תרגיל הגשה

שם הקורס: מבוא לבינה מלאכותית

מספר התרגיל: 1

מגישים: עופר ניסים – ת.ז 312367576

רועי קריניץ – ת.ז 205907777

תאריך: 1.12.22

מבוא לבינה מלאכותית - 236501

תרגיל בית 1

<u>מרחבי חיפוש</u>

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 236501-hw1.ipynb. אנחנו ממליצים לעבוד ב-[Google Colab](#). כדי לעשות זאת עליכם להעלות את תוכן התיקייה של התרגיל לתוך תיקייה ב-Drive האישי שלכם. לאחר מכן פתחו את המחברת דרך Google Colab ופעלו לפי ההוראות.

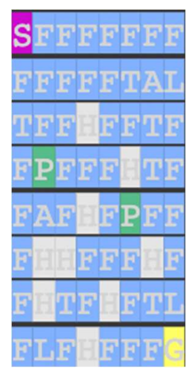
מומלץ לעבור על הקוד במחברת במקביל לקובץ הנוכחי וככה שלב שלב לענות על השאלות השונות.

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 BFS-G ועיצרו שם.
2. יבש (1 נק׳): תחילה נרצה להגדיר את מרחב החיפוש כפי שנלמד בתרגול. הגדר את $(S,0,I,G)$ עבור סביבת האגם הקפוא. מה גודל מרחב המצבים S? הסבירו.

נגדיר S: מיקום השחקן על גבי הלוח, כאשר לכל משבצת בלוח יש מספר state (מיקום השחקן הוא מספר בין 0-63, ניתן לייצוג גם כזוג סדור (i,j) כאשר i,j בין 0-7).

נגדיר O: מעבר השחקן מהמשבצת הנוכחית שלו בלוח למשבצת שנמצאת מעליו, מתחתיו, מימינו או משמאלו.

נגדיר I: המצב ההתחלתי של המשחק השחקן ממוקם במשבצת (0,0) וממנה מתחיל לזוז, state 0.

נגדיר G: מצב סופי, כאשר השחקן נמצא על משבצת (7,7) כאשר השחקן מגיע לשום מסתיים המשחק, state 63.

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

הפונקציה תחזיר לנו את כל המצבים בלוח חוץ מהמצבים שהם חור. מכיוון שמכל מצב נתן ללכת למעלה אם הוא לא חור, גם המצבים בשורה העליונה, אם נפעיל עליהם את האופרטור "UP" נקבל את אותו המצב.

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

הפונקציה Succ על המצב ההתחלתי תחזיר לנו את המצב 0 (הראשוני), את המצב עם האינדקס 1, ואת המצב עם האינדקס 8. כלומר משבצות: { (0,0), (1,0), (0,1) }

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

כן קיים מעגל, למשל מהמצב ההתחלתי נבצע את הפעולות: DOWN, UP, DOWN ונחזור למצב ההתחלתי.

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

מקדם הסיעוף הוא 4, מאחר והוא מקדם הסיעוף המקסימלי בגרף (קיימת משבצת שממנה ניתן לפנות למעלה, למטה, ימינה, שמאלה).

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

אינסוף פעולות, מכיוון שהוא יכול לחזור על אותם משבצות לנצח, למשל UP DOWN UP DOWN וחוזר חלילה.

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

9 פעולות: DOWN, DOWN, DOWN, RIGHT, RIGHT, RIGHT, DOWN, DOWN, DOWN

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 בהכרח, בלוח כללי אנו לא יודעים איפה נמצא הPORTAL ויתכן שבמצב מסוים הוא יאריך לנו את המסלול / יביא לחורים.

