מבוא לבינה מלאכותית - 236501

תרגיל בית 1

<u>מרחבי חיפוש</u>

מגישים: רות גוטקוביץ ת.ז 209287085

ניקיטה ליסוקון ת.ז 332684190

מטרות התרגיל

- נתמודד עם בעיות פרקטיות ותיאורטיות של חיפוש במרחבי מצבים.
- נתרגל את הנלמד בהרצאות ובתרגולים.
- נתנסה בתכנות ב-python לפתרון בעיות פרקטיות.

הנחיות כלליות

- תאריך הגשה: מוצאי שבת, 1.12.2022, בשעה 23:59.
- את המטלה יש להגיש בזוגות בלבד.
- יש להגיש מטלות מוקלדות בלבד. פתרונות בכתב יד לא ייבדקו.
- ניתן לשלוח שאלות בנוגע לתרגיל בפיאצה בלבד.
- המתרגלת האחראית על תרגיל: **טל חקלאי**.
- בקשות דחיה מוצדקות (מילואים, אשפוז וכו') יש לשלוח למתרגל האחראי (ספיר טובול) בלבד.
- במהלך התרגיל ייתכן שנעלה עדכונים, למסמך הנ"ל – תפורסם הודעה בהתאם.
- העדכונים הינם מחייבים, ועליכם להתעדכן עד מועד הגשת התרגיל.
- שימו לב, התרגיל מהווה כ- 10% מהציון הסופי במקצוע ולכן העתקות תטופלנה בחומרה!
- ציון המטלה יורכב מהגורמים הבאים:
  - 60% - המסמך היבש. מעבר לתשובות הנכונות, אתם נבחנים גם על הצגת הנתונים והתוצאות בצורה קריאה ומסודרת במקומות בהם התבקשתם לכך. הניקוד המפורט בסעיפים של מסמך זה הינו מתוך הציון היבש בלבד.
  - 40% - הקוד המוגש. הקוד שלכם ייבדק באופן מקיף ע״י טסטים. הטסטים יבדקו את התוצאות שלכם לעומת התוצאות המתקבלות במימוש שלנו. אנו מצפים שתקבלו את אותם הערכים בדיוק. נבדוק בין היתר את המסלול המתקבל, את עלותו ואת מס׳ הפיתוחים. לכן עליכם להיצמד להוראות בתרגיל זה. הבדיקות יהיו כמובן מוגבלות בזמן ריצה. יינתן לכם זמן סביר ביותר להרצת כל טסט. אם תעקבו אחר ההוראות במסמך זה ובקוד אין סיבה שלא תעמדו בזמנים אלו. בנוסף, יש להקפיד על הגשת קוד מסודר בהתאם להנחיות. יש לכתוב הערות במקומות חשובים בקוד כדי שיהיה קריא וקל לבדיקה ידנית.
- אנו יודעים שעבור חלקכם זו התנסות ראשונה בכתיבת קוד בפיתון ותרגיל זה מתוכנן בהתאם לכך.
- שימו לב שלא יענו שאלות בסגנון: ״איך מוצאים את עלות הפתרון שהוחזר?״ / ״איך ניגשים למפות הכבישים מתוך המימוש של הפונק׳ ההיא?״ / ״באיזה שדה שמור ה...?״ וכדומה.
- אנחנו רוצים לעודד אתכם לעיין בקוד ולמצוא פרטים אלו בכוחות עצמכם. הכרת סביבת העבודה שסיפקנו לכם והתמצאות בה הן למעשה חלק מהתרגיל.
- בתרגילי הבית בקורס הרצת הניסויים עשויה לקחת זמן רב. לכן מומלץ מאוד להימנע מדחיית העבודה על התרגיל ו/או כתיבת הדו״ח לרגע האחרון. לא תינתנה דחיות על רקע זה.
- מסמך זה כתוב בלשון זכר מטעמי נוחות בלבד, אך מתייחס לנשים וגברים כאחד.

אנחנו קשובים לפניות שלכם במהלך התרגיל ומעדכנים את המסמך הזה בהתאם. גרסאות עדכניות של המסמך יועלו לאתר. הבהרות ועדכונים שנוספים אחרי הפרסום הראשוני יסומנו כאן בצהוב. בנוסף, לכל עדכון יהיה מספר גרסה כדי שתוכלו לעקוב. ייתכן שתפורסמנה גרסאות רבות – אל תיבהלו מכך. השינויים בכל גרסה יכולים להיות קטנים.

חלק א' – מבוא

במטלה זו נעסוק בהפעלת אלגוריתמי חיפוש על מרחבי מצבים לבעיות ניווט. מומלץ לחזור על שקפי ההרצאות והתרגולים הרלוונטיים לפני תחילת העבודה על התרגיל.

במהלך התרגיל תתבקשו להריץ מספר ניסויים ולדווח על תוצאותיהם. אתם נדרשים לבצע ניתוח של התוצאות, כפי שיוסבר בהמשך.

סיפור מסגרת

ריק ומורטי יצאו לעוד אחת מההרפתקאות שלהם והפעם ריק לקח את מורטי לסיור בבר הגאזורפאזור בכוכב הלכת 9-טאוב . לאחר שריק הופך למלפפון חמוץ ונקלע לקטטה עם יצור מזן בלארפ הם בורחים מחוץ לבר. ריק מתכוון להשתמש באקדח הפורטל שלו כדי לחזור הביתה (אקדח שפותח שער ירוק שדרכו אפשר להשתגר למקומות שונים), אבל הוא מגלה שאזל לו דלק אקדחי הפורטל. מורטי זוכר שיש מאגר דלק שנמצא בקצהו של האגם הקפוא, הבעיה היא שצריך לחצות את האגם. והוא מלא בחורים (Holes, not Guys).

למזלם של ריק ומורטי אתם לוקחים הסמסטר את הקורס ״מבוא לבינה מלאכותית״. הם מבקשים מכם לעזור להם לתכנן את המסלול הטוב ביותר אל מאגר הדלק.

חלק ב' – מתחילים לתכנת (40 נקודות)

משימה – רטוב

לפני שמתחילים בבקשה צפו בסרטון הזה.

כעת נעבור לחלק הרטוב של התרגיל. אנו נעבוד בסביבה שבנינו לתרגיל זה על בסיס הסביבה Frozen-Lake שפותחה ע"י OpenAI.

את החלק הרטוב אתם צריכים לפתור במחברת hw1.ipynb-236501. אנחנו ממליצים לעבוד ב-Google Colab. כדי לעשות זאת עליכם להעלות את תוכן התיקייה של התרגיל לתוך תיקייה ב-riveD האישי שלכם. לאחר מכן פתחו את המחברת דרך Google olabC ופעלו לפי ההוראות.

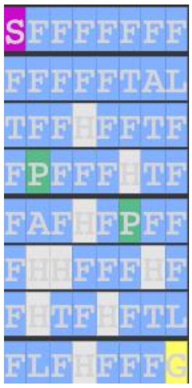
מומלץ לעבור על הקוד במחברת במקביל לקובץ הנוכחי וככה שלב שלב לענות על השאלות השונות.

חלק ג' – שאלות יבשות על הרטוב (**48** נקודות)

התחילו לענות על חלק זה רק לאחר שהבנתם את סביבת העבודה.

שאלה 1 – מבוא (8 נק׳):

השאלות בחלק זה מתבססות על הלוח "8x8" שמופיע במחברת אלא אם נכתב אחרת:



1. **רטוב**: עברו על המחברת עד שאתם מגיעים לחלק של G-BFS ועיצרו שם.
2. יבש (1 נק׳): תחילה נרצה להגדיר את מרחב החיפוש כפי שנלמד בתרגול. הגדר את $(S, 0, I, G)$ עבור סביבת האגם הקפואה. מה גודל מרחב המצבים S? הסבירו.

