מבוא לבינה מלאכותית - 236501

תרגיל בית 1

מרחבי חיפוש

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:**29.2** יום חמישי, בשעה 23:59.(לשנות מועד ולפתוח פיאצה)
- את המטלה יש להגיש בזוגות בלבד.
- יש להגיש מטלות מוקלדות בלבד בעברית או באנגלית. פתרונות בכתב יד לא ייבדקו.
- ניתן לשלוח שאלות בנוגע לתרגיל בפיאצה בלבד.
- המתרגל האחראית על תרגיל: **שאדי דיאב** .
- 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- 15% מהציון הסופי במקצוע ולכן העתקות תטופלנה בחומרה!
- ציון המטלה יורכב מהגורמים הבאים:
  - 65% - המסמך היבש.
  - 35% - הקוד המוגש.
- 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.

הנחיות לחלק היבש

1. ככלל אצבע, בהינתן שאלה ראשית ספקו את התשובה המיידית ולאחר מכן תרחיבו ותסבירו. למשל, אם שואלים מה סיבוכיות הזמן של אלגוריתם BFS תשובה תהיה ״$O(b^d)$, מכיוון שבקרה הכי גרוע נאחסן את כל עץ החיפוש של הבעיה בCLOSE״.

הנחיות לחלק הרטוב

1. אנו מעודדים אתכם לעבור על הקבצים המצורפים ולהבין כיצד הסביבה בנויה ובאילו פונקציות תוכלו להשתמש במימוש שלכם.
2. הקוד שלכם ייבדק בקפדנות על ידי טסטים. הטסטים יבדקו את הפתרונות המוחזרים על ידי האלגוריתמים שלכם אל מול המימוש שלנו על פני בעיות שונות. אנו מצפים ממכם (אלא אם צוין אחרת)

להחזיר את אותם ערכים בדיוק. אנחנו נבדוק את המסלול המוחזר, מספר הצמתים שפתחו ואת עלות הפתרון המוחזר. הטסטים יהיו מוגבלים בזמן אך תקבלו זמן גדול מאוד לכל טסט.

3. ספקו קוד ברור ונקי הניתן לבדיקה ידנית.

מבוא ורקע

התרגיל מתפרש על פני מסמך זה והמחברת המצורפת. מומלץ לענות על השאלות לפי הסדר במסמך זה.

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

סיפור מסגרת

לקאקרוטו וגוהאן יש 5 כדורי דרקון וחסר להם שני כדורים, והם ממש צריכים אותם כדי להזמין הדרקון שן-ראן ולבקש ממנו להחזיר את החברים שלהם לחיים, לכן הם הלכו לכוכב לכת נאמיק כדי לחפש כדורי הדרקון, קאקרוטו הציע שיחפשו על הכדור דרך ה ג׳׳פי.אס שלהם אבל גוהאן מסביר לקאקרוטו שיש לו חברים שלוקחים הסמסטר את קורס ״מבוא לבינה מלאכותית״. גוהאן מבקש ממכם לעזור לו לתכנן את המסלול הטוב ביותר כדי לאסוף כדורי הדרקון ולהגיע לקאקרוטו שמחקה לו.

שאלה 1 – מבוא (8 נק׳):

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

```
SFFFFFFF
FFFFFTAL
TFFHFFTF
FFFFFHTF
FAFHFFFF
FHHFFFHF
DFTFHDTL
FLFHFFFG
```

1. **רטוב**: עברו על המחברת עד שאתם מגיעים לחלק של BFS-G ועצרו שם.
2. יבש (1 נק׳): תחילה נרצה להגדיר את מרחב החיפוש כפי שנלמד בתרגול. הגדר את $(S,0,I,G)$ עבור סביבת כדורי הדרקון. כאשר S זה מרחב המצבים, O, זה מרחב האופרטורים, I, זה המצב ההתחלתי וG הוא קבוצת מצבי המטרה. מה גודל מרחב המצבים S? הסבירו.

S-מרחב המצבים יוגדר לפי גודל הלוח ומיקום הסוכן וכן האם יש לנו את כדור הדרקון הראשון והשני.

$$S = \{p, b_1, b_2\} | p \in [0,1,\ldots,63], b_1, b_2 \in [0,1]$$

לפיכך גודל מרחב המצבים הוא 64*2*2 =256

O- יוגדר לפי האופרטורים האפשריים שלנו כלומר

$$O = \{Down, Right, Up, Left\}$$

I – יוגדר לפי המצב ההתחלתי כלומר $\{0,0,0\}$ המשבצת השמאלית עליונה וללא כדורי דרקון

- **G** יוגדר לפי מצב הסיום כלומר $\{63,1,1\}$ המשבצת הימנית התחתונה עם שני כדורי הדרקון

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

הפונקציה Domain מחזירה לכל אופרטור את כל המצבים שהפעלת האופרטור עליהם לא מחזירה קבוצה ריקה. כן
$Domain(up) = \{s|s \in S \ and \ s[0] \neq hole\}$

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

הפונקציה Succ על המצב ההתחלתי 0 תחזיר לנו

{0,0,0}

{8,0,0}

{1,0,0}

כאמור בפורמט שהגדרנו

{ position,dargonball1,dargonball2}

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

כן , קיימים מעגלים במרחב. ניתן דוגמא מהמצב ההתחלתי ניתן לנוע ימינה -> למטה -> שמאלה-> למעלה וככה הגענו למצב ההתחלתי שוב.

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

מקדם הסיעוף בבעיה הוא 4 מכל משבצת ניתן לנוע לכל היותר ל4 משבצות שונות

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

במקרה הגרוע ביותר יכול להיות שהסוכן כלל לא יגיע למצב הסופי, בין אם הוא יתקע במעגל או בין אם הוא יפול לחור. כלומר לסוכן ידרשו אינסוף פעולות

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

במקרה הטוב ביותר נלך עד למטה עד לכדור הדרקון הראשון. משם נפנה ימינה ונעקוף את החור מלמעלה ונאסוף את כדור הדרקון השני, ומשם נלך למצב הסופי – סך הכל 16 פעולות

9. יבש (1 נק׳): עבור לוח כללי, המסלול הקל ביותר הוא המסלול שמגיע למצב מטרה שהכי קרוב למצב ההתחלתי (במונחים של manhattan distance)? אם כן, הוכיחו. אם לא, ספקו דוגמא נגדית.