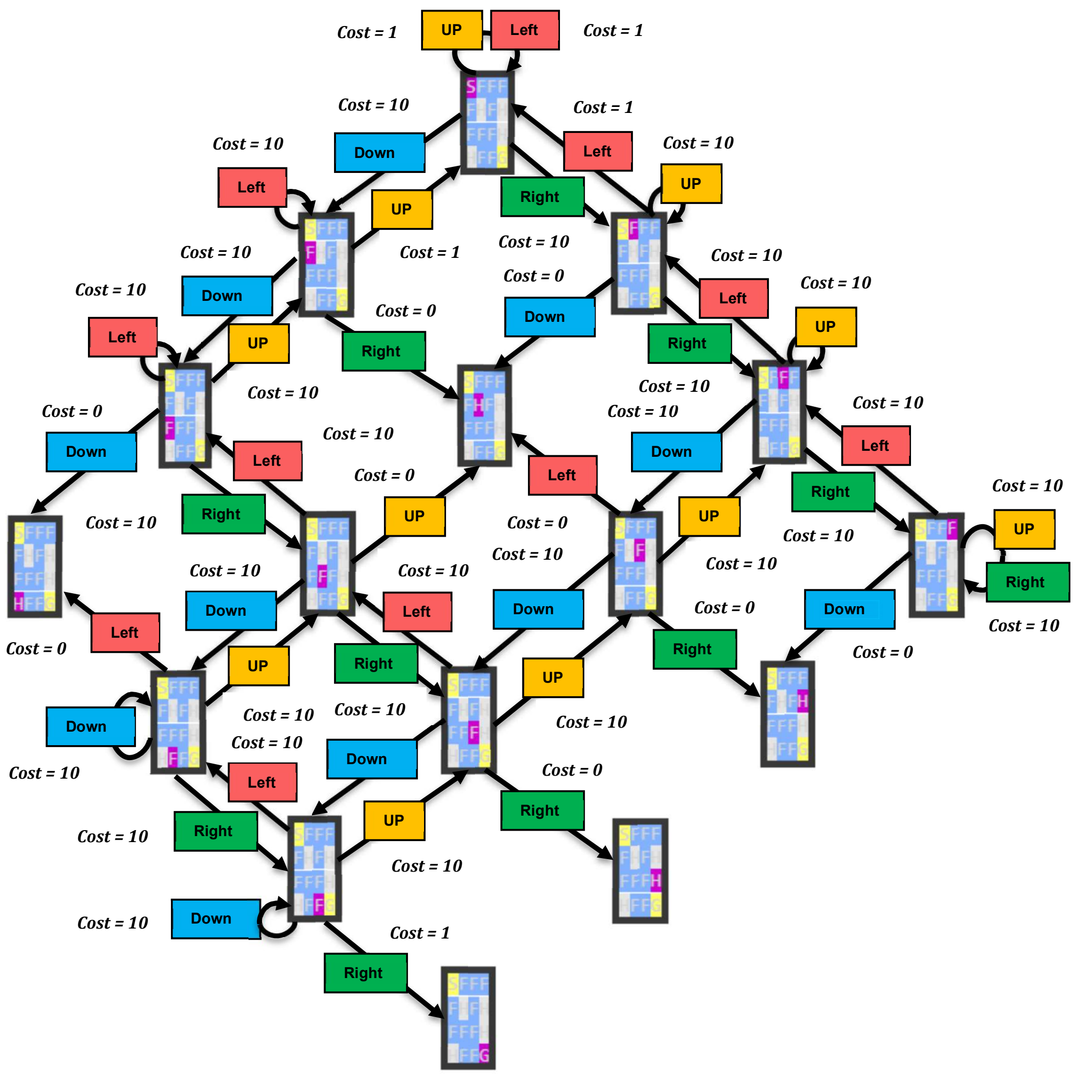
שאלה 2 – BFS-G (7 נק׳):

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?

התנאי על גרף החיפוש צריך להיות שהגרף הוא עץ מכוון, כך ש-BFS על עץ לא יאפשר חזרה למצב שכבר נוצר (דבר שלא יכול לקרות ב-BFS על גרף).

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



4. יבש (2 נק׳): יבש (2 נק׳): נתון לוח בגודל NxN שלא מכיל portals. הציעו דרך להשתמש באלגוריתם BFS-G כך שיחזיר פתרון אופטימלי (עלות מינימלית) והסבירו.
   - רמז: עליכם לספק פונקציה $T: G \to G'$ המקבלת את גרף המצבים G ויוצרת גרף חדש G' ובעזרתה למצוא את המסלול האופטימלי בגרף G.

כדי שהאלגוריתם BFS-G יחזיר פתרון אופטימלי (בעל עלות מינימלית, כלומר "קל ביותר") ולא רק קצר ביותר - נדאג שהגרף 'G שהפונקציה תחזיר, יהיה גרף בעל משקל אחיד על הקשתות. לשם כך נגדיר כי הפונקציה T תיצור העתק של הגרף המקורי G עם השינויים הבאים:

- **עבור כל צומת בעל עלות 0 בגרף המקורי, הפונקציה לא תיצור את הצומת בגרף החדש.**
- **עבור צומת בעל עלות 1 בגרף המקורי, הפונקציה תיצור את הצומת בגרף החדש כפי שהיה.**
- **עבור צומת בעל עלות גדולה מ-1 (שנסמנה a) בגרף המקורי, הפונקציה תיצור "שרוך" בעל עלות אחידה, כלומר רצף של a צמתים וקשתות, כך שדרגת הכניסה והיציאה של כל צומת פנימי בשרוך היא 1 (ראש וסוף השרוך הם הצמתים שמייצגים את הצומת עצמו מהגרף המקורי, כך שכל הקשתות שנכנסות לראש הן הקשתות שנכנסות לצומת בגרף המקורי, ובאופן דומה כל הקשתות שיוצאות מהסוף הן הקשתות שיוצאות מהצומת בגרף המקורי), ועלות כל צומת בשרוך היא 1.**
- **הפונקציה תיצור את הקשתות בהתאם לקשתות מהגרף המקורי עבור כל צומת שהוכנס לגרף, לפי הכללים שצוינו.**

בעזרת השינויים שלעיל הפונקציה T יוצרת גרף 'G מתאים לגרף G, כך שעלות כל הצמתים אחידה (ערך 1 לכל צומת) ומכאן שהאלגוריתם BFS-G אכן יחזיר פתרון אופטימלי.

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?