S- מרחב המצבים יוגדר כמיקום ריק ומורטי באגם, סה"כ יהיו מספר המצבים כגודל הלוח שמייצג אגם (8*8).

O-קבוצת אופרטורים\פעולות תוגדר כצעדים שניתן לעשות באגם, שזה למעלה,למטה,ימינה,שמאלה. סהכ 4 אופרטורים

I-מצב התחלתי יסומן כמיקום התחלתי של ריק ומורטי באגם

G- קבוצת מצבים סופיים תורכב ממצב יחיד שבו ריק ומורטי נמצאים על הלוח ביעד שרצו להגיע אליו

3. יבש (1 נק׳): מה תחזיר לנו הפונקציה Domain על אופרטור 2 (UP)?

פונקציית דומיין אמורה להחזיר את כל המצבים שעליהם ניתן להפעיל אופרטור זה. ניתן להפעיל פעולת UP בכל מצב אם אין בצד מלמעלה קצה האגם\לוח וגם אין חור.

4. יבש (1 נק׳): מה תחזיר לנו הפונקציה Succ על המצב ההתחלתי 0?

היא תחזיר מצבים שאליהם ניתן להגיע תוך צעד אחד ממצב ההתחלתי. ניתן לפנות ימינה או למטה בהינתן שמצב התחלתי הוא בקצה עליון השמאלי של הלוח. לכן פונקציה תחזיר מצבים 1 ו-8.

5. יבש (1 נק׳): האם קיימים מעגלים במרחב החיפוש שלנו?

כן, ניתן לחזור למיקום ממנו חזרנו ע"י אופרטור הפוך לזה שהופעל קודם וכך ליצור מעגלים

6. יבש (1 נק׳): מה הוא מקדם הסיעוף בבעיה?

מכל מצב אפשר להגיע לעד 4 מצבים אחרים שונים, לכן מקדם הסיעוף הוא 4

7. יבש (1 נק׳): במקרה הגרוע ביותר, כמה פעולות ידרשו לסוכן random (כפי שממומש במחברת) להגיע למצב הסופי?

במקרה הגרוע סוכן יכנס למעגל ולא יגיע למצב סופי, לכן ידרשו אינסוף צעדים\פעולות.

8. יבש (1 נק׳): במקרה הטוב ביותר, כמה פעולות ידרשו לסוכן random (כפי שממומש במחברת) להגיע למצב הסופי?

במקרה הטוב סוכן יגיע למצב סופי במסלול הכי קצר דרך פורטל וידרשו 9 צעדים

9. יבש (1 נק׳): עבור לוח כללי, כאשר המצב ההתחלתי והסופי נמצאים בקצוות הלוח (בדומה ללוח "8X8"), האם סוכן שמחפש את המסלול הקצר ביותר יחליט תמיד לעבור דרך ה-portal?

לא, עבור מצב שקצה השני של פורטל יהיה מוקף בחורים שאין ממנו יציאה, ובטח שזה לא מסלול קצר ביותר למצב סופי. אז במקרה כללי לא כדאי לסוכן תמיד להיכנס לפורטל.

שאלה 2 – BFS-G (7 נק׳):

השאלות בחלק זה מתבססות על הלוח "8x8" שמופיע במחברת אלא אם נכתב אחרת.

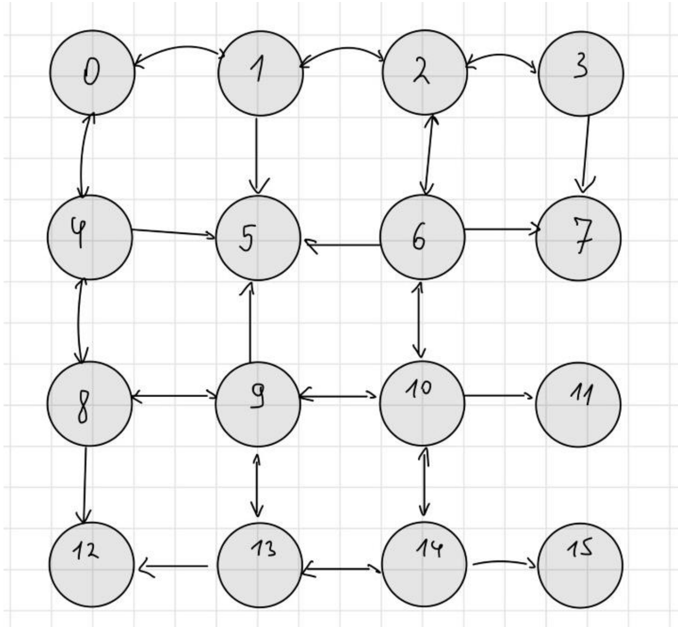
1. **רטוב**: ממשו את אלג׳ BFS-G (על גרף) במחברת ע״פ ההנחיות המופיעות שם.
2. יבש (1 נק׳): מה צריך להיות התנאי על גרף החיפוש (לא בהכרח בבעיית האגם הקפוא) כך שBFS על גרף ו-BFS על עץ ייצרו ויפתחו צמתים זהים באותו הסדר?

במקרה הכללי, גרף לא אמור לחזור על אותם צמתים בחיפוש, אבל עץ יכול לעשות זאת.

אפשר להגביל את עץ לכך שלא יפתח נודים שונים שמייצגים אותו מצב

3. יבש (2 נק׳): עבור הלוח "4x4" שמופיע במחברת, ציירו את גרף המצבים.

```
"4x4":      ["SFFF",
             "FHFH",
             "FFFH",
             "HFFG"],
```



בגרף הנ"ל כל מצב ממוספר מ0 עד 15, כאשר מצב 0 הוא מצב התחלתי ומצב 15 הוא מצב סופי. לכל קרחון ניתן להגיע מחור אחר ובחזרה אבל לחור אפשר רק להגיע ואי אפשר לצאת ממנו, לכן מסלולים לחורים מסומנים בחצים חד צדיים.

4. יבש (2 נק׳): הציעו דרך להשתמש באלגוריתם BFS-G כך שיחזיר פתרון אופטימלי (עלות מינימלית) ללוח "4x4". הסבירו.
   - רמז: עליכם לספק פונקציה $T: G \rightarrow G'$ המקבלת את גרף המצבים G ויוצרת גרף חדש G' ובעזרתה למצוא את המסלול האופטימלי בגרף G.

   ניתן להפעיל אלגוריתם BFS על גרף G ולמצוא מסלול כלשהו למצב סופי ולחשב עלות שלו.

   פונ' T תמחק אחת מהקשתות באופן אקראי במסלול שמצאנו מהגרף, ועל גרף החדש 'G נפעיל אלגוריתם BFS ובכך נכריח אלגוריתם למצוא מסלול אחר למצב סופי. נחזור על תהליך עם הפעלת פונ' T על G וגרפים 'G הנגזרים ממנה עד שBFS לא ימצא מסלול למצב סופי.

   בסוף נשוואה בין עלויות של כל המסלולים ונבחר מסלול הזול ביותר.

5. יבש (2 נק׳): נתון לוח בגודל NxN, ללא חורים, ללא Portals, המכיל N-2 משבצות רגילות (F,T,A,L) ושתי משבצות של מצב סופי ומצב התחלתי. כמה צמתים <u>יפותחו</u> <u>וייווצרו</u> במהלך חיפוש BFS-G? הסבירו?