ניקח כדוגמא נגדית את הלוח שבעמוד הבא. מסלול שמגיע למצב שהכי קרוב למצב ההתחלתי במונחי מנהטן דיסטנס לעולם לא יעבור על שני המשבצות הימניות העליונות ביותר. ולכן כל מסלול כזה הינו כבד יותר.

S	F	L	L
F	F	F	F
F	F	F	F
F	F	F	G

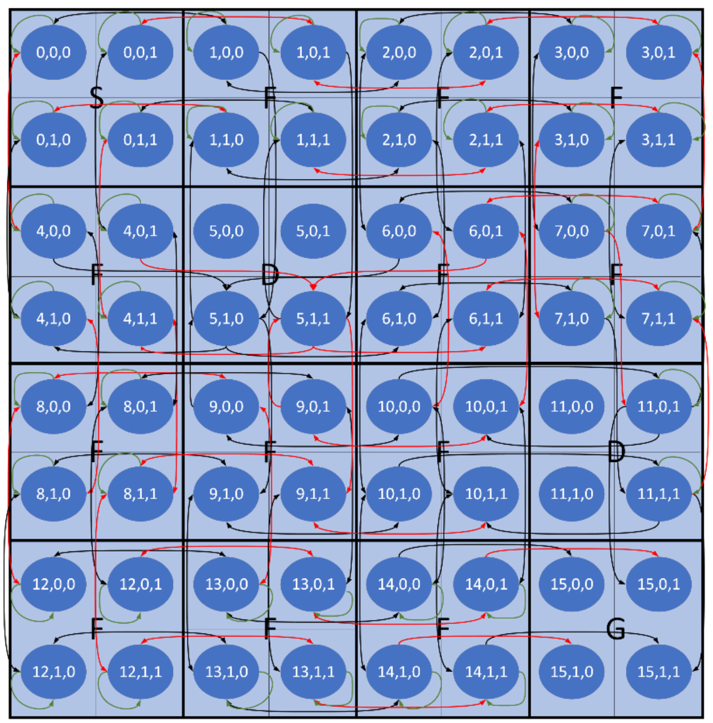
שאלה 2 – Breadth First Search-G (7 נק׳):

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 כדורי הדרקון) כך שBFS על גרף ו-BFS על עץ ייצרו ויפתחו צמתים זהים באותו הסדר?

התנאי על גרף החיפוש צריך להיות כזה שלא ניתן לפתח יותר צמתים שכבר פותחו, באמצעות שמירת הצמתים שפותחו בזיכרון.

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



*החצים מסומנים בצבעים שונים לצורך נוחות ומעקב נוח יותר אחר החצים בלבד.

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

מאחר ו BFS-G מגיע לפתרון אופטימלי תחת מחיר אחיד על הקשתות, ניצור פונקציה T אשר תקבל את גרף המצבים G ותיצור גרף חדש G' אשר מאריך (לעומק) כל צומת בעלות x לx צמתים משורשרים במשקל אחיד.

5. יבש (2 נק׳): נתון לוח בגודל NxN, ללא חורים, המכיל $N^2 - 2$ משבצות רגילות (F,T,A,L) מצב התחלתי בפינה השמאלית עליונה ומצב מטרה בפינה הימנית תחתונה. כמה צמתים יפותחו וייווצרו במהלך חיפוש BFS-G? הסבירו?

בעבור לוח שכזה ייווצרו NxN צמתים ויפותחו $N^2 - 2$ צמתים, מאחר ומדובר בלוח ריבועי ומאחר והמצב ההתחלתי ומצב המטרה נמצאים בשתי פינות מנוגדות, ומאחר והעומק למטרה d הינו מרחק מנהטן מהמצב ההתחלתי למצב המטרה והינו העומק המקסימלי בלוח זה $d = 2N$, אלגוריתם BFS-G מפתח לרוחב ומגיע לפתרון הרדוד ביותר d כאשר כל המסלולים בעומק d-1 ייווצרו לפני המסלולים בעומק d, ולכן כל הצמתים בלוח ייווצרו. לא כל הצמתים יפותחו מאחר וצומת המטרה אינו מפותח ובנוסף לכך, יהיה צומת נוסף שלא יפותח מאחר ותמיד בפיתוח האחרון יהיו שני צמתים שמהם יהיה ניתן להגיע לצומת המטרה בפינה, וBFS-G תמיד יגיע לפתרון דרך הצומת שפותחה קודם.

שאלה 3 – Depth First Search-G (6 נק׳):

1. יבש (1 נק׳): עבור בעיית כדורי הדרקון עם לוח NxN, האם האלגוריתם שלם? האם הוא קביל?

האלגוריתם שלם – מדובר בחיפוש בגרף סופי. לכן תחזוקת מבנה נתונים של CLOSE יביא לכך שלא נבצע מעגלים אינסופיים ולכן האלגוריתם כן יתכנס לפתרון אם קיים.
לא קביל – האלגוריתם מחזיר את הפתרון הראשון שהוא מוצא. הפתרון הראשון לא בהכרח מחייב שהוא האופטימלי ולכן יתכן שהאלגוריתם יחזיר פתרון שאינו אופטימלי. זאת מכיוון שהוא מחפש לעומק.

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

אלגוריתם על עץ היה יכול להתקע בלולאה אינסופית של מעגלים לכן לא מן ההכרח שהיה מוצא פתרון.

3. יבש (2 נק׳): נתון לוח בגודל NxN, ללא חורים, המכיל $N^2 - 2$ משבצות רגילות (F,T,A,L) מצב התחלתי בפינה השמאלית עליונה ומצב מטרה בפינה הימנית תחתונה (תניחו כי שני כדורי הדרקון הם בפינה ימנית תחתונה) . כמה צמתים יפותחו וייווצרו במהלך חיפוש DFS-G? הסבירו?

מספר הצמתים שיפותחו: $2N - 2$
מספר הצמתים שיווצרו: $4N - 5$
נשים לב שבמקרה הזה הצמתים שפותחו הן כל הצמתים על העמודה השמאלית ביותר והשורה התחתונה ביותר למעט צומת המטרה – כלומר $2N - 2$
מספר הצמתים שיווצרו הן כל הצמתים שפותחו וגם השכנות שלהן כלומר העמודה השניה משמאל והשורה התחתונה , כל אחת ללא הצומת האחרון בה ולכן נקבל $4N - 5$

4. יבש (2 נק׳): נתון לוח בגודל NxN, ללא חורים, המכיל $N^2 - 2$ משבצות רגילות (F,T,A,L) מצב התחלתי בפינה השמאלית עליונה ומצב מטרה בפינה הימנית תחתונה (תניחו כי שני כדורי הדרקון הם בפינה ימנית תחתונה). כמה צמתים יפותחו וייווצרו במהלך חיפוש DFS-G backtracking? הסבירו?