יפותחו N^2-2 צמתים וייווצרו N^2 צמתים. הלוח מכיל N^2 משבצות כך שהמצב ההתחלתי במשבצת 0 והמצב הסופי במצב N^2-1, וזהו המרחק הגדול ביותר בין שתי משבצות על גבי הלוח (2N-2 משבצות בהתאם למרחק מנהטן שראינו בכיתה), לכן היות ש-BFS-G יוצר ומפתח צמתים בהתאם למרחקן מההתחלה, ניאלץ ליצור את כל הצמתים על הלוח כדי להגיע למצב הסופי (שהוא הרחוק ביותר מההתחלה). נשים לב כי למצב הסופי יש שתי משבצות סמוכות, שתיהן במרחק זהה מן ההתחלה ולכן נגיע למצב הסופי בעת הפיתוח של הראשונה מתוכן. כמו כן, לא נפתח גם את המצב הסופי, ולכן יפותחו כל הצמתים מלבד הסופי ואחת המשבצות השכנות, כלומר N^2-2 צמתים.

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?

האלגוריתם שלם, שכן אם קיים מסלול מההתחלה עד למצב הסופי – DFS-G ימצא אותו (מעבר על כל הלוח במקרה הגרוע, ללא חזרה על מצבים שכבר פותחו).
האלגוריתם לא קביל מאחר שיכול להיות ש-DFS ימצא מסלול יותר ארוך מהמסלול הקצר ביותר או מסלול עם עלות גבוהה יותר, למשל בהרצה שביצענו בחלק הרטוב.

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?

במצב זה ייתכן שלא היינו מוצאים פתרון, מכיוון שבאלגוריתם זה אנחנו לא שומרים את רשימת הצמתים שכבר ביקרנו בהם ולכן ההתקדמות על גבי הלוח הייתה לכיוון DOWN בלבד עד המשבצת בפינה השמאלית-תחתונה (במצב בו לא הופיעו חורים במסלול) ומשם חזרה אינסופית על צעד DOWN תוך הישארות בפינה זו (כפי שהוגדר האופרטור על תזוזה בפינות, לפי סדר הפעולות ב-ACTIONS). אם הופיעו חורים במסלול, ייתכן מעבר דרכן ל-ACTION שאינו DOWN (למשל RIGHT) כך שהאלגוריתם היה מוצא פתרון אפשרי.

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?

יפותחו 2N-2 צמתים וייווצרו 4N-5 צמתים. המסלול DFS-G ינסה להתקדם "לעומק" עד שימצא פתרון או שיאלץ לחזור אחורה ולבחור במסלול הסתעפות אחר. אנו יודעים כי התיעדוף ב-ACTIONS הוא קודם DOWN ואז RIGHT, כך שהמסלול שהאלגוריתם ימצא יהיה התקדמות למטה עד הפינה השמאלית-תחתונה ואז התקדמות ימינה עד המצב הסופי. במהלך זה יפותחו N-1 צמתים כדי להגיע לפינה השמאלית-תחתונה ו-N-1 צמתים נוספים כדי להגיע למצב הסופי (ללא פיתוח של המצב הסופי), כלומר 2N-2 צמתים בסך הכל. הצמתים שייווצרו ע"י האלגוריתם יהיו 2N-1 צמתי המסלול ובנוסף צמתי N-1 העמודה השנייה ו-N-3 צמתי השורה שמעל האחרונה (ללא פיתוח פעמיים של אותו צומת ולא כולל הצומת שמעל צומת המטרה), לכן ייווצרו בסך הכל $2N-1+N-1+N-3=4N-5$ 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 יפותחו וייווצרו במהלך חיפוש Backtracking-G? הסבירו?

יפותחו 2N-2 צמתים וייווצרו 2N-1 צמתים. האלגוריתם Backtracking-G יוצר את הצמתים בשיטת "lazy", כלומר בכל שלב ייווצר הבן שאותו אנו רוצים לפתח. לכן בדומה למסלול ההתקדמות שתואר בסעיף הקודם, יפותחו N-1 צמתים כדי להגיע לפינה השמאלית-תחתונה ו-N-1 צמתים נוספים כדי להגיע למצב הסופי, כלומר 2N-2 צמתים בסך הכל. עבור כל פיתוח אני יודעים שנוצר צומת ובנוסף לא ייווצרו הצמתים מימין/מעל לצמתי המסלול, בניגוד לאופן היצירה שתואר בסעיף הקודם (בשל אופן הפעולה ה"עצל", המפתח בן יחיד בכל פעם), ולבסוף ייווצר המצב הסופי. בסך הכל נקבל שנוצרו $2N-2+1=2N-1$ צמתים.

שאלה 4 – ID-DFS-G (5 נק׳):

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

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, מכיוון שהפתרון קיים בעומק סופי נמצא את הפתרון לפי מה שלמדנו בכיתה (עומק של $O(N^2)$).