אם אין פורטלים וחורים, חיפוש BFS יתנהל כך שבכל פעם שנרד בעומק החיפוש, נבדוק אלכסון חדש בגרף למשל באיטרציה ראשונה זה מצב 0, איטריציה שנייה זה מצב 2 ו-N וכו'.

בשיטת פיתוח כזו נגיע למצב סופי רק באלכסון האחרון, לכן ניצור כל ה N*N צמתים ונפתח N*N-2 צמתים (צומת מטרה ואחד לפניו לא נפתח)

*

שאלה 3 – DFS-G (6 נק׳):

1. **רטוב**: ממשו את אלג׳ DFS-G.
2. יבש (1 נק׳): עבור בעיית האגם הקפוא עם לוח NxN, האם האלגוריתם שלם? האם הוא קביל?

אלגוריתם שלם כי לוח סופי, בשלב מסוים חייבים להגיע למצב סופי, אם הוא לא מוקף בחורים

אלגוריתם אינו כביל כי הוא יפלוט מסלול הראשון שימצא למצב סופי, והוא לא בהכרח אופטימלי כי אולי קיים מסלול אחר עם פורטלים או שמכיל יותר משבצות מסוגי L,A,T שמשמעותית עולה פחות ולא הגענו אליו בבדיקה.

3. יבש (1 נק׳): האם אלגוריתם DFS (על עץ), עבור בעיית האגם הקפוא על לוח NxN, היה מוצא פתרון כלשהו? אם כן, מה המסלול שיתקבל? אם לא, כיצד האלגוריתם היה פועל?

כן, הוא בהכרח ימצא פתרון כי אפילו עבור חזרה על אותם צמתים בנודים שונים העץ יהיה סופי ואלגוריתם DFS יוכל לעבור על כולו ולמצוא פתרון

4. יבש (2 נק׳): נתון לוח בגודל NxN, ללא חורים, ללא Portals, המכיל N-2 משבצות רגילות (F,T,A,L) ושתי משבצות של מצב סופי ומצב התחלתי. כמה צמתים <u>יפותחו</u> <u>וייווצרו</u> במהלך חיפוש DFS-G? הסבירו?

יפתח מספר צמתים שייווצרו שווה לסיבוכיות מקום של אלגוריתם DFS כאשר סיעוף הוא 4 ועומק מקסימלי בגרף הוא N. לכן סה"כ 4N-4 צמתים יווצרו וייפתחו 2N-2 צמתים.

5. יבש (2 נק׳): נתון לוח בגודל NxN, ללא חורים, ללא Portals, המכיל N-2 משבצות רגילות (F,T,A,L) ושתי משבצות של מצב סופי ומצב התחלתי. כמה צמתים <u>יפותחו</u> <u>וייווצרו</u> במהלך חיפוש Backtracking-G? הסבירו?

אלגוריתם זה יוצר צמתים רק לפני פיתוח שלהם, לכן יווצרו ויפתחו אותם צמתים, סה"כ 2N-2

שאלה 4 – ID-DFS-G (5 נק׳):

השאלות בחלק זה מתבססות על הלוח "8x8" שמופיע במחברת אלא אם נכתב אחרת.

```
"8x8": [
        "SFFFFFFF",
        "FFFFFTAL",
        "TFFHFFTF",
        "FPFFFHTF",
        "FAFHFPFF",
        "FHHFFFHF",
        "FHTFHFTL",
        "FLFHFFFG",
    ],
```

1. **רטוב**: ממשו את החלקים החסרים של אלג׳ ID-DFS-G בקובץ ע״פ ההנחיות המופיעות שם.
2. יבש (1 נק'): האם האלגוריתם ID-DFS-G שלם?

כן, אלגוריתם מגדיל עומק חיפוש בכל ריצה של DFS-L ובשלב מסוים יגיע לפתרון. במקרה הגרוע נפתח כל הצמתים כמו באלגוריתם BFS.

3. יבש (2 נק'): נניח כי עלות כל פעולה היא 1, האלגוריתם ID-DFS-G לא קביל. הסבירו מדוע. בנוסף, הציעו דרך לעדכן את האלגוריתם על מנת שיהיה קביל במקרה הזה.

אלגוריתם אינו קביל, כי מסלול מתפתח לפי חיפוש בירידה DOWN למטה בגרף לעומק הגרף. נגיע למצב סופי בדרך עקיפה ונעבור בפינה שמאלית תחתונה של הגרף

זה לא פתרון הקצר כי אולי קיים פורטל שמקצר דרך למצב סופי, אבל אלגוריתם לא יגיע אליו כי יחזיר פתרון הראשון שימצא

אלגוריתם יהיה קביל רק אם ימשיך לחפש פתרונות גם אחרי שמצא פתרונות אחרים, וישווה בסוף מחירים של כולם ויפלוט את מסלול הזול ביותר.

4. יבש (2 נק׳): הציגו גרף המראה את השפעת L (לפחות 5 ערכים שונים) על מספר הצמתים שמפותחים בכל העמקה. הסבירו בקצרה את הגרף.

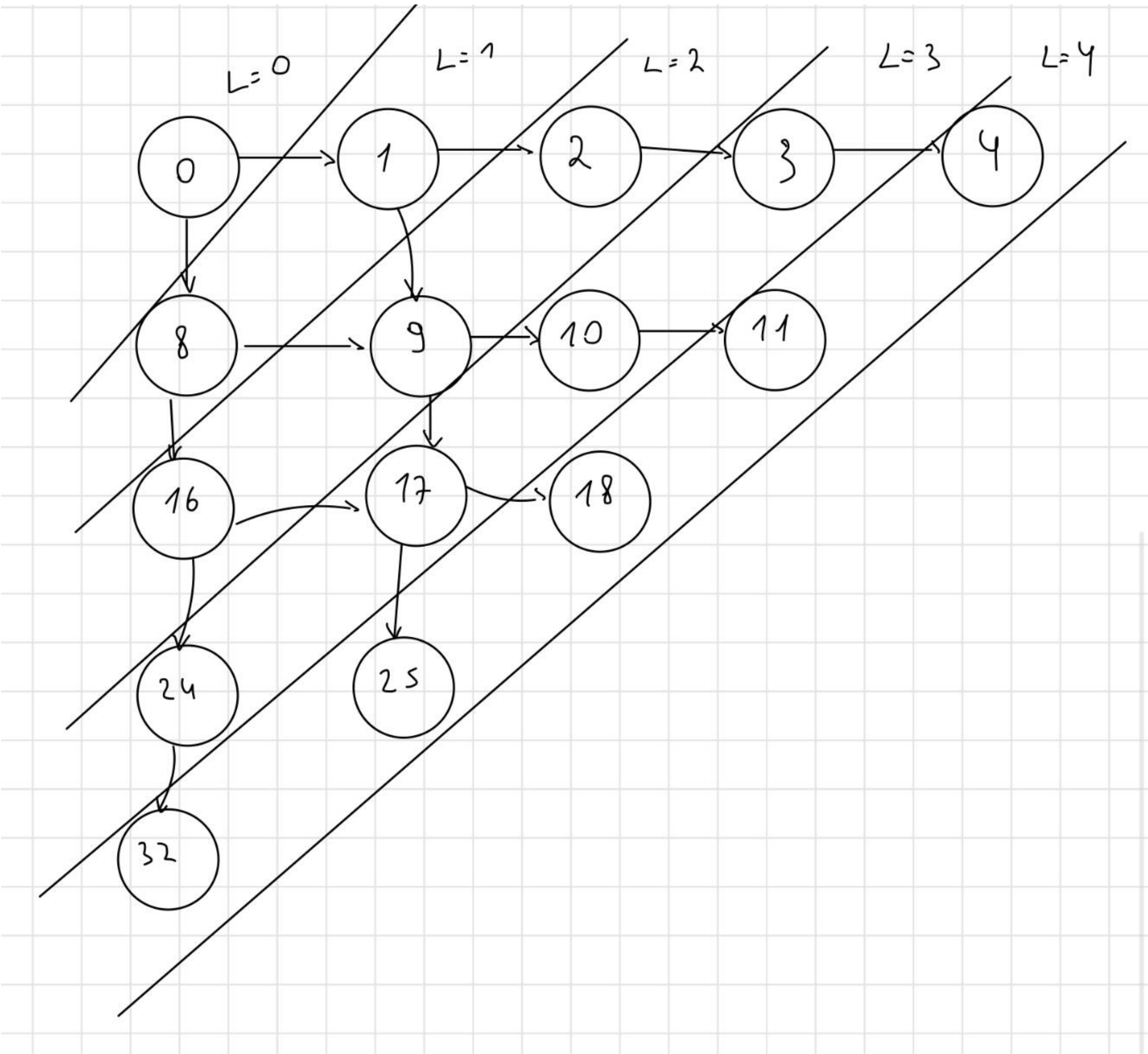
עבור עומק 0 (מצב התחלתי) נפתח 0 רק צומת של מצב התחלתי