מספר הצמתי שיפותחו $2N - 2$
מספר הצמתי שיווצרו $2N - 1$
במצב הזה אפשר לראות כי ההיווצרות נעשית בצורה עצלנית ולכן מספר הצמתים שיפותחו זהה לסעיף הקודם אך מספר הצמתי שיווצרו כולל את כל הצמתי שפותחו + רק הצומת האחרון.

שאלה 4 – ID-DFS (6 נק׳):

.1

a. (1 נק׳) האם האלגוריתם שלם? אם כן, הוכיחו. אם לא, ספקו דוגמה נגדית.

כן, האלגוריתם שלם מאחר ומדובר בחיפוש בגרף בעומק סופי. לכן תחזוקת מבנה נתונים של CLOSE יביא לכך שלא נבצע מעגלים אינסופיים ולכן האלגוריתם כן יתכנס לפתרון אם קיים.

b. (1 נק׳) נניח כי עלות כל פעולה היא 1, האם האלגוריתם קביל? אם כן, הוכיחו. אם לא, הסבירו.

כן, האלגוריתם קביל מאחר ותחת מחיר אחיד על הקשות, אלגוריתם זה מחזיר תמיד את המסלול הקצר ביותר. כמו בחיפוש לרוחב, כל המסלולים בעומק d נבדקים לפני המסלולים בעומק d+1, פתרון אופטימלי הינו בעל העומק הקטן ביותר ולכן יימצא לפני כל פתרון אחר.

2. הניחו כי יש לנו ידע מקדים על חסם עליון למרחק למצב מטרה, נסמנן D. בת (Beth) הציעה את האלגוריתם חיפוש הבא:

```
function ReverseDFS (problem, D):

        L ← D
        result ← failure
        While Not Interrupted:
                new_result ← DFS-L (problem, L)
                if new_result = failure:
                        break
                L ← L - 1
                result ← new_result

        return result
```

3. בשאלות הבאות הניחו כי יש מספיק זמן לסיום האיטרציה הראשונה.

a. (1 נק׳) ספקו דוגמה בה ReverseDFS עדיף על ID-DFS ודוגמה בה ID-DFS עדיף על ReverseDFS. הדוגמאות יכולות להיות כלליות ולא בהכרח מסביבת התרגיל.

ReverseDFS יהיה עדיף מבחינת זמן ריצה על ID-DFS כאשר לדוגמא הפתרון האופטימלי (הרדוד ביותר) נמצא בעומק החסם העליון D, מאחר ובמצב זה לאחר איטרציה אחת בעומק זה יימצא פתרון והאלגוריתם ייעצר כאשר לעומת זאת ID-DFS יימצא פתרון רק באיטרציה האחרונה לאחר שיבצע כמה איטרציות של DFS-L על כל העומקים. ID-DFS יהיה עדיף מבחינת זמן ריצה ומקום על ReverseDFS כאשר לדוגמא הפתרון האופטימלי (הרדוד ביותר) נמצא בעומק L=1, מאחר ובמצב זה לאחר איטרציה אחת בעומק זה יימצא פתרון והאלגוריתם ייעצר כאשר לעומת זאת ReverseDFS יימצא פתרון רק באיטרציה האחרונה לאחר שיבצע כמה איטרציות של DFS-L על כל העומקים.

b. (2 נק׳) הציעו כיצד ניתן לייעל את האלגוריתם. רמז: האם אתם יכולים לחשוב על צעד עדכון עדיף לL?
נשנה את האלגוריתם בצורה הבאה :

במקום לעדכן את L בצורה הנתונה, ניתן לעדכן את L באמצעות שיטת חיפוש בינארי ("אריה במדבר"), זאת אומרת נעדכן את L בכל פעם להיות L/2 כל עוד לא מוחזר failure (לא הגענו לפתרון האופטימלי), כאשר יוחזר failure זה אומר שהפתרון האופטימלי נמצא בתחום שבין הL הנוכחי ל2L, בשלב זה נעדכן את L להיות באמצע התחום הנ"ל, עד ששוב לא יוחזר failure, נמשיך בצורה זו עד שנגיע לערך L כאשר L-1 החזיר failure ו L+1 לא החזיר failure.

שאלה 6 - UCS (4 נק׳):

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

1. יבש (1 נק׳): עבור אילו בעיות חיפוש אלגוריתם UCS ואלגוריתם BFS יפעלו באותו האופן? הסבירו.

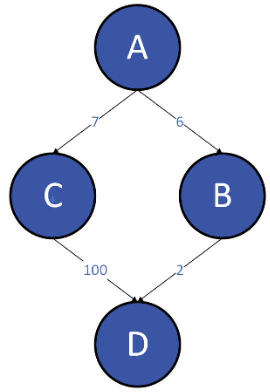
אלגוריתם UCS יפעל באותו אופן כמו אלגוריתם BFS בבעיות חיפוש בהן עלות אחידה לכל הקשתות, מאחר וBFS מבצע חיפוש לרוחב בצורה סיסטמתית על העומקים כאשר הpriority של כל פיתוח נקבע לפי עומק הנוד, וכך מחזיר את הפתרון הרדוד ביותר, בעוד UCS מבצע חיפוש בצורה סיסטמתית על העלות המצטברת g, כאשר הpriority של כל פיתוח נקבע לפי g, ולכן כאשר עלות כל קשת הינה אחידה, העלות המצטברת g תעלה בצורה מונוטונית כתלות בעומק בלבד, וכך האלגוריתמים יפעלו באותו אופן כי הpriority של כל פיתוח יהיה זהה.

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

בבעיית החיפוש שלנו האלגוריתם הינו שלם מאחר וקיים חסם תחתון לפונקציית המחיר $\delta > 0$, המחיר הנמוך ביותר בבעיה שלנו הינו 1, ולכן האלגוריתם תמיד יחזיר פתרון. בנוסף, מאחר והאלגוריתם שלם, בבעיית החיפוש שלנו, הוא גם קביל מכיוון שאלגוריתם UCS מפתח תמיד את הנודים בעלי העלות המצטברת הנמוכה ביותר וכן האלגוריתם מחזיר פתרון כאשר הצומת הראשון ב- open הוא צומת מטרה ולכן תמיד יחזיר קודם את המסלול האופטימלי בעל העלות הכוללת הנמוכה ביותר.