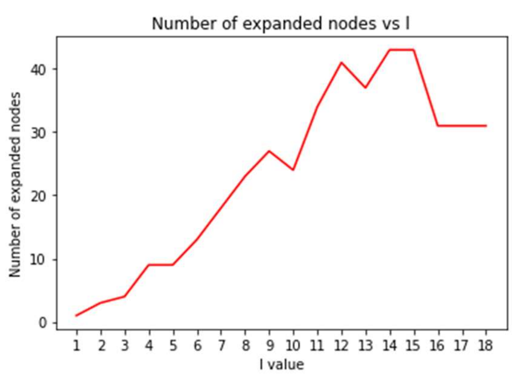
3. יבש (2 נק׳): נניח כי עלות כל פעולה היא 1, האלגוריתם ID-DFS-G לא קביל. הסבירו מדוע. בנוסף, הציעו דרך לעדכן את האלגוריתם על מנת שיהיה קביל במקרה הזה.

האלגוריתם לא קביל בשל ההימנעות מהכנסה חוזרת של צומת ל CLOSE, אם היא כבר שם או שפיתחנו אותה בעבר. במצב כזה, קיים תרחיש בו הגענו לצומת בחיפוש קודם כאשר היא הייתה בעומק L של חיפוש בעומק L, ולא המשכנו לחפש ממנה, כך שבפעם הבאה שנגיע אליה דרך מסלול אחר שאולי יהיה עדיף, לא נמשיך לחפש דרכה כי כבר ראינו אותה.

כדי לפתור את הבעיה, כל פעם שנמצא צומת נבדוק אם הוא נמצא ב-CLOSE ואם המסלול הנוכחי קצר יותר מהמסלול הקודם שגילינו, ואם כן להוציא אותו מCLOSE חזרה לOPEN.

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

<u>הגרף שהתקבל:</u>



ניתן לראות כי מספר הצמתים שהאלגוריתם מפתח גדל ככל ש-l גדל, שכן אנו מגדילים את עומק החיפוש. נשים לב כי החל מהאיטרציה ה-16 מספר הצמתים המפותחים קבוע, והוא קטן יותר ממספרם באיטרציה הקודמת, לכן אנו מסיקים שבשלב זה נמצא פתרון והאלגוריתם הפסיק לפתח צמתים נוספים.

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.

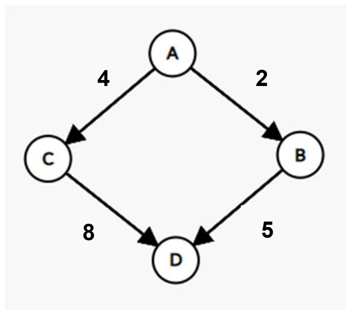
שני האלגוריתמים הנ"ל יפעלו באותו אופן עבור בעיות חיפוש בהן עלות כל הצעדים אחידה.

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

בבעיית החיפוש שלנו, האלגוריתם שלם (פונקציית המחיר חסומה מלמטה ע"י הערך 1, שכן עבור H שערכו 0 האלגוריתם לא ממשיך בפיתוח הצומת) וקביל, נובע מהוכחת הנכונות של אלגוריתם Dijkstra – כלומר אם הוצא צומת מ-OPEN, אנו יודעים שמדובר במסלול הזול ביותר מההתחלה לצומת הנוכחי. לכן כאשר מוצא צומת המטרה מ-OPEN, מתקבל בסה"כ המסלול הזול ביותר מההתחלה ועד היעד.

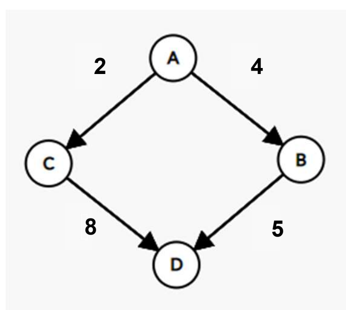
4. יבש (2 נק׳): דן טעה במימוש של אלגוריתם UCS ובטעות בדק בעת יצירת הצומת האם היא צומת מטרה במקום בפיתוח שלה. הביאו דוגמה לגרף חיפוש שעבורו דן יחזיר בכל זאת את המסלול הקל ביותר ודוגמה לגרף חיפוש שעבורו דן לא יחזיר את המסלול הקל ביותר. עבור כל דוגמה הסבירו מה המסלול והעלות ש-UCS השגוי החזיר, ומה המסלול והעלות שהאלגוריתם הנכון היה מחזיר. נדגיש שגרף החיפוש לא בהכרח צריך לייצג את בעיית האגם הקפוא. אתם יכולים לתת דוגמה לגרף שמייצג בעיית חיפוש אחרת. הגרף צריך להכיל קשתות מכוונות ואת העלות של כל קשת.

דוגמה לגרף חיפוש שעבורו דן יחזיר בכל זאת את המסלול הקל ביותר: (צומת התחלה A, צומת מטרה D)



המסלול שה-UCS השגוי החזיר הוא A→B→D והעלות שלו היא 2+5=7. המסלול והעלות של האלגוריתם הנכון זהים, היות שצומת היעד נוצר לאחר פיתוח הצומת שמביא למסלול הקל ביותר (ללא צורך בעדכון העלות של הצומת ב-OPEN).

דוגמה לגרף חיפוש שעבורו דן <u>לא</u> יחזיר בכל זאת את המסלול הקל ביותר: (צומת התחלה A, צומת מטרה D)



המסלול שה-UCS השגוי החזיר הוא A→C→D והעלות שלו היא 2+8=10.