עבור עומק 1 נפתח צומת של מצב התחלתי ושני צמתים אליהן הוא יכול להגיע, סה"כ 3 צמתים

עבור עומק 2 נפתח עוד 3 צמתים נוספים ביחס לעומק הקודם, סה"כ 6

עבור עומק 3 נפתח 4 צמתים חדשים ביחס לעומק הקודם, סה"כ 10 צמתים

עבור עומק 4 נפתח 5 צמתים יותר ביחס לעומק הקודם, סה"כ 15 צמתים



שאלה 5 - UCS (4 נק׳):

השאלות בחלק זה מתבססות על הלוח "8x8" שמופיע במחברת אלא אם נכתב אחרת.

1. **רטוב**: ממשו את החלקים החסרים של אלג׳ UCS בקובץ ע״פ ההנחיות המופיעות במחברת.
2. יבש (1 נק׳): עבור אילו בעיות חיפוש אלגוריתם UCS ואלגוריתם BFS יפעלו באותו האופן? הסבירו.

תשובה : שני אלגוריתמים יפעלו באותו האופן ויתקבלו אותן התוצאות כאשר ב-UCS כלל המרחקים בין כל שני צמתים הינם שווים ל-1. במקרה זה תור העדיפויות של UCS יפעל בצורה זהה לתור של BFS ולכן הצמתים של עץ החיפוש יפותחו יפותחו בצורה זהה לכל אלגוריתם.

3. יבש (1 נק׳): האם בבעיית החיפוש שלנו, עבור לוח NxN, האלגוריתם הוא שלם? האם הוא קביל?

תשובה : האלגוריתם שלם בהנחה לפתרון הטוב ביותר יש עלות סופית ובנוסף מחיר קשת מינימלי הינו חיובי. מכיון שמרחב החיפוש הינו סופי אזי אלגוריתם ה-UCS יחזיר תמיד את הפתרון האופטימלי ולכן האלגוריתם הינו קביל ושלם.

4. יבש (2 נק׳): דן טעה במימוש של אלגוריתם UCS ובטעות בדק בעת יצירת הצומת האם היא צומת מטרה במקום בפיתוח שלה. הביאו דוגמה ללוח שעבורו דן יחזיר בכל זאת את המסלול הקל ביותר ודוגמה ללוח שעבורו דן לא יחזיר את המסלול הקל ביותר. הסבירו.

- דוגמה ללוח שעבורו דן יחזיר את המסלול הקל ביותר:
נתבונן בלוח זה – מכיוון שעלות המעבר בין כלל הצמתים הינה שווה (כלל הצמתים מסוג F למעט המעבר לצומת האחרון G) נקבל כי אם הגענו לצומת G לא קיים מסלול זול יותר ולכן שינוי אופן מימוש האלגוריתם כפי שביצע אותו דן לא ישנה את התשובה ולכן האלגוריתם יחזיר את המסלול הקל ביותר.

S	F	F	F	F	F	F	F
F	F	F	F	F	F	F	F
F	F	F	F	F	F	F	F
F	F	F	F	F	F	F	F
F	F	F	F	F	F	F	F
F	F	F	F	F	F	F	F
F	F	F	F	F	F	F	F
F	F	F	F	F	F	F	G

S	H	H	H	H	H	H	H
F	H	H	H	H	H	H	H
F	H	H	H	H	H	H	H
F	H	H	H	H	H	H	H
F	H	H	H	H	H	H	H
F	F	F	F	F	F	F	F
F	F	F	F	F	F	F	T
F	A	A	A	A	A	A	G

- דוגמה ללוח שעבורו דן לא יחזיר את המסלול הקל ביותר:
הסבר: בשלב מסוים במהלך האלגוריתם נגיע לצומת F (המסומן בצבע צהוב). הגדרנו בתרגיל כי במידה וישנם שני מצבים בעלי עלות זהה אזי נבחר את הצומת הבא לפיתוח לפי המצב שבו הם נמצאים כאשר נעדיף לבחור צומת אם מצב קטן יותר. לכן כאשר נגיע לצומת F נבחר להתקדם בשורה שבה הוא נמצא בלוח (המצבים בשורה זו קטנים יותר מהמצבים בשורה שאחרי) . לאחר שנגיע לסוף השורה ונבצע succ לצומת האחרון בשורה זו נוכל להוסיף אך ורק את הצומת T ל-opened. מכיוון שהעלות של T תהיה הנמוכה ביותר כאשר נבצע pop ל- opened (בעת בחירת הצומת הבא לפיתוח) אנו נבחר אותו . כאשר נפתח את T אנו ניצור את הצומת G ומכיוון שדן שינה את אופן מימוש של UCS אנו נחזיר את המסלול שמצאנו (מסומן באפור בלוח) . אבל ישנו מסלול קל יותר (מסומן בלוח בצבע תכלת) – מסלול ה קל יותר מאופן הגדרת העלות בתרגיל.
ולכן האלגוריתם של דן לא יחזיר את המסלול הקל ביותר.

שאלה 6 – יוריסטיקה (2 נק׳):

נגדיר יוריסטיקה חדשה:

$$h_{SAP}(s) = \min\{h_{Manhatan}(s,g), Cost(p)\}, g \in G$$

כאשר הביטוי הראשון הוא מרחק מנהטן מהמצב הנוכחי למצב הסופי והביטוי היא עלות קשת המביאה למשבצת שיגור.

1. יבש (1 נק׳): האם היוריסטיקה קבילה על כל לוח? אם כן הסבר, אם לא הבא קודמה נגדית.

נוכיח כי היוריסטיקה הינה קבילה:

מתקיים עבור צומת מטרה:

$$h_{SAP}(g) = \min\{h_{Manhatan}(g,g), Cost(p)\} = \min\{0, cost(p)\} = 0 = h^*(s)$$

כעת נוכיח כי מתקיים $0 \le h(s) \le h^*(s)$. מכיון שמחשבים מינימום בין מרחק כלשהו לערך כלשהו חיובי נקבל כי היוריסטיקה בהכרח מחזירה ערך שאינו שלילי כלומר מתקיים: $0 \le h(s)$ לכל צומת.