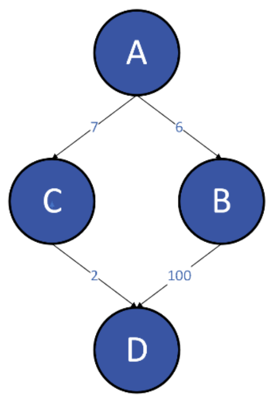
3. יבש (2 נק׳): שאדי טעה במימוש של אלגוריתם UCS ובטעות בדק בעת יצירת הצומת האם היא צומת מטרה במקום בפיתוח שלה. הביאו דוגמה לגרף חיפוש שעבורו שאדי יחזיר בכל זאת את המסלול הקל ביותר, ודוגמה לגרף חיפוש שעבורו שאדי לא יחזיר את המסלול הקל ביותר. עבור כל דוגמה הסבירו מה המסלול והעלות ש-UCS השגוי החזיר, ומה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

במקרה זה שאדי יחזיר את המסלול הקל ביותר כאשר העלות המצטברת g , בזמן פיתוח צומת האב של צומת המטרה, הינה הנמוכה ביותר מבין כל המסלולים שפותחו, לדוגמא בגרף החיפוש הבא:



המסלול האופטימלי במקרה הזה הינו A->B->D וכן גם האלגוריתם השגוי במקרה הזה יחזיר את המסלול הזה והעלות הכוללת תהיה 8, מאחר והאלגוריתם יעצור ברגע שיפתח את B והצומת D תיווצר, לעומת זאת האלגוריתם הנכון ייעצר רק לאחר שייפתח את C ויעביר את צומת המטרה D לclose, אבל יחזיר את אותו המסלול ואותה עלות כוללת מאחר וזהו המסלול האופטימלי.

לעומת זאת שאדי לא יחזיר את המסלול הקל ביותר כאשר תנאי זה לא יתקיים לדוגמא בגרף החיפוש הבא:



המסלול האופטימלי במקרה הזה הינו A->C->D עם עלות כוללת 9, אבל האלגוריתם השגוי במקרה הזה לא יחזיר את המסלול הזה, אלא את המסלול A->B->D והעלות הכוללת תהיה 106, מאחר והאלגוריתם יעצור ברגע שיפתח את B והצומת D תיווצר, לעומת זאת האלגוריתם הנכון ייעצר רק לאחר שייפתח את C ויעדכן את צומת המטרה D לעלות המצטברת הנמוכה ביותר ויעבירו לclose, ולכן יחזיר את המסלול האופטימלי.

שאלה 7 - יוריסטיקות (8 נק׳):

יהי מרחב חיפוש (S,O,I,G) , נסתכל על בעיית הניווט לכדור דרקון יחיד. . המטרה היא למצוא מסלול זול ביותר מהמוצא I ליעד יחיד G . פונק׳ העלות מוגדרת כאורך הכביש המחבר בין שתי נקודות. ניתן להניח כי העולם שטוח . מלבד זאת, לא ניתן להניח דבר נוסף על מרחב החיפוש.

הגדרה : יוריסטיקה h היא ε-קבילה אם קיים $\varepsilon \geq 1$ כך שלכל מצב $s \in S$ מתקיים $h(s) \leq \varepsilon \times h^*(s)$.
נזכיר כי $h^*(s)$ הינה פונקציית המחיר המסלול האופטימאלי מ-s לצומת היעד .

עבור כל אחת מהיוריסטיקות הבאות קבעו האם קיים $\varepsilon \geq 1$ כך שהיוריסטיקה תהיה ε-קבילה . אם כן מצאו את ה-ε ההדוק ביותר המקיים את זאת. נמקו היטב .

1. יבש (1 נק'): מרחק מנהטן : $h_{MD}(p) = |P-G|_1 = |G_x - P_x| + |G_y - P_y|$

נראה כי היוריסטיקה קבילה עם $\sqrt{2}$
עבור נקודה שנמצאת במרחק (x,x)

מהנקודה $(0,0)$ המרחק הקצר ביותר הוא מעבר על אלכסון במישור אוקלידי כלומר $\sqrt{2x^2}$

ומרחק מנהטן במקרה הזה הינו $2x$ לכן היחס המתקבל הוא האפסילון ההדוק ביותר $\sqrt{2}$ נשים לב שהדבר מתקיים בפרט עבור כל נקודה אחרת במרחב ולכן זהו האפסילון ההדוק ביותר.

2. יבש (1 נק'): $h(p) = |P-G| = \min\{G_x - P_x \ , \ G_y - P_y\}$

נראה כי היוריסטיקה קבילה עם $\epsilon = 1$
עבור צומת כללי בעולם שלנו נניח בה"כ כי $\min\{G_x - P_x \ , \ G_y - P_y\} = G_x - P_x$
כעת בהנחה שהעולם שטוח והמרחק הכי קצר בין שתי נקודות הוא המרחק האוקלידי ביניהן נראה כי
$G_x - P_x = \sqrt{|G_x - P_x|^2} \leq \sqrt{(|G_x - P_x|^2 + |G_y - P_y|^2)} = h_{eculidian}$
הדבר מתקיים עבור אפסילון ששוה ל1 לכן זהו החסם הכי הדוק שיש.

3. יבש (1 נק'): L^3 : $h(p) = |P-G|_3 = \sqrt[3]{|G_x - P_x|^3 + |G_y - P_y|^3}$

נראה כי היוריסטיקה קבילה עם $\epsilon = 1$
בשביל להראות זאת נשתמש בייצוג פולארי וכן בזהויות טריגונומטריות
נסמן $G_x - P_x = r \cdot \cos(\theta) \, , G_y - P_y = r \cdot \sin(\theta)$

ולפי זהויות טריגו $\sin 3\theta = 3\sin\theta - 4\sin^3\theta$ | $\cos 3\theta = 4\cos^3\theta - 3\cos\theta$ נקבל כי

$$\sqrt[3]{|G_x - P_x|^3 + |G_y - P_y|^3} = r \cdot \sqrt[3]{\cos^3(\theta) + \sin^3(\theta)} = r \cdot \sqrt[3]{\frac{1}{4}(3(\cos(\theta) + \sin(\theta) + \cos(3\theta) - \sin(3\theta))} < r \cdot \sqrt[3]{\frac{4}{4}}$$
$$\leq r\sqrt{cos(\theta)^2 + \sin(\theta)^2}$$

גורר כי אפסילון ששוה ל1 הוא החסם ההדוק ביותר שיש

4. יבש (1 נק'): נתונות יוריסטיקות h_1 , h_2 שהן ε_1 , ε_2 קבילות בהתאמה וכי ε_1 , ε_2 הם האפסילונים ההדוקים ביותר.
הראו כי $h_3 = h_1 + h_2$ היא ε_3 – קבילה , מצאו את ε_3 ההדוק ביותר והוכיחו .