המסלול שה-UCS הנכון החזיר הוא A→B→D והעלות שלו היא 4+5=9.

השוני בין המסלולים והעלויות נובע מכך שצומת היעד נוצר לאחר פיתוח הצומת שאינו מביא למסלול הקל ביותר, למרות שעלותו נמוכה יותר משל הצומת שמביא למסלול הקל ביותר (כלומר ישנו צורך בעדכון העלות של הצומת ב-OPEN, אך עדכון זה לא יתרחש בשל המימוש השגוי והאלגוריתם יעצור לאחר יצירת צומת המטרה D).

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.

היוריסטיקה קבילה על כל לוח. נשים לב שאם מדובר בצומת חור אז המרחק לצומת סופית יהיה אינסוף, והיוריסטיקה תניב לכל היותר 100, כלומר תקיים את התנאי הנדרש לקבילות. לכל צומת אחר על הגרף, עלות כל מעבר במסלול האופטימלי שלה הוא לכל הפחות 1, ולכן מרחק מנהטן שנותן 1 לכל צעד היא יוריסטיקה קבילה. בנוסף, אם עברנו בפורטל במסלול אז עלות המסלול תהיה גדולה שווה ל-100, ובמקרה כזה h_{SAP} תניב לכל היותר 100, לכן היוריסטיקה קבילה.

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

היוריסטיקה לא עקבית, ניתן דוגמא נגדית:

נגדיר לוח כללי 8X8 עם 2 חורים בלבד וכל שאר המשבצות, מלבד משבצת המטרה ומשבצת ההתחלה, הן משבצות F רגילות. המשבצות (1,0) ו-(1,1) הן חורים, אזי: $h_{sap}((1,0))-h_{sap}((1,1))=1>0=cost((1,0),(1,1))$, כלומר אינה מקיימת את התנאי הנדרש 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 אך לא קביל, שכן הוא ימשיך לפתח את המצבים האפשריים עד שיגיע לצומת מטרה, כאשר היוריסטיקה רק קובעת את הסדר. לכן מכיוון שקיים פתרון, נמצא אותו. אבל האלגוריתם לא קביל, נשים לב שאלגוריתם UCS מוצא מסלול בעל עלות נמוכה יותר בלוח הנתון בתרגיל הרטוב.

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

<u>חסרון:</u> Greedy Best First Search אינו אלגוריתם קביל, לעומת UCS.

<u>יתרון:</u> Greedy Best First Search מתקדם באופן ממוקד לכיוון המטרה לפי היוריסטיקה (אלגוריתם מיודע), ולכן יפתח פחות צמתים, כך שהוא צפוי לזמן ריצה טוב יותר תוך פיתוח מספר קטן יותר של צמתים (חיסכון בזיכרון נוסף).

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 דוגמא נגדית את קבילותן של היוריסטיקות:
   a. $GreedyHeuristic(s)=0\ if\ s\ is\ goal,else\ 1$
   b. $h_{MD}(s)=h_{Manhata}(s,g),g\in G$
   c. $NearestPortalOrGoalHeuristic(s)=\min\{h_{Manhatan}(s,p_1),h_{Manhatan}(s,p_2),h_{Manhatan}(s,g)\},g\in G$
   כאשר p_1,p_2 הם משבצות שיגור $(p_1\neq p_2)$ ו-g הוא צומת מטרה.
   d. $NearestPortalToFinalHeuristic(s)=NearestPortalOrGoalHeuristic(s)+h_{Manhatan}(p_1,p_2)$

a. היוריסטיקה קבילה מאחר של הערך היוריסטי של כל צומת הוא 1 (חוץ מהצומת הסופי), והמרחק שלה מהצומת הסופי בגרף יהיה לפחות 1, שכן עלות כל צעד הוא לפחות 1. עבור מצב שהוא חור, עלות המסלול ממנו לצומת היעד היא אינסוף והיורסטיקה a שלו תהיה 1. עבור מצב סופי עלות מסלול ממנו למצב סופי הוא 0 ואכן ערך היורסטיקה שלו הוא 0. לכן היורסטיקה קבילה.

b. היוריסטיקה קבילה, לכל מצב בלוח, ערך המסלול מממנו למצב היעד יהיה גדול יותר או שווה למרחק מנהטן בינו לבין המצב הסופי. זאת מכיוון שבמקרה הטוב ביותר, במסלול זול ביותר יהיו את אותו מספר הצעדים כמו במרחק מנהטן. נשים לב כי לכל מצב, הערך היוריסטי קטן או שווה מעלות מסלול אופטימלי, גם אם בכל מסלול אופטימלי (מהמצב לפתרון) יש פורטל שעלותו היא 100 ולכן יהיה גדול מאשר יורסיטיקת מנהטן המקסימלית, שהיא 14.

c. מכיוון שיורסטיקה b מיודעת יותר מיורסטיקה c, ניתן להניח כי יורסטיקה c קבילה.

d. היורסטיקה לא קבילה – בלוח הנתון המצב הסופי יקבל ערך יוריסטי של 5, מאחר ומרחק מנהטן בין הפורטלים הוא 5, אבל המסלול ממצב סופי למצב סופי הוא 0.

