מבוא לבינה מלאכותית

תרגיל בית 1

שם סטודנט: אבראהים סעיד

תעודת זהות: 209112036

שם סטודנט: עומר מחאמיד

תעודת זהות: 308198134

שאלה 1:

2. תחילה נרצה להגדיר את מרחב החיפוש כפי שנלמד בתרגול. נגדיר:

$S = X^d$ = מרחב המצבים, אשר כל מצב בו מתואר על ידי וקטור של d פיצ׳רים.

O = מרחב האופרטורים

I = המצב ההתחלתי

G = קבוצת מצבי המטרה

הגדירו את (S, O, I, G) עבור סביבת הקמפוס. (1 נק׳)

נגדיר את מרחב החיפוש $(S,\ I,\ O,\ G)$ באופן הבא:

$S\ =\ \{0,\ 1,\ \ \cdots\ ,\ 63\}$
$O\ =\ \{DOWN,\ UP,\ RIGHT,\ LEFT\}$
$I\ =\ \{0\}$
$G\ =\ \{63\}$

כאשר בקבוצת המצבים נגדיר לכל משבצת נגדיר מספר בתחום $[0.\ 63]$ לפי המשבצת החל מצד שמאל לעיל, המצב ההתחלתי הוא המצב של המשבצת הראשונה, ומצב המטרה הוא מצב המשבצת האחרונה.

---

3. מה גודל מרחב המצבים S? הסבירו. (1 נק׳)

לפי מה שהגדרנו בסעיף הקודם עבור מרחב המצבים S, הגודל שלו הוא: 64.

---

4. מה תחזיר לנו הפונקציה Domain על האופרטור DOWN (אופרטור 0) (1 נק׳).

$DOMAIN(DOWN)\ =\ \{s\ \in\ S\ |\ s$ הוא אינו חור$\}$

---

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

הפעלת הפונקצייה $Succ$ על המצב ההתחלתי 0 תחזיר לנו את הבא:

$UP(0)\ =\ LEFT(0)\ =\ 0\ \Longrightarrow$ כי נישאר באותו מצב
$RIGHT(0)\ =\ 1\ \Longrightarrow$ כי הולכים משבצת אחת ימינה
$DOWN(0)\ =\ 8\ \Longrightarrow$ כי מגיעים למשבצת השמינית

---

6. האם קיימים מעגלים במרחב החיפוש שלנו? (1 נק׳)

כן, ניקח לדוגמה את המסלול הבא:

$$0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow 8 \rightarrow 0$$

שנגיע אליו ע"י הפעלת $Right$ שמונה פעמים, ואז נפעיל UP ונחזור למצב התחלתי 0.