כעת נראה כי : $h(s) \le h^*(s)$. נחלק למקרים:

1. אם g אינו ישיג מצומת ההתחלה אזי מתקיים: $h^*(s) = \infty$ ונקבל כי $h(s) < h^*(s)$.
2. אחרת, אם g ישיג מצומת ההתחלה אזי :

אם מתקיים: $cost(p) > h_{manhatan}(s,g)$

$$h(s) = \min\{h_{manhatan}(s,g), cost(p)\} = cost(p) =$$
$$\leq number\ of\ edges\ in\ best\ path\ to\ gaol * 1$$
$$+ cost(p) \underset{cost(e)\geq 1}{\leq} \sum_{e \in best\ path\ to\ goal} cost(e) + cost(p) = h^*(s)$$

אחרת, נניח כי מתקיים: $cost(p) < h_{manhatan}(s,g)$:

$$h(s) = \min\{h_{manhatan}(s,g), cost(p)\} = h_{manhatan}(s,g) = |g_x - s_x| + |g_y - s_y|$$
$$\leq number\ of\ edges\ in\ best\ path\ to\ gaol * 1 \underset{cost(e)\geq 1}{\leq} \sum_{e \in best\ path\ to\ goal} cost(e)$$
$$= h^*(s)$$

כלומר, קיבלנו כי תמיד מתקיים: $0 \leq h(s) \leq h^*(s)$ ולכן היוריסטיקה הינה קבילה.

2. יבש (1 נק׳): האם היוריסטיקה עקבית על כל לוח? אם כן הסבר, אם לא הבא דוגמה נגדית.

H	S
G	H_1

היוריסטיקה אינה עקבית לכל לוח- ניתן דוגמה נגדית :

נשים לב כי בתרגיל הוגדר כי $Cost(p) = 100$. על מנת להראות כי יוריסטיקה היא עקבית יש להוכיח כי לכל $\forall s \in S, s' \in succ(s): h(s) - h(s') \leq cost(s,s')$

נראה כי קיים צומת s כך שעבורו לא מתקיים התנאי הנ"ל. נבחר צומת s שהינו צומת ההתחלה ונקבל-

$$h_{manhatan}(s,g) = |g_x - s_x| + |g_y - s_y| = |2-1| + |2-1| = 2$$
$$h_{manhatan}(H_1,g) = |g_x - h_x| + |g_y - h_y| = |2-2| + |2-1| = 1$$
$$h(s) = \min\{h_{manhatan}(s,g), cost(p)\} = \min\{2,100\} = 2$$
$$h(H_1) = \min\{h_{manhatan}(H_1,g), cost(p)\} = \min\{1,100\} = 1$$
$$h(s) - h(H_1) = 2 - 1 = 1 \geq cost(s,H_1) = 0$$

ולכן היוריסטיקה אינה עקבית.

שאלה 7 – Greedy Best First Search (3 נק׳):

השאלות בחלק זה מתבססות על הלוח "8x8" שמופיע במחברת אלא אם נכתב אחרת.

1. רטוב: ממשו את החלקים החסרים באלג׳ Greedy Best First Search בקובץ ע״פ ההנחיות המופיעות במחברת. עליכם להשתמש ביוריסטיקה h_{SAP}.
2. יבש (1 נק׳): האם האלגוריתם שלם? האם הוא קביל?

האלגוריתם שלם בהנחה ומדובר במספר מצבים סופי.
מכיוון שלא מובטח כי האלגוריתם יחזיר פתרון אופטימלי אזי הוא אינו קביל.

3. יבש (2 נק׳): תנו יתרון וחיסרון של אלגוריתם Greedy Best first Search לעומת UCS.

יתרון של האלגוריתם Greedy: זהו אלגוריתם מיודע כלומר משתמש במידע שיש לו (היוריסטיקה) על מנת לקבל תוצאה טובה יותר.
חסרון של האלגוריתם Greedy : זהו אינו אלגוריתם קביל כלומר הוא יכול להחזיר תוצאות לא אופטימליות בעוד ש-UCS הינו אלגוריתם קביל.

שאלה 8 – W-A* (7 נק׳):

השאלות בחלק זה מתבססות על הלוח "8x8" שמופיע במחברת.

1. רטוב: ממשו את החלקים החסרים באלג׳ W-A* בקובץ ע״פ ההנחיות המופיעות שם. עליכם להשתמש ביוריסטיקה h_{SAP}.
2. יבש (4 נק׳): לפניכם מספר יוריסטיקות, הוכיחו או הפריכו בעזרת דוגמאת נגדית את קבילותן של היוריסטיקות:
   - $GreedyHeuristic(s) = 0\ if\ s\ is\ goal, else\ 1$
   - $h_{MD}(s) = h_{Manhatan}(s,g), g \in G$
   - $NearestPortalOrGoalHeuristic(s) = \min\{h_{Manhatan}(s,p_1), h_{Manhatan}(s,p_2), h_{Manhatan}(s,g)\}, g \in G$
     כאשר p_1, p_2 הם משבצות שיגור ($p_1 \neq p_2$) ו-g הוא צומת מטרה.
   - $NearestPortalToFinalHeuristic(s) = NearestPortalOrGoalHeuristic(s) + h_{Manhatan}(p_1,p_2)$

- היוריסטיקה $GreedyHeuristic(s) = 0\ if\ s\ is\ goal, else\ 1$ הינה קבילה.

הוכחה- מהגדרת היוריסטיקה נקבל כי $h(goal) = 0$, ולכל צומת s שאינו צומת המטרה מתקיים כי : $h(s) = 1$.

נראה כי מתקיים: $0 \le h(s) \le h^*(s)$.

נשים לב כי מתקיים : $0 \le h(s) = 1$. בנוסף נשים לב כי עבור S שאינו צומת מטרה קיימת לפחות קשת אחת במסלול בינו לבין צומת המטרה ולכן, מאופן הגדרת היוריסטיקה מחיר הקשת הוא לפחות 1, לכן, מחיר המסלול הזול ביותר מ-s לצומת המטרה הוא לפחות 1, ומכך נקבל כי :

$$h(s) = 1 \le cost\ of\ best\ path\ to\ goal \le \underbrace{\min}_{path:s\to\cdots\to goal} \sum_{e\in path} cost(e) = h^*(s)$$

כלומר, לפי הגדרה מתקיים כי היוריסטיקה הינה קבילה.

- היוריסטיקה $h_{MD}(s) = h_{Manhatan}(s,g), g \in G$ הינה קבילה. הוכחה:

על מנת להוכיח כי יוריסטיקה היא קבילה יש להוכיח כי מתקיים $0 \le h(s) \le h^*(s)$.
עבור צומת מטרה מתקיים כי : $h(g) = h_{manhatan}(g,g) = |g_x - g_x| + |g_y - g_y| = 0 = h^*(g)$.
עבור צומת שאינו צומת מטרה מתקיים כי מכיוון שמרחק מנהטן הינו מרחק אזי $0 \le h(s)$. נראה כי מתקיים: $h(s) \le h^*(s)$.
לפי אופן הגדרת התרגיל מרחק קשת מינימלי בין צומת לצומת במסלול קביל (מסלול שאינו כולל holes -הוגדר בתרגיל כי ממצב זה לא ניתן להתקדם לשום מצב אחר בלוח) הינו 1. לכן, נקבל כי

$$h(s) = h_{manhatan}(s,g) = |g_x - s_x| + |g_y - s_y| \le number\ of\ edges\ in\ best\ path\ to\ gaol * 1 \underbrace{\le}_{cost(e)\ge 1} \sum_{e\in best\ path\ to\ goal} cost(e) = h^*(s)$$

- היוריסטיקה $NearestPortalOrGoalHeuristic(s) = \min\{h_{Manhatan}(s,p_1), h_{Manhatan}(s,p_2), h_{Manhatan}(s,g)\}, g \in G$:

נחלק למקרים:

1. במידה ולא קיימים כלל משבצות שיגור בלוח אזי המרחק $h_{Manhatan}(s,p_2)$ אינו מוגדר כלל ונקבל כי $NearestPortalOrGoalHeuristic(s) = \min\{h_{Manhatan}(s,p_1), h_{Manhatan}(s,p_2), h_{Manhatan}(s,g)\}, g \in G = \min\{h_{Manhatan}(s,g)\} = h_{Manhatan}(s,g)$ ואז נקבל את המקרה מהסעיף הקודם – כלומר היוריסטיקה קבילה.