לפי הנתונים מתקיים
$h_3(s) = h_1(s) + h_2(s) \leq \epsilon_1 \cdot h^*(s) + \epsilon_2 \cdot h^*(s)$
$\epsilon_3 = \epsilon_1 + \epsilon_2$ היא החסם וגם ההדוקה ביותר .

נגדיר יוריסטיקה חדשה :
- D היא קבוצת כדורי הדרקון , $D = \{d1, d2\}$.

$$h_{MSAP}(s) = \min\{h_{Manhatan}(s,g) | g \in G \cup D\}$$

הערה: בנוסחת המרחק מתייחסים למיקום של צומת.
שימו לב שבמקרה זה אנחנו לוקחים את המינימום על פני כל צמתי היעד.

5. יבש (1 נק'): האם היוריסטיקה h_{MSAP} קבילה על כל לוח? אם כן הסבר, אם לא הבא קודמה נגדית.

כן היוריסטיקה קבילה על כל לוח. היוריסטיקה תמיד תבצע הערכת חסר עבור כל צומת במסלול. נוכל לחשוב על צומת במסלול כעל צומת בדרך לדרגון בול 1 או בדרך לדרגון בול 2 או בדרך לצומת המטרה. המרחק הקטן ביותר שנוכל לבצע בדרך למסלול שלה הוא מינימום המרחקים של הצמתים הללו מן הצומת הנתונה – שכן כל צעד עולה לנו מינמום של 1.

לכן היוריסטיקה תמיד תבצע הערכת חסר והיא תמיד קבילה.

6. יבש (1 נק׳): האם היוריסטיקה h_{MSAP} עקבית על כל לוח? אם כן הסבר, אם לא הבא דוגמה נגדית.(לחשוב אם היא עקבית ולתקן בהתאם)

כן, היוריסטיקה עקבית. ההפרש ביוריסטיקות בין שני צמתים סמוכים הוא לכל היותר 1. נשים לב שגם המחיר בלוח כפי שהוגדר הוא לכל הפחות 1. לכן הדרישה לעקביות מתקיימת עבור כל זוג צמתים סמוכים
הדרישה:

$$\forall s \in S, s' \in succ\ s : h\ s - h\ s' \leq cost\ s, s'$$

נגדיר יוריסטיקה חדשה :

- D היא קבוצת כדורי הדרקון, $D = \{d1, d2\}$.

$$h_{new}(s) = \max\{h_{Manhatan}(s, g) | g \in G \cup D\}$$

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

היוריסטיקה לא קבילה נראה דוגמה נגדית
יוריסטיקה קבילה מקיימת

$$\forall s \in S: 0 \leq h\ s \leq h*s$$

נסתכל על הלוח הבא, ערכי היוריסטיקה מסומנים באדום

L 4	L 4 עבור הצומת הנ"ל מחיר המסלול למצב המטרה הוא בדיוק 3 לכן הערכה של היוריסטיקה החדשה עושה הערכת יתר ולכן לא קבילה	S 5
L 3	D 3	L 4
L 2	D 2	L 3
G 3	G 2	L 2

8. יבש (1 נק׳): האם היוריסטיקה h_{new} עקבית על כל לוח? אם כן הסבר, אם לא הבא דוגמה נגדית.

כפי שהראנו בסעיף הקודם היוריסטיקה לא קבילה ולכן גם לא עקבית על כל לוח.

שאלה 8 – Greedy Best First Search (3 נק׳):

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

1. יבש (1 נק׳): האם האלגוריתם שלם? האם הוא קביל?

האלגוריתם שלם מאחר ומרחב המצבים בלוח הינו סופי וקשיר, ולכן לא משנה איזו יוריסטיקה נבחרה, בסוף האלגוריתם יחזיר פתרון.
האלגוריתם לא קביל מכיוון שגם בהינתן יוריסטיקה קבילה ועקבית, האלגוריתם לא בהכרח יחזיר את המסלול האופטימלי מאחר ולא מתחשב בעלות כלל.

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

יתרון אחד של Greedy Best first Search לעומת Beam Search הוא איכות הפתרון, מכיוון שאלגוריתם Beam Search זורק חלק מהצמתים ברשימת open שעוד לא פותחו, ייתכן כי נפספס מסלולים יותר קלים שבאלגוריתם Greedy Best first Search אין אופציה לפספס כי מפתחים את כל הצמתים בopen עד להגעה למטרה.

חיסרון אחד של Greedy Best first Search לעומת Beam Search הוא כמות הזיכרון, מאחר וBeam Search מתחזק רשימת open מוגבלת, כך גם גודל הזיכרון שתופס מוגבל בניגוד לGreedy Best first Search שמתחזק רשימת open מלאה.

שאלה 9 – W-A* (2 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_{MSAP}.

2. (יבש 2 נק׳) בהינתן $1 \leq w_2 > w_1$, נסמן את המסלולים המחוזרים על ידי W-A* תחת הפורמולציה $f = g + w \cdot h$ בp_1, p_2 עבור w_1, w_2 בהתאמה. אזי $cost(p_1) < cost(p_2)$ עבור:

   a. יוריסטיקה קבילה h. אם כן הסבירו. אם לא, ספקו דוגמה נגדית.

לא , נסתכל על דוגמה נגדית של יוריסטיקה שמחזירה אפס לכל צומת. היוריסטיקה קבילה מפני שהמצב שקול ל*UCS* שלמדנו כי הוא קביל. מצד שני מתקיים לכל צומת

$$f_1(s) = g(s) + w_1 \cdot h(s)_{=0} = g(s) + w_2 \cdot h(s)_{=0} = f_2(s)$$

   b. יוריסטיקה כללית (לא בהכרח קבילה) h. אם כן הסבירו. אם לא, ספקו דוגמה נגדית.

באופן דומה לסעיף א' , היוריסטיקה שם היא יוריסטיקה כללית (ובפרט גם קבילה) ולכן מייצגת דומה נגדית גם לסעיף זה.