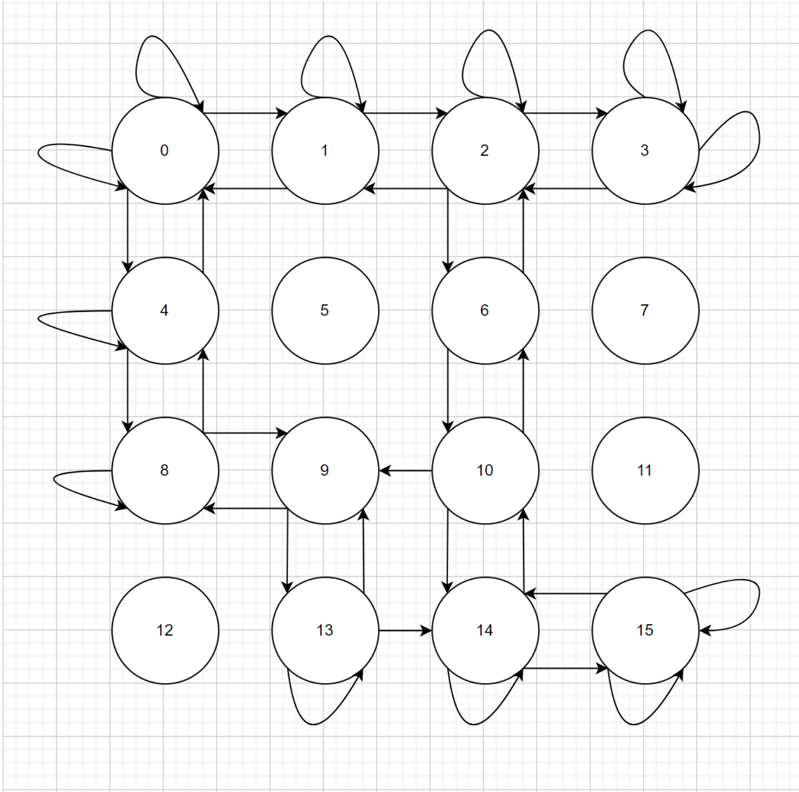
---

7. מה הוא מקדם הסיעוף בבעיית הניווט בקמפוס? (1 נק׳)

מקדם הסיעוף בבעיית הניווט בקמפוס הוא 4, כי מכל צומת בבעיה ניתן להגיע רק ל- 4 צמתים לכל היותר ע"י הפעלת אחד מארבעת האופרטורים $\{DOWN,\ UP,\ RIGHT,\ LEFT\}$.

---

8. עבור המפה "4x4" שמופיעה במחברת, ציירו את גרף המצבים. (1 נק׳)



---

9. במקרה הגרוע ביותר, כמה פעולות ידרשו לסוכן Random Agent להגיע למצב הסופי? ובמקרה הטוב ביותר? (1 נק')

במקרה הגרוע ביותר מספר הפעולות במקרה הגרוע ביותר הוא אינסוף, כי לא נגיע למצב סופי, והסוכן ייתקע בלולאה או שהוא יגיע למצב חור.
במקרה הטוב ביותר, מספר הפעולות הוא 9, אם נבחר את הפעולות המתאימות לכך.
3 פעולות $DOWN$, ואז פעולה אחת $RIGHT$ שמעבירה אותנו דרך הפורטל P, ואז 2 פעולות $RIGHT$, ובסוף 3 פעולות $DOWN$ ונגיע למצב הסופי, וזהו מסלול קצר ביותר מהמצב ההתחלתי למצב המטרה.

---

10. עבור מפה כללית בסביבת הקמפוס ,בה יכולים להיות מספר מצבי מטרה (לדוגמה במקרה שבו איימי תתרצה גם אם תגיע לחומוס בבית הסטודנט), האם המסלול הזול ביותר (מבחינת עלות המסלול) הוא גם המסלול שמגיע למצב המטרה הקרוב ביותר למצב ההתחלתי (במונחים של Manhattan Distance, כפי שנלמד בכיתה)? אם כן, הוכיחו. אם לא, ספקו דוגמה נגדית. (1 נק')

דוגמא נגדית:

S	F	F	G_2
L	H	H	F
L	H	H	F
L	L	L	G_1

מרחק $Manhattan$ של המסלול מהמצב ההתחלתי אל מצב המטרה G_1 הוא 6, והעלות שלו היא גם 6.
מרחק $Manhattan$ של המסלול מהמצב ההתחלתי אל מצב המטרה G_1 הוא 3, והעלות שלו היא יותר מ- 10 כי חייבים לעבור דרך F.

שאלה 2:

1. עבור בעיית הניווט הקמפוס עם מפה NxN, האם האלגוריתם שלם? האם הוא קביל? (1 נק')

האלגוריתם שלם כי הוא סורק את כל המסלולים האפשריים לכן אם קיים פתרון אז הוא כן מחזיר אותו.
אבל אלגוריתם זה אינו קביל, דוגמה:

S	P1	F
A	A	P2
A	A	G

אם נריץ את האלגוריתם על דוגמה זאת לא מקבלים פתרון אופטימלי.