2. נניח כי קיימות משבצות שיגור בלוח: נוכיח כי היוריסטיקה קבילה:

עבור צומת מטרה מתקיים כי $NearestPortalOrGoalHeuristic(s) = \min\{h_{Manhatan}(s,p_1), h_{Manhatan}(s,p_2), h_{manhatan}(g,g)\} = h_{manhatan}(g,g) = |g_x - g_x| + |g_y - g_y| = 0 = h^*(g)$
מכיוון שמחשבים מינימום בין מרחקי מנהטן אזי מתקבל כי $0 \le h(s)$. כעת נראה כי מתקיים: $h(s) \le h^*(s)$.

א. נניח כי $\min\{h_{Manhatan}(s,p_1), h_{Manhatan}(s,p_2), h_{Manhatan}(s,g)\} = h_{Manhatan}(s,p_2)$ אזי

$$h(s) = \min\{h_{Manhatan}(s,p_1), h_{Manhatan}(s,p_2), h_{Manhatan}(s,g)\} = h_{Manhatan}(s,p_2) \le h_{Manhatan}(s,g) \underbrace{\le}_{\text{סעיף קודם}} h^*(s)$$

ב. נניח כי $\min\{h_{Manhatan}(s,p_1), h_{Manhatan}(s,p_2), h_{Manhatan}(s,g)\} = h_{Manhatan}(s,g)$ אזי מתקיים המקרה שבסעיף הקודם ולכן היוריסטיקה קבילה.

- היוריסטיקה $NearestPortalToFinalHeuristic(s) = NearestPortalOrGoalHeuristic(s) + h_{Manhatan}(p_1,p_2)$ אינה קבילה

נראה זאת על ידי דוגמה נגדית:
על מנת להוכיח שהיוריסטיקה קבילה יש להראות כי מתקיים: $0 \le h(s) \le h^*(s)$. נראה כי קיים צומת שאינו מקיים תנאי זה.
נסתכל על צוממת ההתחלה ונקבל כי $h_{Manhatan}(s,p_1) = 7, h_{Manhatan}(s,p_2) = 7, h_{Manhatan}(s,g) = 14$
כלומר: $NearestPortalOrGoalHeuristic(s) = 7$, $h_{Manhatan}(p_2,p_1) = 12$
כלומר מתקיים כי: $h(s) = NearestPortalToFinalHeuristic(s) = NearestPortalOrGoalHeuristic(s) + h_{Manhatan}(p_1,p_2) = 19$
ו- $h^*(s) = \sum_{e\in best\ path\ to\ goal} cost(e) = 14$.
כמו כן נשים לב כי מתקיים : $h(g) = NearestPortalToFinalHeuristic(g) = NearestPortalOrGoalHeuristic(g) + h_{Manhatan}(p_1,p_2) = 12 \ne 0$

ולכן היוריסטיקה אינה קבילה.

S	A	A	A	A	A	A	A
H	H	H	H	H	H	P_2	A
H	H	H	H	H	H	H	A
H	H	H	H	H	H	H	A
H	H	H	H	H	H	H	A
H	H	H	H	H	H	H	A
H	H	H	H	H	H	H	A
P_1	H	H	H	H	H	H	G

3. יבש (נק׳ 1): איזו יוריסטיקה (מבין הקבילות) היא המיודעת ביותר (כולל היוריסטיקה h_{SAP})?

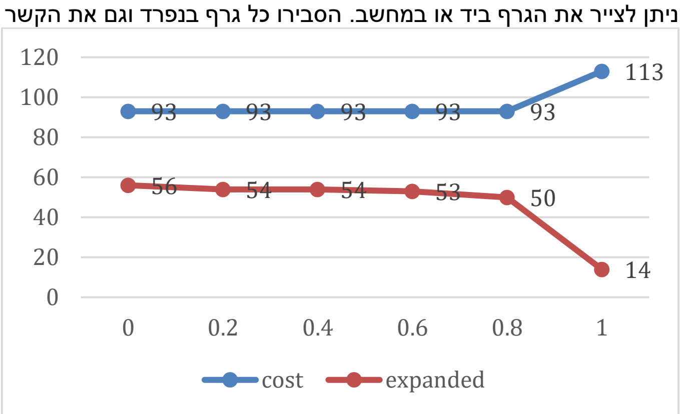
היוריסטיקות הקבילות שקיבלנו הן- $GreedyHeuristic(s), h_{MD}(s), NearestPortalOrGoalHeuristic(s), h_{sap}$
לפי הגדרה יוריסטיקה קבילה h_1 תקרא מיודעת מיוריסטיקה קבילה h_2 אם מתקיים: $\forall s \in S, h_1(s) \geq h_2(s)$
לפי הגדרה זה מתקיים:

$$0 \leq h_{sap}(s) = \min\{h_{Manhatan}(s,g), Cost(p)\} \leq h_{Manhatan}(s,g) = h_{MD}(s) \leq h^*(s)$$
$$0 \leq NearestPortalOrGoalHeuristic(s) \leq h_{Manhatan}(s,g) = h_{MD}(s) \leq h^*(s)$$
$$0 \leq GreedyHeuristic(s) \leq 1 \leq h_{Manhatan}(s,g) = h_{MD}(s) \leq h^*(s)$$

ולכן נקבל כי היוריסטיקה המיודעת ביותר היא היוריסטיקה $h_{MD}(s)$.

4. יבש (נק׳ 2): הריצו את W-A* עם ערכי W שונים והציגו שני גרפים:
   - גרף 1: מספר הפיתוחים כתלות ב-W .
   - גרף 2: עלות הפתרון שנמצא כתלות ב-W.

ניתן לצייר את הגרף ביד או במחשב. הסבירו כל גרף בנפרד וגם את הקשר ביניהם.



ניתן לראות כי ככל שהמשקל עולה כמות ה- expanded כלומר כמות הצמתים שפותחו הולכת וקטנה כלומר אכן היוריסטיקה עוזרת בהקטנת מספר הצמתים שפותחו, אבל ככל שהמשקל עולה אזי עלות הפתרון שנמצא גדלה. כלומר ישנו קשר בין שני המדדים וכאשר מצליחים לשפר את מספר הצמתים שפותחו אזי עלות הפתרון הינה פחות טובה וכאשר עלות הפתרון הינה מינימלית כמות הצמתים שפותחו גבוהה.