שאלה 10 – IDA* (2 נק׳):

1. יבש (1 נק׳): ספקו יתרון וחסרון של IDA* ביחס לA*. באילו מקרים הייתם מעדיפים להשתמש בכל אחד מהם?

יתרון של IDA* ביחס לA* הינו צריכת הזיכרון, חיסרון הינו שIDA* מבצע חיפוש בעץ, זאת אומרת מפתח מצבים חוזרים מבלי לדעת שביקר בהם בעבר, לעומת A* שמתחזק רשימת close ולכן מבצע חיפוש בגרף ללא פיתוח מצבים חוזרים כאשר אין שיפור.
אעדיף להשתמש בIDA* כאשר מקדם הסיעוף מאוד גבוה והזיכרון מוגבל, בנוסף כאשר המחיר על הקשתות אחיד והיוריסטיקה מושלמת, אעדיף להשתמש בIDA* מאחר ויימצא פתרון מהיר לאחר איטרציה אחת כאשר A* ייצטרך לפתח את כל הצמתים בהנתן שובר שוויון רנדומלי מאחר וכל הצמתים יהיו בעל ערכי f זהים. לעומת זאת אעדיף להשתמש בA* כאשר נתונה לי יוריסטיקה עקבית ולא מושלמת וכאשר המחיר אינו אחיד, ארצה להמנע מפיתוחים של נודים מיותרים וכן מאיטרציות מיותרות, וכך לחסוך בזמן ריצה.

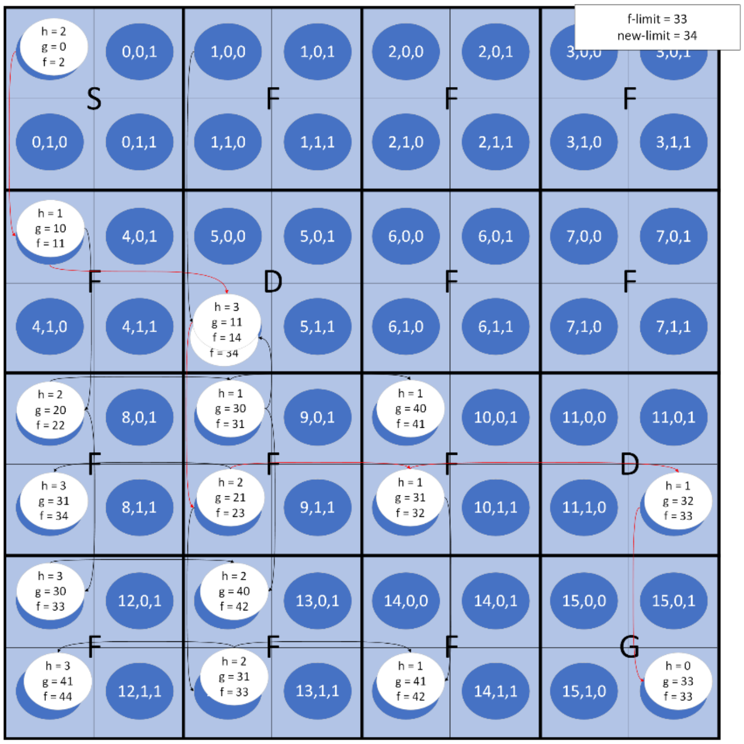
2. יבש (1 נק׳): ספק המחשה שלב אחר שלב של אלגוריתם IDA* על הלוח (4x4) שמופיע במחברת, המראה כיצד החיפוש מתקדם באמצעות העמקה איטרטיבית ?

ביצענו הרצת IDA* על הלוח הנ"ל כפי שהוסבר בתרגול - בכל איטרציה נקבע f-limit אשר עד אליו מבצעים DFS-f, זאת אומרת כאשר DFS-f פוגש צומת v, נחשב את f(v), אם $f(v) \leq f\_limit$: המשך להעמיק, אחרת: עצור העמקה.
באיטרציה הראשונה הf-limit נקבע להיות :

$$f\_limit(I) = f(I) = h(I) = 2$$

ערך f-limit באיטרציה הבאה (new-limit) נקבע להיות ערך ה f -הקטן ביותר שפגשנו באיטרציה הקודמת, מבין אלו שגדולים מ-f-limit.

נציג את הלוח באיטרציה האחרונה של הרצת אלגוריתם זה:



מסלול הפתרון שהוחזר מהאלגוריתם הינו המסלול המסומן בחיצים האדומים. כפי שניתן לראות האלגוריתם החזיר את המסלול האופטימלי.

שאלה 11 – A* epsilon (6 נק׳):

1. **רטוב**: ממשו את החלקים החסרים באלג׳ W-A* בקובץ ע״פ ההנחיות המופיעות שם. עליכם להשתמש ביוריסטיקה h_{MSAP}.

2. יבש (2 נק׳): תנו יתרון וחיסרון של A*-epsilon לעומת A*.

   תחת ההנחה שפונק' המחיר חסומה מלמטה וחיובית והיוריסטיקה קבילה
   יתרון של A*-epsilon: יכול להביא לנו פתרון מהיר יותר מA*
   חסרון של A*-epsilon: הפתרון שלו יכול להיות אופטימלי עד כדי 'מחיר' שנשלם באופטימליות.

3. יבש (3 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.

   בבעיית כדורי הדרקון נציע את היוריסטיקה להוצאה מFOCAL את h_{MSAP}
   לפי מה שהראינו בשאלה 7.5 היוריסטיקה קבילה ומוצאת לנו את אותו הפתרון שמצאנו לפי g. למרות זאת היוריסטיקה מפתחת יותר צמתים. היוריסטיקה מנסה "ללכת בקו ישר" אל כדורי הדרקון דבר שמוביל להעדפת מסלולים שמפתחים יותר צמתים בדרך
   נשווה את התוצאות בין היוריסטיקה הזו לבין g(v) על מפת 8 על 8 הנתונה לנו

h_{MSAP}

```
Total_cost: 103.0
Expanded: 88
Actions: [0, 0, 0, 0, 0, 0, 1, 1, 1, 2, 1, 1, 0, 1, 1, 0]
```

g(v)

```
Total_cost: 103.0
Expanded: 81
Actions: [0, 0, 0, 0, 0, 0, 1, 1, 1, 2, 1, 1, 0, 1, 1, 0]
```

4. יבש (1נק׳): אם נגדיר שאפסילון שווה לאינסוף איך תהיה ההתנהגות של האלגוריתם עם סביבת כדורי הדרקון.