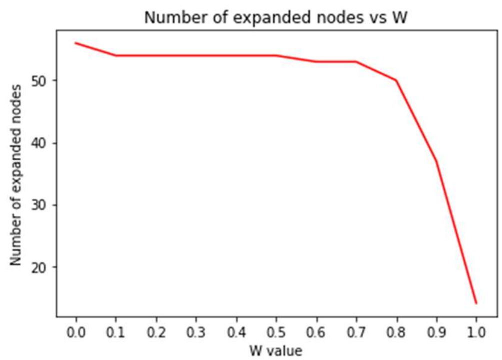
3. יבש (נק׳ 1): איזו יוריסטיקה (מבין הקבילות) היא המיודעת ביותר (כולל היוריסטיקה h_{SAP})?

h_{sap} (וגם h_{MD} שמניבה ערכים זהים בלוח) היא היוריסטיקה המיודעת ביותר, מאחר שלכל מצב היא נותנת את מרחק מנהטן שלו, שמהווה יורסטיקה עם ערכים מקסימליים מבין היוריסטיקות המוצעות. (לא כולל d שאינה קבילה).

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

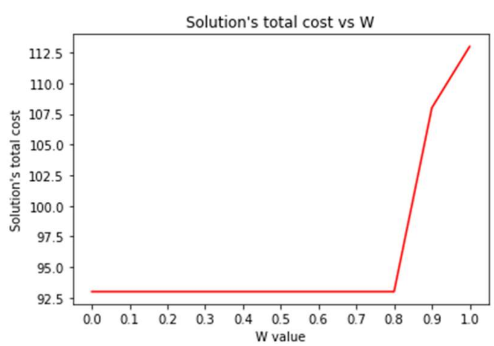
   ניתן לצייר את הגרף ביד או במחשב. הסבירו כל גרף בנפרד וגם את הקשר ביניהם.

<u>גרף 1:</u>



ניתן לראות כי מספר הצמתים שפותחו קטן ככל שערכו של W גדל (כלומר כאשר האלגוריתם נוטה יותר לכיוון GRBFS, לעומת UCS שנמצא ב-W=0), תוצאה שמתיישבת באופן הגיוני עם העובדה שהחיפוש נעשה מיודע יותר ומסתמך יותר על היוריסטיקה כדי להגיע לפתרון בצורה ממוקדת (בהתאם לנימוקים אודות היתרון שציינו בשאלה 7 סעיף 3). הירידה החדה במספר הצמתים שפותחו מתרחשת סביב הערכים $0.8 \le W \le 1$.

<u>גרף 2:</u>



נבחין כי עבור אלגוריתם ה-UCS (כאשר W=0) עלות הפתרון היא אופטימלית, ועלות זו נותרת קבועה עבור $0 \le W \le 0.8$, אך עבור ערכים גבוהים יותר (כאשר האלגוריתם נוטה יותר לכיוון GRBFS) ישנה עליה חדה בעלות הפתרון.

הקשר בין הגרפים קל להבחנה, שכן גם במספר הצמתים שפותחו וגם בעלות הפתרון ישנו ערך קבוע בגרפים עבור $0 \le W \le 0.8$, ולאחר מכן ($0.8 \le W \le 1$) ישנה ירידה חדה במספר הצמתים שפותחו ובמקביל עלייה חדשה בעלות הפתרון. נראה כי החיפוש המיודע אמנם מעניק פתרון מהירים יותר, תוך פיתוח מספר מצומצם יותר של צמתים, אך זאת על חשבון עלות הפתרון שעשויה להיות רחוקה מאופטימלית במקרים מסוימים.

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*.

<u>יתרון:</u> מאפשר בחירה לא אופטימלית, בחירה משנית לפי יורסיטיקה FOCAL בשאיפה למצוא פתרון מהר יותר.

<u>חסרון:</u> אנו עלולים לקבל פתרון שעלותו תהיה גדולה יותר מהפתרון האופטימלי, לכל היותר פי (epsilon+1).

3. יבש/**רטוב** (4 נק׳): תנו הצעה ליוריסטיקה כדי לבחור את הצומת הבא 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.**

נגדיר יורסטיקה h_{state}, כך שלכל מצב state, הפעלת היוריסטיקה עליו תחזיר את האינדקס שלו, לדוגמה: $h_{state}(I) = 0$,$h_{state}(63) = 63$.

<u>עבור epsilon=1:</u>

	יורסיטיקה g	יוריסטיקה State
מסלול	[1, 1, 1, 1, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0]	[1, 1, 1, 1, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0]
עלות המסלול	**93**	**93**
כמות הצמתים שפותחו	**56**	**67**

<u>עבור epsilon=0.5:</u>

	יורסיטיקה g	יוריסטיקה State
מסלול	[1, 1, 1, 1, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0]	[1, 1, 1, 1, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0]
עלות המסלול	**93**	**93**
כמות הצמתים שפותחו	**56**	**64**

ניתן לראות שבחירה משנית ביוריסטיקת g טובה יותר מהיוריסטיקה החדשה שהגדרנו.