שאלה 9 – A*-epsilon (6 נק׳):

1. רטוב: ממשו את החלקים החסרים באלג׳ A*-epsilon בקובץ ע״פ ההנחיות המופיעות שם. עליכם להשתמש ביוריסטיקה h_{SAP} כדי ליצור את FOCAL וב-$g(v)$ כדי לבחור את הצומת הבא לפיתוח מתוך FOCAL.
2. יבש (2 נק׳) תנו יתרון וחיסרון של A*-epsilon לעומת A*.
   יתרון של A*-epsilon: מאפשר למצוא פתרון בצורה מהירה יותר.
   חסרון של A*-epsilon: לא בהכרח מתקבל הפתרון האופטימלי (מבטיח פתרון אופטימלי עד כדי פקטור של ε + 1).
3. **יבש/רטוב (4 נק׳): תנו הצעה ליוריסטיקה כדי לבחור את הצומת הבאה לפיתוח מתוך FOCAL. תארו את היוריסטיקה והציגו השוואה בין השימוש ביוריסטיקה זו לעומת השימוש ב-$g(v)$, מבחינת מספר פיתוחים, מסלול שנבחר ועלות המסלול שנבחר. שימו לב- בקוד שאתם מגישים על הסוכן להשתמש ב-g(v) ולא ביוריסטיקה שבחרתם בסעיף זה.**

נבחר את היוריסטיקה לבחירת צומת מתוך focal – $new_{heuristic}(v) = g(v) + h_{manhatan}(v)$. התוצאות שהתקבלו :
עבור היוריסטיקה הנתונה בתרגיל קיבלנו כי

$$expanded = 54, \quad total_{cost} = 93, \quad actions = [1,1,1,1,1,0,1,1,0,0,0,0,0,0]$$

עבור היוריסטיקה החדשה שהגדרתי בתרגיל זה קיבלתי

$$expanded = 54, \quad total_{cost} = 93, \quad actions = [1,1,1,1,1,0,1,0,0,1,0,0,0,0]$$

חלק ד׳ – שאלות בסגנון מבחן (12 נקודות)

1. וריאציות של A* (6 נק׳)

1. נגדיר n' להיות צומת האב של צומת n בגרף. כמו כן נניח ש- h היא יוריסטיקה קבילה ש<u>אינה</u> יוריסטיקת האפס וכן קיים במרחב מצב מטרה ישיג מהמצב ההתחלתי.
עבור כל אחד מהאלגוריתמים הסבר האם הוא שלם והאם הוא קביל:
   a. A* כפי שנלמד בהרצאה.

אלגוריתם יהיה שלך בהנחה שמחיר הקשתות חסום מלמטה ע"י מספר חיובי, כפי שהוסבר בהרצאה.
ואם תנאי של חסימות מתקיים אלגוריתם זה גם קביל כי יוריסטיקה h קבילה.

   b. A* שמתעלם מערך ה-h.

אם נתעלם מערך h נקבל אלגוריתם uniform cost search .
אלגוריתם זה שלם בהנחה שמחיר הקשתות חסום מלמטה ע"י מספר חיובי
ואם הוא שלם אז בהכרח קביל מהוכחה בהרצאה

   c. A* שמתעלם מערך ה-g.

אם נתעלם מערך g נקבל אלגוריתם greedy best first search.
תנאי לשלמות האלגוריתם הוא שמרחב יהיה קשיר וסופי.
אלגוריתם זה אינו קביל לפי הוכחה מהרצאה

   d. A* כך ש- $f(n) = g(n) + h(n')$

אלגוריתם יהיה שלך בהנחה שמחיר הקשתות חסום מלמטה ע"י מספר חיובי, כפי שהוסבר בהרצאה.
ואם תנאי של חסימות מתקיים, נבדוק אם יוריסטיקה חדשה קבילה:
נסמנה $h_2(n) = h(n')$
$0 <= h(n') = h_2(n) <= h^*(n')$
אבל לא בהכרח מתקיים כי $h^*(n') <= h^*(n)$. צומת הבן יכולה להיות יותר רחוקה ממצב סופי בהשוואה לאב שלה, אם ביצענו פעולה כלשהי שמרחיקה אותנו ממצב מטרה.

לכן יוריסטיקה לא קבילה ואלגוריתם אינו קביל בהכרח

   e. A* כך ש- $f(n) = g(n') + h(n)$

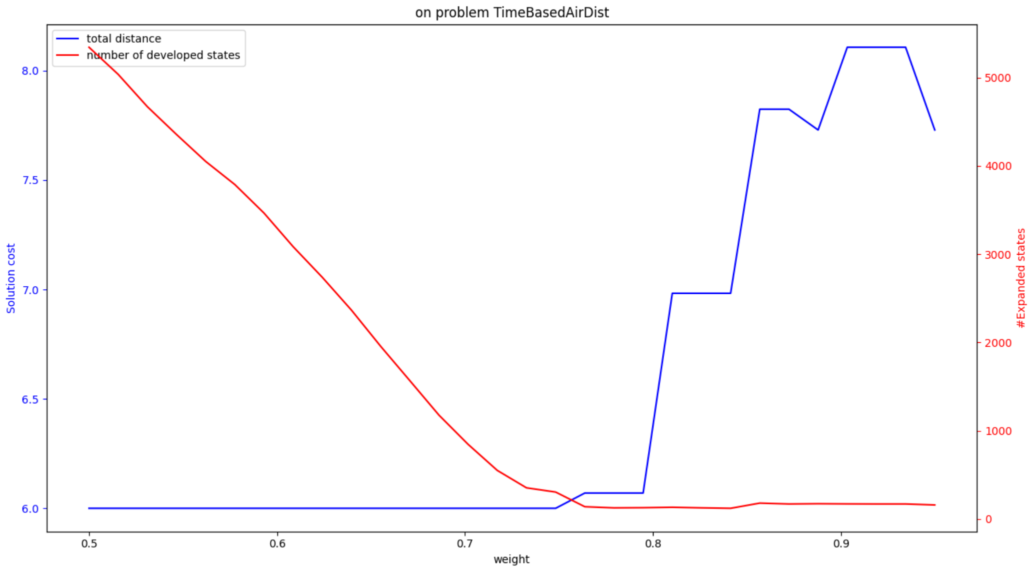
אלגוריתם שלם אם פונקציית מחיר חסומה מלמטה ע"י מספר חיובי, אך במקרה שלנו כל מחיר של קשתות ממצב התחלתי הוא 0 ואין חסם תחתון חיובי על מחיר.

אבל עדיין אם לשאר מחירי קשתות קיים חסם תחתון חיובי, אלגוריתם חייב להיות שלם כי לא נגיע למצב דעיכה לצמתים בעלי מחיר קשתות שואף ל-0 כמו שנלמד בתרגול

אלגוריתם אינו קביל. נניח שיש 2 מסלולים עם מחירי קשתות זהות לגמרה חוץ מ-2 קשתות שונות שיוצאות מצומת מצב התחלתי. עם פונקציית מחיר החדשה ל-2 הקשתות הללו יהיה אותו מחיר ולא נגלה איזה מסלול זול יותר לצומת מטרה.

2. כעת נתון $0.5 \leq w_1 < w_2 \leq 1$. נסמן P_1, P_2 להיות המסלולים המוחזרים מריצת w-A* עם המשקלים w_1, w_2 בהתאמה.
תזכורת לחישוב $f(n) = w * h(n) + (1 - w) * g(n)$. הוכח /הפרך: $cost(P_1) \leq cost(P_2)$.