הגדרה של אפסילון לאינסוף יכולה ליצור מסלולים עם מעבר בתוך חורים. שכן f_value של חורים שווה לאינסוף ולכן הם יוכלו להכנס לFOCAL. מכיוון שההוצאה מFOCAL נעשית על פי יוריסטיקה נוספת יכול להיות שמה שנוציא הוא חור ולא נמצא מסלול כלל לבעיה.

שאלה 12 – Benchmarking (2 נק׳):

בשאלה זאת נשווה בין אלגוריתמי חיפוש שונים על בעיות שונות. הריצו את החלק הרלוונטי במחברת ותיראו שנוצר קובץ csv. (ניתן לפתוח עם Excel).

1. **רטוב**: הריצו את החלק הרלוונטי במחברת ותיראו שנוצר קובץ csv. (ניתן לפתוח עם Excel).

2. יבש (2 נק׳): הסבירו את התוצאות. האם הן תואמות לציפיות שלכם? האם התוצאות היו משתנות עם יוריסטיקה יותר מיודעת? נתחו והסבירו את התוצאות במונחים של מספר פיתוחים, מסלול מוחזר ומחיר הפתרון. שימו לב שבסעיף זה אין תשובה נכונה או לא נכונה אבל נדרש ממכם לספק הסבר מפורט ומבוסס.

נציג תחילה את התוצאות שהתקבלו:

Map	BFS-G		WA* (0.5)		WA* (0.7)		WA* (0.9)	
	Cost	Expanded	Cost	Expanded	Cost	Expanded	Cost	Expanded
12x12	140	445	118	224	118	200	118	240
15x15	215	858	178	651	178	604	195	707
20x20	203	1045	188	684	188	587	188	1002

ניתן לראות בטבלה זו כי באלגוריתם BFS-G, התקבלו המסלולים בעלי המחיר הגבוה ביותר, מאחר ואלגוריתם זה אינו מתייחס כלל לעלות הקשתות (עלות דריכה על כל משבצת במקרה שלנו), בדרך אל הפתרון אלא רק לעומק הפתרון, ולכן מחזיר את המסלול האפשרי הרדוד ביותר (במקרה שלנו בעל מספר הצעדים הנמוך ביותר), ללא תלות בעלות כל משבצת. בנוסף, נשים לב כי באלגוריתם זה התקבל מספר הפיתוחים הגבוה ביותר, מאחר ואלגוריתם זה אינו מיודע כלל ופועל ללא כל יוריסטיקה, ולמעשה באלגוריתם זה הפועל בשיטת החיפוש לרוחב, הדורשת פיתוח של כל הצמתים בעומק d-1 לפני הצמתים בעומק הפתרון d, אשר מובילה לכך שאלגוריתם זה צריך לפתח את מספר הצמתים הגבוה ביותר ולכן הינו האיטי ביותר, כפי שציפינו.

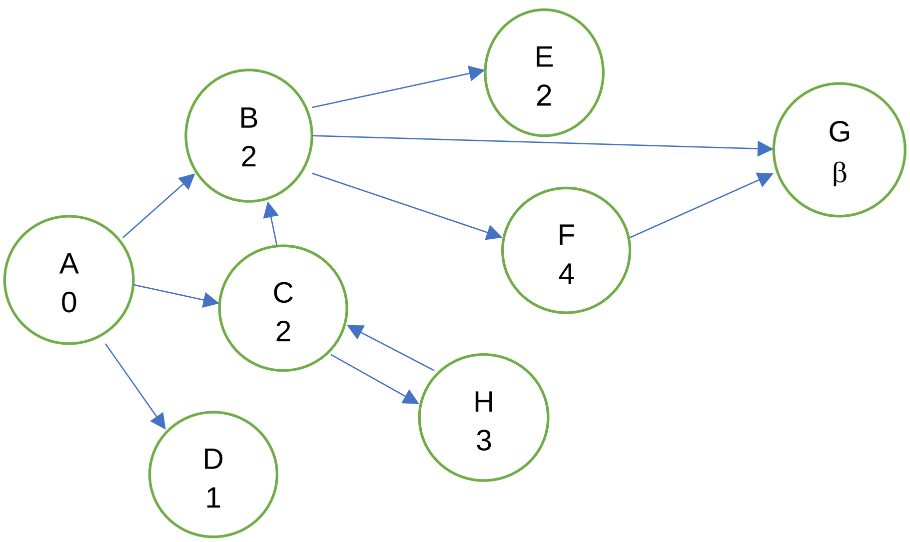
בנוסף, ניתן לראות בטבלה זו כי באלגוריתם WA* עם משקל 0.5 התקבלו המסלולים בעלי העלות הנמוכה ביותר, מאחר ואלגוריתם זה (עם משקל 0.5) הינו קביל ומחזיר את הפתרון האופטימלי תחת יוריסטיקה קבילה, וכפי שהראינו בסעיף 7.5, היוריסטיקה h-msap שבה השתמשנו הינה קבילה.
נשים לב כי גם במשקל 0.7 התקבלו המסלולים בעלי אותה עלות (הנמוכה ביותר), למרות שבמשקל זה האלגוריתם כבר אינו קביל, ולכן היינו מצפים שתהיה ירידה באיכות הפתרון ככל שהמשקל עולה מעל ל 0.5, נשים לב כי אמנם עלות המסלולים שנבדקו כאן הינם זהים לאלו שהתקבלו במשקל 0.5, אך המסלולים עצמם הינם זהים לחלוטין, ולכן ניתן להניח כי העלאת המשקל לא גרעה מאיכות הפתרון במקרים אלו רק בגלל אופן סידור הלוחות הנ"ל ואילו הלוחות היו מסודרים בצורה שונה היה מתקבל מסלול יקר יותר.
כמו כן, ניתן לראות שכבר כאשר המשקל עלה ל-0.9, בלוח אחד מתוך השלושה (15x15), כבר התקבל מסלול יקר יותר, כפי שהיינו מצפים לראות, מאחר וככל שהמשקל w באלגוריתם WA* גדול יותר מ0.5, כך יש פחות משקל לעלות המסלול ולכן נצפה לקבל פתרון באיכות יותר נמוכה (יקר יותר). בשני הלוחות האחרים התקבלו מסלולים בעלי עלות זהה לאלו שהתקבלו במשקלים 0.5,0.7 , אך המסלולים עצמם אינם זהים לחלוטין, ולכן, בדומה למשקל 0.7, ניתן להניח כי העלאת המשקל לא גרעה מאיכות הפתרון במקרים אלו רק בגלל אופן סידור הלוחות הנ"ל ואילו הלוחות היו מסודרים בצורה שונה היה מתקבל מסלול יקר יותר.