---

2. מה צריך להיות התנאי על גרף החיפוש (לא בהכרח בבעיית הניווט הקמפוס) כך שאלגו' BFS (שרץ על עץ) ואלגו' BFS-G (שרץ על גרף) ייצרו ויפתחו צמתים זהים באותו הסדר? (1 נק')

התנאי הוא שהקשתות שנמצאות בגרף אך לא בעץ יהיו רק קשתות המחברות בין צמתים הנמצאים באותו עומק בעץ.

---

3. נתונה מפה בגודל NxN שלא מכילה portals. הציעו דרך להשתמש באלגוריתם BFS-G כך שיחזיר פתרון אופטימלי (עלות מינימלית) והסבירו בקצרה. (2 נק')

הרעיון הוא לשנות את גרף החיפוש G כך שכל הקשתות בו יהיו בעלות משקל שווה ל-1. כך, הפתרון האופטימלי יהיה גם המסלול הקצר ביותר ו-BFS ימצא אותו. לשם כך, במקום כל קשת בעלת משקל X גדול מ-1, נחליף אותה במסלול באורך X המורכב מקשתות שמשקל כל אחת מהן שווה ל-1. נציין שמשקל הקשת תלוי בצומת שהיא נכנסת אליו ולא בצומת שהיא יוצאת ממנו, והקשת הכבדה ביותר מגיעה ל-F ומשקלה 10. לכן, עבור כל קשת שנחליף במסלול כפי שתואר, נצטרך להוסיף לכל היותר 9 צמתים חדשים, ובסך הכל נצטרך להוסיף לכל היותר E10 צמתים חדשים, שבהתחלה דרגתם שווה ל-0. כל פעם שנבחר צומת, נוסיף צומת חדש ודרגתו תישאר 0. כעת נסביר כיצד בונים את הגרף 'G:

$$V' = V \cup \left\{ n^2 , n^2 + 1 , \ldots , n^2 + 10 \cdot |E| \right\}$$

$$E' = \{ e \in E \mid w(e) = 1 \} \cup E_{new}$$

נגדיר את E_{new}:

- אם משקל הקשת $u \to v$ שווה ל- 2: נסיר את הקשת ונחליף אותה במסלול הזה:

$$p = \left\{ u \to i \to v \mid i \in \left\{ n^2 \cdots n^2 + 10 \cdot |E| \right\} \wedge (d(i) = 0) \right\}$$

- אם משקל הקשת $u \to v$ שווה ל- 3: נסיר את הקשת ונחליף אותה במסלול הזה:

$$p = \left\{ u \to i \to j \to v \mid j,\, i \in \left\{ n^2 \cdots n^2 + 10 \cdot |E| \right\} \wedge d(i) = d(j) = 0 \right\}$$