לא נכון



לפי הגרף, כאשר המשקל קרוב לקצוות הקטע, אנו משלמים יותר בעלות (זמן) הפתרון או במספר הפיתוחים שהאלגוריתם מבצע. ככל שמתקרבים למרכז הקטע מתקבלת תוצאה אופטימלית. (יוריסטיקה היא מרחב אווירי)

ניתן לראות בגרף באופן מגמתי, שכלל האצבע אכן נכון, אך עם זאת אכן יתכנו נקודות שבהן הוא לא מתקיים. ניתן לראות זאת בגרף הכחול כאשר המשקל מוגדר 0.95. ניתן לראות שקיים פתרון עם משקל קטן יותר, אך עלות הפתרון היא המקסימלית המתקבלת מבין כל הפתרונות (כמות הפיתוחים נשארה במגמה אחידה). ובכך ניתן לראות בגרף כי הכלל והדגש שהוזכרו לעיל מתקיימים.

3. הוכח/הפרך: לכל בעיית חיפוש, המסלול שיחזור על ידי אלגוריתם UCS יכול להשתנות אם <u>נוסיף</u> ערך קבוע $C > 0$ לכל עלות קשת במהלך החיפוש.

לא נכון

אלגוריתם USC תמיד מפתח צומת מחוברת בקשת עם מחיר G מינימלי. אם נוסיף ערך חיובי קבוע לעלויות של כל הקשתות, סדר פניות לקשתות לא ישתנה (הסבר של הגיון: אם לכל איברים חיוביים במערך ממוין מוסיפים קבוע חיובי, הוא יישאר ממוין).

אם לא מעדכנים עלות של כל הקשתות מראש, אלא רק כאשר מגיעים אליהן בחיפוש, עדיין כל הקשתות שנמצאות "בחזית" יגדלו באותו ערך בזמן החיפוש של קשת עם עלות הזולה ביותר

אזי, כל גרף עם משקלים ניתן להביא כדוגמה נגדית לטענה.

4. הוכח/הפרך: לכל בעיית חיפוש, המסלול שיחזור על ידי אלגוריתם UCS יכול להשתנות אם <u>נכפיל</u> ערך קבוע $C > 0$ כל עלות קשת במהלך החיפוש.

אלגוריתם USC תמיד מפתח צומת מחוברת בקשת עם מחיר G מינימלי. אם נכפול עלות כל קשת בערך חיובי קבוע, סדר פניות לקשתות לא ישתנה (הסבר של הגיון: אם לכל איברים חיוביים במערך ממוין כופלים בקבוע חיובי, הוא יישאר ממוין).

אם לא מעדכנים עלות של כל הקשתות מראש, אלא רק כאשר מגיעים אליהן בחיפוש, עדיין כל הקשתות שנמצאות "בחזית" יוכפלו באותו ערך בזמן החיפוש של קשת עם עלות הזולה ביותר

אזי, כל גרף עם משקלים ניתן להביא כדוגמה נגדית לטענה.

2. ריק ומורטי (6 נק׳)

ריק ומורטי הלכו לאיבוד באגם הקפוא שלנו, שגודלו NxN. הם מעוניינים להיפגש, לא משנה באיזו משבצת, העיקר שעלות המסלול עד למפגש שלהם תהיה הזולה ביותר. בכל תור, כל אחד מהם מבצע את אחד מהצעדים הבאים {ימינה, שמאלה, למטה, למעלה, צעד במקום}. הניחו שכעת בלוח יש שני מצבים התחלתיים s_1, s_2.

1. הגדירו את (S, O, I, G) .

S- מרחב המצבים יוגדר כמיקומי ריק ומורטי באגם, סה"כ יהיו N*N*N*N מצבים.

O-קבוצת אופרטורים\פעולות תוגדר כצעדים שניתן לעשות באגם, שזה למעלה,למטה,ימינה,שמאלה, צעד במקום. סה"כ 5 אופרטורים

I-מצב התחלתי יסומן כמיקומים התחלתיים של ריק ומורטי באגם

G- קבוצת מצבים סופיים תורכב מכל המצבים בהם ריק ומורטי נמצאים באותה משבצת, סה"כ N*N מצבים סופיים

2. הגדירו את ה-$Domain$ לאחד האופרטורים לבחירתכם.

דומיין של אופרטור "צעד במקום" הוא כל המצבי S כי בכל מצב אפשר להישאר במקום

3. הגדירו את פונקציית ה-$Succ$ למצב ההתחלתי.

פונקציית Succ תלויה באיפה בדיוק באגם ממוקמים מצבים התחלתיים.
בכללי, עבור מצב התחלתי s_1 Succ שלו יהיו s_1 עבור פעולת עמידה במקום, וכל השכנים שאפשר להגיע אליהם בלי להתקע בקצה האגם או חור
במקרה שאין בעיות האלה נקבל קבוצה $s_1,s_1\text{-}N,s_1\text{+}N,s_1\text{-}1,s_1\text{+}1$

4. הציעו יוריסטיקה קבילה (שאינה יוריסטיקה ה-0).

יוריסטיקה קבילה אפשרית היא Manhetten, כי באגם אפשר ללכת רק בצורה אנכית למסלול.

יוריסטיקה לא אפס כי היא מייצגת מרחק בין ריק ומורטי במהלכים אנכיים

יוריסטיקה קבילה כי מרחק בין ריק ומורטי במקרה הטוב הוא מהלך ישר בקו אחד או שניים, שזה בדיוק היוריסטיקה שלנו. לכן היא תמיד מציגה גדלים שהם במקרה הטוב כגודל מרחק האמיתי, ולא גדולים ממנו.

חלק ה' – הגשת המטלה

מעבר למימוש ולדו״ח, ציונכם מורכב גם מהגשה תקינה של המטלה לפי הכללים הבאים:

1. יש לכתוב קוד ברור:
   a. קטעי קוד מסובכים או לא קריאים יש לתעד.
   b. לתת שמות בעלי משמעות למשתנים.
2. הדוח:

a. יש לכתוב בדוח את תעודת הזהות של שני המגישים.
b. הדו״ח צריך להיות מוקלד במחשב ולא בכתב יש. הדוח צריך להיות בפורמט pdf.
c. יש לשמור על סדר וקריאות בתוך הדו״ח.
d. אלא אם נכתב אחרת יש לנמק את התשובות.

3. הגשה:
   a. יש להעלות לאתר קובץ zip בשם: AI1_123456789_123456789.zip (עם תעודות הזהות שלכם במקום המספרים).
   b. בתוך ה-zip צריך להיות:
      i. הדו״ח הסופי בפורמט pdf בשם AI1_123456789_123456789.pdf.
      ii. קובץ ה-notebook בשם: AI1_123456789_123456789.ipynb

שימו לב: הקוד שלכם ייבדק ע״י מערכת בדיקות אוטומטיות תחת מגבלות זמני ריצה. במידה וחלק מהבדיקות יכשלו (או לא יעצרו תוך זמן סביר), הניקוד עבורן יורד באופן אוטומטי. לא תינתן הזדמנות להגשות חוזרות. אנא דאגו לעקוב באדיקות אחר הוראות ההגשה. שימו לב כי במהלך חלק מהבדיקות ייתכן שחלק מהקבצים שלכם יוחלפו במימושים שלנו. אם עקבתם אחר כל הדגשים שפורטו במסמך זה -עניין זה לא אמור להוות בעיה.

לא תתאפשרנה הגשות חוזרות, גם לא בגלל טעות טכנית קטנה ככל שתהיה. אחריותכם לוודא טרם ההגשה שהתרגיל רץ בסביבה שהגדרנו ושהקוד עומד בכל הדרישות שפירטנו.

אנא עברו בשנית על ההערות שפורסמו בתחילת מסמך זה. וודאו שאתם עומדים בהם.

שימו לב: **העתקות תטופלנה בחומרה.** אנא הימנעו מאי-נעימויות.

מקווים שתיהנו מהתרגיל!