו מבצעים הוא באמת אופטימלי בנקודת זמן מסויימת אך תוך כדי זמן החישוב זמני הנסיעה שהאלגוריתם התחשב בהם זמנים אלו משתנים ומכיוון שהאלגוריתם שלנו אינו מקבל עדכונים על שינויים תוך כדי פעולתו לכן יתכן ולא יהיה אופטימלי) וזה בעצם ישנה את המהירות שבה אנו נוסעים שהיא תהיה נמוכה כאשר ישנם פקקים מהמהירות המקסימלית לסוג כביש זה ולכן אנו יכולים לבחור מסלול שאמנם המהירות המקסימלית שלו גבוהה והמרחק לא גדול אך יש בכביש זה פקקים רבים ולכן המהירות בפועל קטנה בהרבה מהמהירות המקסימלית בכביש זה ולכן במקרה כזה יכול להיות שיהיה כביש עם אותו מרחק אך עם מהירות מקסימלית (ומהירות בפועל, שכן אין פקקים) נמוכה יותר מהמהירות המקסימלית של הכביש השני, ויותר גבוהה מהמהירות בפועל (עם הפקקים) בכביש השני ולכן יתכן והמסלול האופטימלי בהתחשב מצב הפקקים באותה נקודת זמן הוא דווקא לא המסלול שאותו בחרנו. ניתן דוגמה:



כעת נתאר את העלויות (זמני הנסיעה) שהאלגוריתם מחשב (זמני נסיעה מינימליים בכביש בלי להתחשב בפקקים ותנאי דרך אחרים אלה רק במרחק והמהירות המקסימלית באותו כביש)-

כביש 1: 10 דקות, כביש 2: 15 דקות, כביש 3: 20 דקות, כביש 4: 20 דקות.

ולכן האלגוריתם יבחר את המסלול הבא מs לt: כביש 1 ולאחר מכן כביש 3 ובסה״כ נקבל כי אכן זה זמן נסיעה של 30 דקות לעומת 35 דקות במסלול השני האפשרי.

ונתאר את זמני הנסיעה בפועל כולל התחשבות בפקקים באותה נקודת זמן-

כביש 1: 20 דקות, כביש 2: 15 דקות, כביש 3: 20 דקות, כביש 4: 20 דקות.

ונקבל כי המסלול האופטימלי עם פקקים מs לt הינו דווקא כביש 2 ולאחר מכן כביש 4 עם זמן נסיעה של 35 דקות לעומת 30 דקות במסלול השני ואכן נקבל כי האלגוריתם שלנו טעה והמסלול שהחזיר אינו אופטימלי.

13. לכל אחד מהאלגורתמים (*UCS, A*, IDA*), כתבו את זמן ריצת האלגוריתם הממוצע (ללא טעינת המפה). מי מהאלגוריתם רץ בזמן הקצר ביותר? מדוע?

תשובה-

זמן ריצה ממוצע לכל אלגוריתם על 10 הדוגמאות הבאות-

261410,261419

806087,806097

122393,122413

369012,369018

59457,59451

291494,291498

292890,292895

757669,757675

459731,459739

546072,546078

עבור UCS זמן הריצה הממוצע הוא 0.00014617999999773713.

עבור *A זמן הריצה הממוצע הוא 0.00017290000000097973.

עבור *IDA זמן הריצה הממוצע הוא 0.02394206000000253335.

נציין כי בחרנו דוגמאות אלו מתוך 100 הדוגמאות מכיוון ש *IDA הוא אלגוריתם שעובד על חיפוש בעצים, ולכן מכיוון שרוב הדוגמאות מכילות מעגלים לכן על דוגמאות אלו לוקח זמן רב להריץ אלגוריתם זה. נשים לב כי הדבר המשותף לכלל דוגמאות אלו שהמסלולים הקצרים ביותר והמסלולים של *IDA זהים וסדרתיים (עולים או יורדים) לדוגמה מסלול 7 6 5 4 3 2 1. נשים לב כי במקרה הסדרתי אכן *IDA מסיים את ריצתו יחסית מהר, למרות שיכולים להיות מעגלים גם במסלולים האפשריים שלו, מכיוון שניתן לראות כי במאגר הנתונים Israel.csv הצומת השכן הראשון של צומת כלשהו הוא לרוב הסדרתי הבא בתור או הקודם לו, ולכן מכיוון ש *IDA עובר בdfs החל מהשכן הראשון בכל פעם אכן נגיע די מהר (כאשר נקבל כי פונקציית העלות המקסימלית f_limit אכן כוללת את פונקציית העלות של צומת המטרה). בכל מקרה גם מקרה זה שהוא המתאים והאידיאלי ביותר ל *IDA הזמן הממוצע הוא הגרוע ביותר מכיוון שלקחנו 10 דוגמאות שהן לא עם מסלולים יותר מידי ארוכים ולכן הקבועים פה בזמן הריצה משחקים תפקיד יותר משמעותי- אנו מבצעים כמה וכמה פעמים dfs עד לגבול כלשהו של פונקציית העלות וזה יוצא לנו יותר יקר מביצוע UCS או *A (כל אחד מהם מבצע את אלגוריתם החיפוש ללא חזרה לקודקוד ההתחלה, אמנם לא בdfs אך יותר מהיר במקרה שלנו על אף יעילותו של dfs עבור הדוגמאות הנ״ל). נציין כי *IDA הוא לא יעיל עבור עצים שהם יותר ״עמוקים״ כלומר מעט שכנים לכל קודקוד כמו במקרה שלנו עם הכבישים יש לכל קודקוד עד 10 שכנים ולכן *IDA לא יהיה יעיל במקרה זה.

כעת נוסיף את הזמן הכולל שלקח לפתור את כל 100 בעיות החיפוש שהגרלנו-

עבור UCS זמן הריצה הכולל הוא 4.117501799999999.

עבור *A זמן הריצה הכולל הוא 0.3496065000000015.

ונקבל למעשה כי עבור 10 הדוגמאות האלגוריתם שרץ בממוצע בזמן הטוב ביותר הוא דווקא UCS במעט, אך עבור כל 100 בעיות החיפוש שהן הרבה יותר מסובכות ברובן ולא בהכרח סדרתיות אזי *A יותר מהיר פי 10 בערך. ונקבל כי עבור בעיות רנדומליות בסיבוכיותן (סדרתי או לא) ובאורכן (מבחינת מספר קודקודים במסלול האופטימלי) אכן *A הוא בעל זמן הריצה הקצר ביותר. כמו כן גם ציינו בהרצאה שאם פונקציית היוריסטיקה שלנו הינה קונסיסטנטית אזי *A מבטיח לנו פיתוח הכי מעט בנים בכל צעד ומכיוון שפונקציית היוריסטיקה שלנו הינה קבילה (אמנם לא הוכחנו שהיא קונסיסטנטית, אך היא כן קונסיסטנטיות), כך הוכחנו בסעיף 8, על בעיות גדולות יותר *A הוא מהיר בהרבה יותר מUCS מכיוון שמפתח פחות בנים בכל פיתוח וב10 הדוגמאות הפשוטות שלנו מכיוון שהן יחסית קצרות ופשוטות לכן *A וUCS די דומים בזמנים כי אין הרבה בנים לפתח ולכן ההבדל בין מספר הבנים שכל אחד מפתח הוא קטן מאוד ולא מתבטא בזמן.