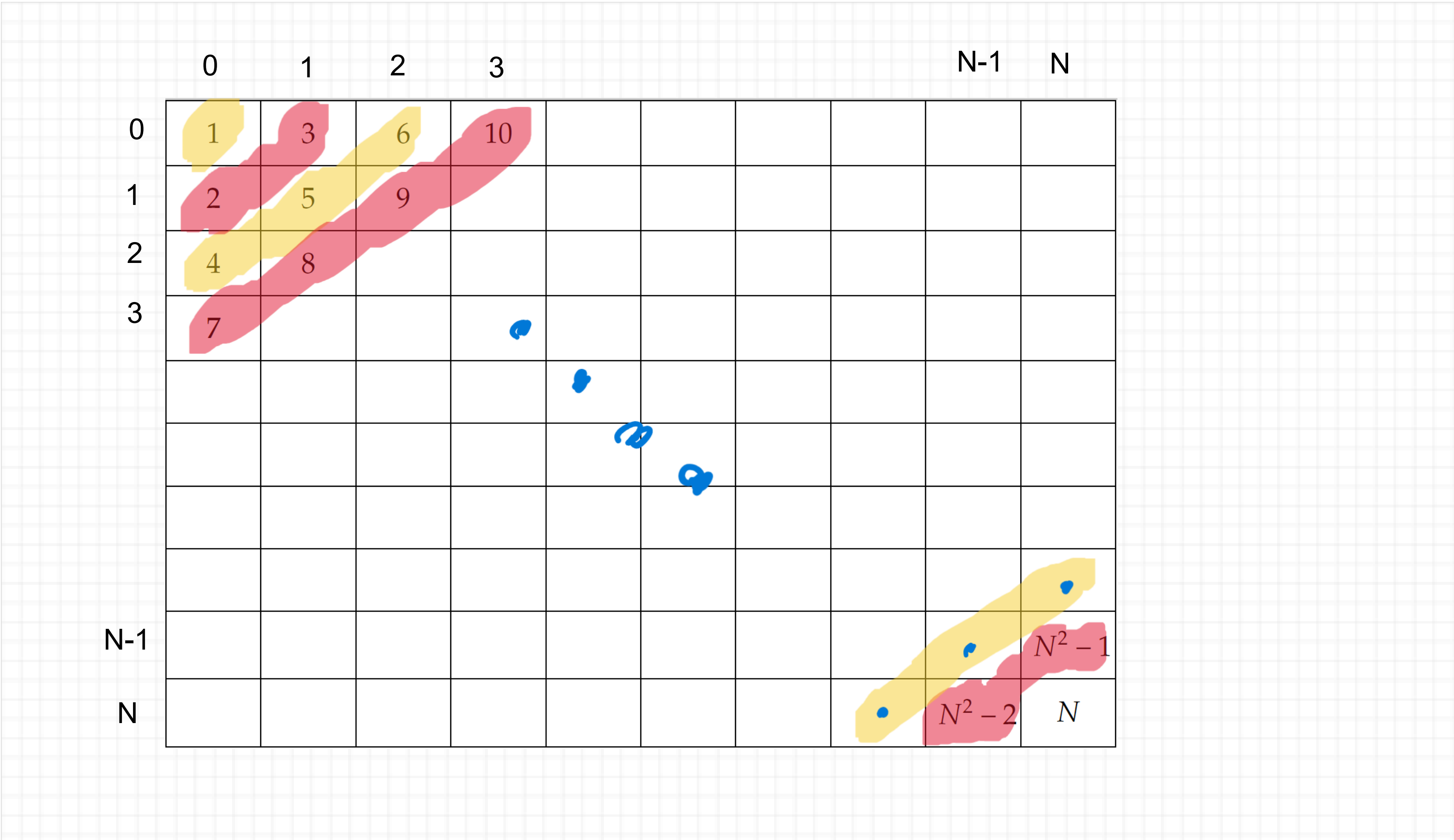
- אם הקשת u->v עם משקל 10: נסיר את הקשת ונחליף אותה במסלול הזה:

$$p = \left\{ u \to i_1 \to i_2 \to \cdots \to i_9 \to v \mid \forall j \in \{1 \cdots 9\} : \left(i_j \left\{ n^2 \cdots n^2 + 10 \cdot |E| \right\} \wedge d(i_j) = 0 \right) \right\}$$

---

4. נתונה מפה בגודל NxN, ללא חורים, ללא Portals, 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. (2 נק')

מכיוון שאין חורים ואין פורטלים, ניתן להגיע לכל הצמתים בארבעה כיוונים. מכיוון שב-BFS עץ החיפוש נבנה בשכבות לפי אורך המסלול הקצר ביותר מבחינת כמות הקשתות, וב-$BFS - G$ צמתים לא מתפתחים פעמיים, הפיתוח יתבצע בצורת אלכסונים כפי שמוצג בציור. כתוצאה מכך, כל הצמתים יפותחו למעט הצומת הסופית והצומת שמעליה, כי נגיע לפתרון לאחר פיתוח צומת מספר $n^2 - 2$. כלומר, יפותחו $n^2 - 2$ צמתים וייווצרו עוד שני צמתים נוספים.