חלק ד׳ – שאלות בסגנון מבחן (12 נקודות)

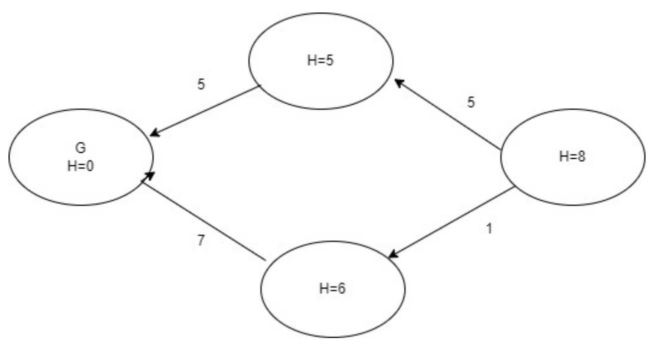
1. וריאציות של *A (6 נק׳)

1. נגדיר n' להיות צומת האב של צומת n בגרף. כמו כן נניח ש- h היא יוריסטיקה קבילה שאינה 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.
   b. *A שמתעלם מערך ה-h.
   c. *A שמתעלם מערך ה-g.
   d. *A כך ש- $f(n) = g(n) + h(n')$
   e. *A כך ש- $f(n) = g(n') + h(n)$

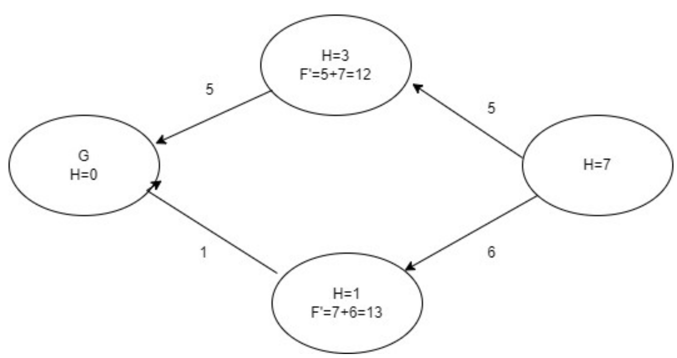
a. לפי מה שלמדנו בהרצאה *A הוא אלגוריתם שלם וקביל, בהינתן שעלות הפתרון האופטימלי היא סופית.

b. כאשר *A מתעלם מערכי h הוא מתנהג כמו אלגוריתם UCS שהוא אלגוריתם שלם וקביל (גם כאן בהינתן שעלות הפתרון האופטימלי היא סופית וערכי הקשתות חסומות מלמטה ע"י ערך חיובי).

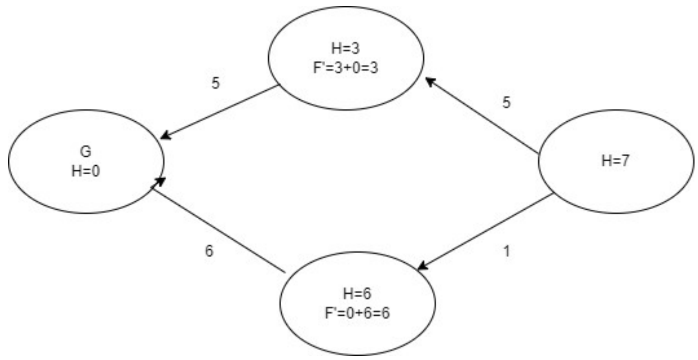
c. כאשר *A מתעלם מ-g הוא מתנהג כמו Greedy best first search, שזה אלגוריתם שלם בהינתן שגרף המצבים במרחב סופי וקשיר, אך זה אלגוריתם לא קביל, למשל בדוגמה שלפנינו - נבחר במסלול העליון במקום במסלול האופטימלי התחתון: (התחלה בצומת הימני וסיום בשמאלי)



d. במקרה המתואר *A שלם, אך לא קביל, (בהנחה שמחיר כל הקשתות חסום מלמטה ע"י מספר חיובי). ניתן דוגמה נגדית: בדוגמה הזו שוב נבחר במסלול העליון הלא אופטימלי, אך נמצא פתרון ולכן שלם אך לא קביל.



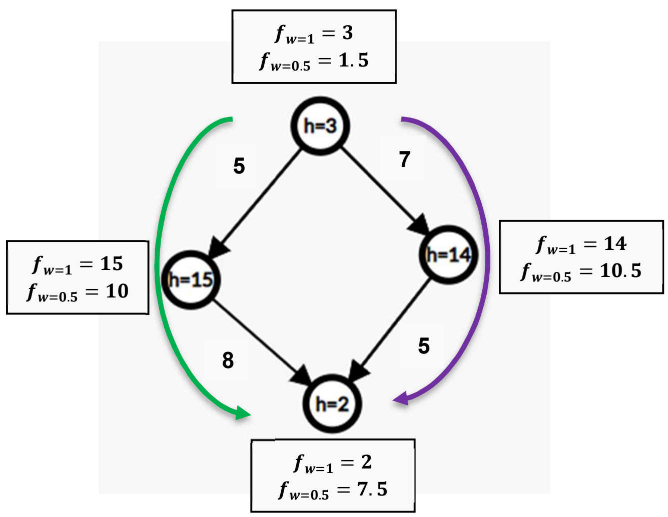
e. במקרה המתואר A* שלם, אך לא קביל, (בהנחה שמחיר כל הקשתות חסום מלמטה ע"י מספר חיובי). נתן דוגמה נגדית:
בדוגמה הזו שוב נבחר במסלול העליון הלא אופטימלי. אך נמצא פתרון ולכן שלם אך לא קביל.