נשווה כעת את מספר הפיתוחים שהתקבלו בין שלושת אלגוריתמי WA* בעלי המשקלים השונים.
נשים לב כי כאשר משווים את מספר הפיתוחים שהתקבלו כאשר w=0.7, לעומת מספר הפיתוחים שהתקבלו כאשר w=0.5, אכן ישנה ירידה בכמות הפיתוחים, כפי שצפינו, מאחר שככל שהמשקל w גדול יותר, כך החיפוש הינו מיודע יותר ולכן מהיר יותר, ובעל מספר הפיתוחים קטן יותר.
לעומת זאת, כאשר אנו משווים גם את מספר הפיתוחים שהתקבלו כאשר w=0.9, נראה כי דווקא באלגוריתם זה התקבל מספר הפיתוחים הגדול ביותר מבין שלושתם, בכל המפות, וזאת בניגוד לציפיתנו. ניתן להניח כי דבר זה קרה מאחר והיוריסטיקה בה השתמשנו אינה מספיק מיודעת, מאחר ואינה לוקחת בחשבון כלל את עלות המשבצות, אלא רק את המרחק (מנהטן) למטרה הקרובה ביותר. לכן כאשר ניתן משקל גבוה יותר ליוריסטיקה זו, דבר זה אמנם יכול להביא למסלול קצר יותר, אך מאחר והלוח הינו ריבועי ולא ניתן לנוע באלכסון בכל צעד אלא רק למטה/למעלה/ימינה/למטה, נוצרת סימטריה בערך היוריסטיקה בכל צעד, התלויה בעומק החיפוש, וכך בדומה לאלגוריתם BFS-G, במשקל גבוה מידי עם יוריסטיקה זו, החיפוש הופך ליותר רוחבי, וכן ליותר איטי ולכן מספר הפיתוחים גדל.

אילו היוריסטיקה הייתה יותר מיודעת, התוצאות של אלגוריתמי *WA היו כנראה משתנות במספר הפיתוחים, סביר להניח כי ככל שהיוריסטיקה יותר מיודעת כך האלגוריתם מוצא פתרון אופטימלי מהר יותר ומספר הפיתוחים קטן, וכן ככל שהמשקל w היה גדול יותר כך החיפוש היה מהיר יותר. בנוסף, ייתכן שעם יוריסטיקה יותר מיודעת לא היינו רואים הבדל בעלות המסלול במשקלי w גדולים מ0.5.
יש לציין כי גם עם יוריסטיקה יותר מיודעת, *WA עם משקל 0.5 יחזיר את אותו פתרון אופטימלי כפי שהחזיר עם היוריסטיקה h-msap מאחר ויוריסטיקה זו קבילה.

שאלה 13 – Local Search (5 נק׳):

בהינתן מרחב המצבים הבא, כאשר a הינו המצב ההתחלתי, $U: S \to \mathbb{R}^+$ הינה פונקציית ערך והערך עבור כל מצב מצוין בצומת. המטרה שלנו היא למצוא מצב שממקסם את ערך U.



נשתמש באלגוריתם <u>Stochastic Hill Climbing</u>.
כמו כן ידוע כי $\boldsymbol{\beta > 3}$.

1. יבש (1 נק׳): מה ההסתברויות למעבר מהצב ההתחלתי לכל אחד מהמצבים b, c, d. רשמו את $p(d|a).p(b|a), p(c|a)$

$$p(d|a) = \frac{U(d)}{\sum U(neigbhors\ of\ a)} = \frac{1}{5}$$
$$p(b|a) = \frac{U(d)}{\sum U(neigbhors\ of\ a)} = \frac{2}{5}$$
$$p(c|a) = \frac{U(d)}{\sum U(neigbhors\ of\ a)} = \frac{2}{5}$$

2. יבש (1 נק׳): מה הוא מספר הצעדים המקסימלי שהאלגוריתם יכול לבצע? צעד מוגדר כמעבר בין מצבים.

מספר הצעדים המקסימלי שהאלגוריתם יכול לבצע הוא 3. המעברים יהיו מA לB אחכ מB לF ולבסוף מF לG. זאת בהנחה שמצב G מגדיל את ערך U.

3. יבש (1 נק׳): בהיתן שבצעד הראשון האלגוריתם עבר למצב c. האם האלגוריתם יתכנס למקסימום הגלובלי?

תחת ההנחה הזו האלגוריתם לא יתכנס למקסימום הגלובלי. נשים לב שממצב זה אפשר לעבור אך ורק למצב H
לעומת זאת מצב F עם ערך U גדול מכל המצבים במסלול הזה ולכן בהנחה המצוינת האלגוריתם לא יתכנס למקסימום הגלובלי

4. יבש (1 נק׳): מה ההסתברות שהאלגוריתם יתכנס לפתרון לא אופטימלי (שאינו מקסימום גלובלי)?

יש שני מקרים :

מקרה א'

$$2 < U(g) < 4$$

$$p(d) + p(c) + p(b) \cdot \big(p(g)\big) = \frac{1}{5} + \frac{2}{5} + \frac{2}{5} \cdot \frac{\beta - 2}{2 + \beta}$$

מקרה ב'

$$4 \leq U(g) \; or \; U(g) \leq 2$$

$$p(d) + p(c) = \frac{3}{5} = 0.6 = 60\%$$

5. יבש (1 נק׳): עבור אילו ערכים של β ההסתברות להגיע מהמצב ההתחלתי למקסימום הגלובלי תוך <u>בדיוק 3 צעדים</u> גדול מ $\frac{1}{5}$?

הדבר לא מתקיים עבור אף בטא.

נדרוש

$$\frac{2}{5} \cdot \frac{2}{2 + \beta} > \frac{1}{5} \rightarrow \beta < 4$$

סתירה לכך שדרשנו שצומת G מקס גלובלי

הוראות הגשה:

עליכם להגיש קובץ יחד בשם AI1_<id1>_<id2>.zip (בלי הסוגריים המשולשים) המכיל:

1. קובץ בשם AI1_<id1>_<id2>.pdf שמכיל את התשובות לחלק היבש.
2. קובץ בשם Algorithms.py המכיל את המימוש לאלגוריתמי החיפוש.