0
1
2
3
N-1
N
0
1
2
3
N-1
N
1
3
6
10
2
5
9
4
8
7
N^2-1
N^2-2
N

שאלה 3:

2. עבור בעיית הניווט הקמפוס עם מפה NxN, האם האלגוריתם שלם? האם הוא קביל? (1 נק')

האלגוריתם שלם כי הגרף הוא סופי ואז תמיד מחזיר פתרון.
אבל הפתרון לא קביל כי הוא עושה סריקה לעומק ואז כאשר הוא מוצא פתרון יחזיר אותו בלי להמשיך לסריקה כל המסלולים.

---

3. עבור בעיית החיפוש בקמפוס, נתונה מפה בגודל NxN. האם אלגוריתם DFS (שרץ על עץ), עבור בעיית הניווט הקמפוס על מפה NxN, היה מוצא פתרון כלשהו? אם כן, מה המסלול שיתקבל? אם לא, כיצד האלגוריתם היה פועל? (1 נק')

לא בהכרח, הוא עלול להיתקע ולהיכנס ללולאה אינסופית. לדוגמה, אם בפעולות הראשונות הוא ינסה לרדת למטה, ואז שוב ינסה לרדת למטה, הוא יישאר במקום. כלומר, הוא יפתח שוב את אותו צומת, ינסה לרדת למטה שוב, יישאר במקום וחוזר חלילה.

---

4. כעת נתונה מפה בגודל NxN, ללא Holes וללא Portals. המפה מכילה $N^2 - 2$ משבצות "רגילות" (F/T/A/L), מצב התחלתי S בפינה השמאלית העליונה שלה ומצב מטרה S בפינה הימנית התחתונה שלה.

1. כמה צמתים יפותחו וייווצרו במהלך חיפוש DFS-G? (2 נק')

האלגוריתם ינסה תחילה לרדת למטה ככל האפשר ויפתח כל צומת בדרך. לכל צומת שנפתח, הוא ייצור את הבנים שלו. כשלא תהיה אפשרות לרדת יותר, הוא יפנה ימינה (אחרי כל פנייה ימינה ינסה שוב לרדת למטה, אך זה לא יהיה אפשרי ולכן ימשיך ללכת ימינה), וימשיך לפתוח כל צומת בדרך עד שיגיע למצב הסופי. לפיכך, יפותחו $2N-2$ צמתים: כל העמודה הראשונה וכל השורה האחרונה, למעט המצב הסופי. בנוסף, ייווצרו כל הצמתים שניתן להגיע אליהם מצמתי העמודה הראשונה (שניתן לרדת למטה לצמתים שכבר פותחו או ללכת ימינה לצמתים שנוצרו). באופן דומה, ייווצרו כל הצמתים שניתן להגיע אליהם מצמתי השורה האחרונה (שניתן ללכת ימינה לצמתים שכבר פותחו, חוץ מצומת המטרה, או לעלות למעלה לצמתים שנוצרו). לכן, בסך הכול ייווצרו עוד $2N-3$ צמתים (לא כולל אלה שפותחו).

2. כמה צמתים יפותחו וייווצרו במהלך חיפוש DFS-G **backtracking**? הסבירו בתשובתכם מהו היתרון בשימוש באלג' זה על פני שימוש באלג' DFS-G מתת הסעיף הקודם. (2 נק')

בדומה לסעיף הקודם, האלגוריתם יפעל באותה צורה, אך ההבדל יהיה באופן יצירת הבנים. במקום לייצר את כל הבנים בפיתוח, נייצר רק את הצומת המעקב. לכן, יפותחו בסך הכל $N-2$ צמתים , וייוצר צומת נוסף אחד בלבד, שהוא המצב הסופי.

---

5. איימי רוצה למצוא מסלול בסביבת הקמפוס עם DFS-L. ידוע כי אורך המסלול הקצר ביותר לצומת מטרה הוא d אך איימי מחליטה להגביל את עצמה לחיפוש בעומק $\frac{d}{2}$ בלבד.