2. כעת נתון $0.5 \le w_1 < w_2 \le 1$. 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: $cost(P_1) \le cost(P_2)$.

הטענה השגויה, דוגמה נגדית:

נתבונן בגרף הבא, בו צומת ההתחלה הוא הצומת העליון וצומת היעד הוא הצומת התחתון. נבחן את המסלולים המוחזרים מריצת האלגוריתם הנ"ל עבור $w = 0.5$ ו-$w = 1$:

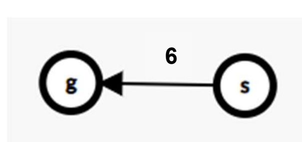


עבור $w = 0.5$ יבחר המסלול השמאלי (צבע ירוק), שעלותו היא $cost(P_1) = 13$
עבור $w = 1$ יבחר המסלול הימני (צבע סגול), שעלותו היא $cost(P_2) = 12$
ובסך הכל נקבל $cost(P_1) = 13 > 12 = cost(P_2)$, בסתירה לטענה.

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.

הטענה השגויה, דוגמה נגדית:

עבור הגרף הבא, קיים מסלול יחיד בין צומת ההתחלה s לצומת היעד g, והוא יוחזר ע"י אלגוריתם UCS ללא תלות בהוספת ערך קבוע כלשהו לכל עלות קשת במהלך החיפוש.



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

הטענה השגויה, דוגמה נגדית:

עבור הגרף שהצגנו בטענה הקודמת, קיים מסלול יחיד בין צומת ההתחלה s לצומת היעד g, והוא יוחזר ע"י אלגוריתם UCS ללא תלות בהכפלת ערך קבוע כלשהו כל עלות קשת במהלך החיפוש.

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) .

נגדיר S: מיקום שני השחקנים על גבי הלוח, כאשר לכל משבצת בלוח יש מספר State (מיקומי השחקנים הם מספרים בין 0-63). את המיקום נייצג על ידי הזוג הסדור (i, j) עבור $0 \leq i, j \leq 63$ כך ש-i מייצג את מיקומו של ריק ו-j את מיקומו של מורטי.

נגדיר O: מעבר השחקנים מהמשבצות הנוכחיות בלוח למשבצות שנמצאות מעליהם, מתחתיהם, מימינם או משמאלם או צעד במקום (באופן בלתי תלוי אחד מהשני). נייצג ע"י זוג סדור (O_1, O_2), כך ש-$O_1, O_2 \in \{DOWN, UP, LEFT, RIGHT, STAY\}$ והאיבר השמאלי O_1 מייצג את המהלך שריק מבצע, האיבר הימני O_2 מייצג את המהלך שמורטי מבצע.

נגדיר I: המצב ההתחלתי של המשחק, בו השחקנים ממוקמים במשבצות s_1, s_2 וממנה נתחיל. נייצג: (s_1, s_2).

נגדיר G: מצב סופי, כאשר שני השחקנים נפגשים על משבצת כלשהי שנסמן את מצבה ב-$g \in \{0, \dots, N^2 - 1\}$. נייצג: (g, g).

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

נתייחס לאופרטור $O = (STAY, STAY)$ בו מתקיים $Domain(O) = S$, כלומר ניתן להפעיל את האופרטור לכל זוג סדור של מצבים על גבי הלוח.

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

נגדיר את הפונקציה עבור המצב ההתחלתי באופן הבא:

$$Succ(I) = Succ\big((s_1, s_2)\big) = \{(a, b) | (a, b) \in o(I), o \in O\}$$

כאשר a, b מהווים משבצות חוקיות שאליהן ניתן להגיע לאחר הפעלת אופרטור כלשהו $o \in O$ על המצב ההתחלתי (s_1, s_2).

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

היוריסטיקה שנציע היא יוריסטיקת מנהטן חלקי 3, כלומר מרחק מנהטן בין המיקום של ריק שנסמנו (x, y), לפי הייצוג של שורה ועמודה, למרחק של מורטי שנסמנו (a, b). נשים לב שהיוריסטיקה קבילה מאחר שכדי שהם יפגשו, הם צריכים יחד לעשות לפחות אותו מספר צעדים בכיוון האנכי והאופקי, כמספר הצעדים במסלול מנהטן (כלומר במסלול האידיאלי יידרש מספר צעדים על הלוח, השווה לחצי ממרחק מנהטן, לכן היוריסטיקה המוצעת תמיד תהיה נמוכה יותר בערכה מערך הפתרון האופטימלי).

$$h_{Man2}\big(((x, y), (a, b))\big) = \frac{|x - a| + |y - b|}{3}$$

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!