1. עבור מפה כללית בגודל NxN, הציעו שינוי **לבעיית החיפוש** (S, O, I, G) כך שאיימי תוכל למצוא פתרון מבלי להפר את מגבלת העומק. הסבירו למה כעת ניתן למצוא פתרון. (3 נק')

לשנות את מרחב האופרטורים כך שבמקום להתקדם בצעד אחד, נתקדם בשני צעדים. כתוצאה, אם נדרש עומק d כדי למצוא את הפתרון,אז יהיה מספיק לנו עומק $\frac{d}{2}$.

---

2. האם השתנה מקדם הסיעוף? אם כן, מה מקדם הסיעוף החדש b'? רשמו את התשובה כתלות בb (מקדם הסיעוף בבעיה המקורית). (1 נק')

כן, עכשיו יש לנו 12 בנים במקום 4 שזה : $b' = 3 \cdot b$

---

3. מהן סיבוכיות הזמן והמקום החדשים? ענו במונחים של b, d והשוו את התשובה ל-DFS-L רגיל עם עומק d. (1 נק')

עבור מרחק $\frac{d}{2}$ נקבל כי סיבוכיות הזמן של $DFS-L$ היא: $O(3b)^{\frac{d}{2}}$ וסיבוכיות המקום היא: $O(b \cdot d) = O\left(3b \cdot \frac{d}{2}\right)$.

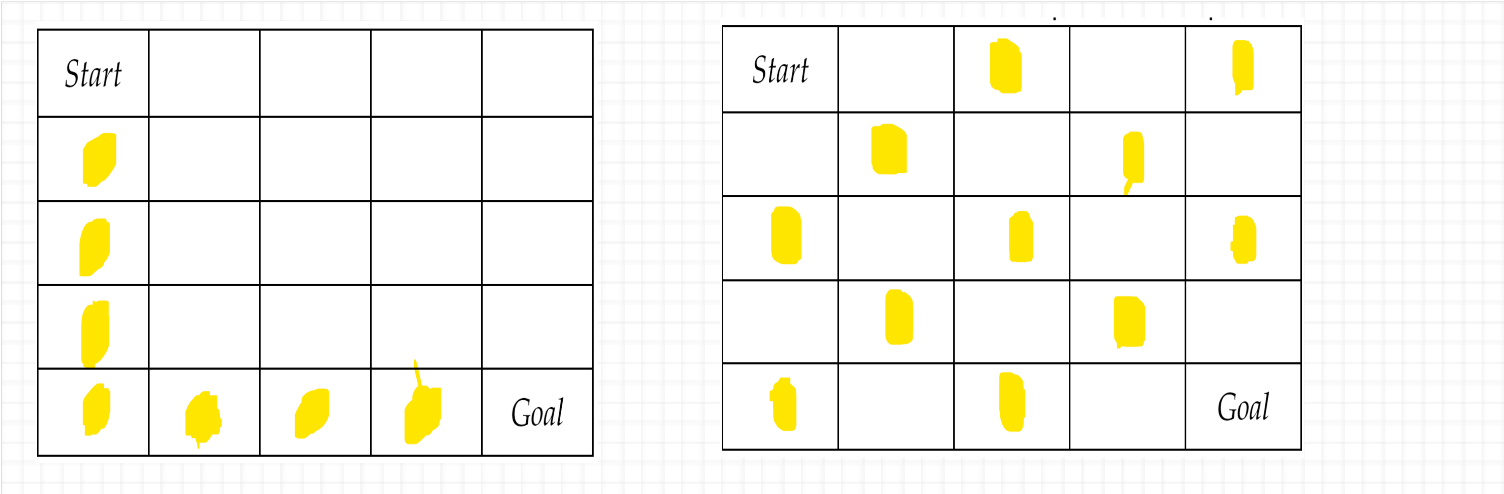
עבור $DFS-L$ הרגיל נקבל כי סיבוכיות המקום נשארת זהה אך סיבוכיות הזמן שקיבלנו טובה יותר:

$$b < 3 \Rightarrow O(3b)^{\frac{d}{2}} < O(b \cdot d)$$

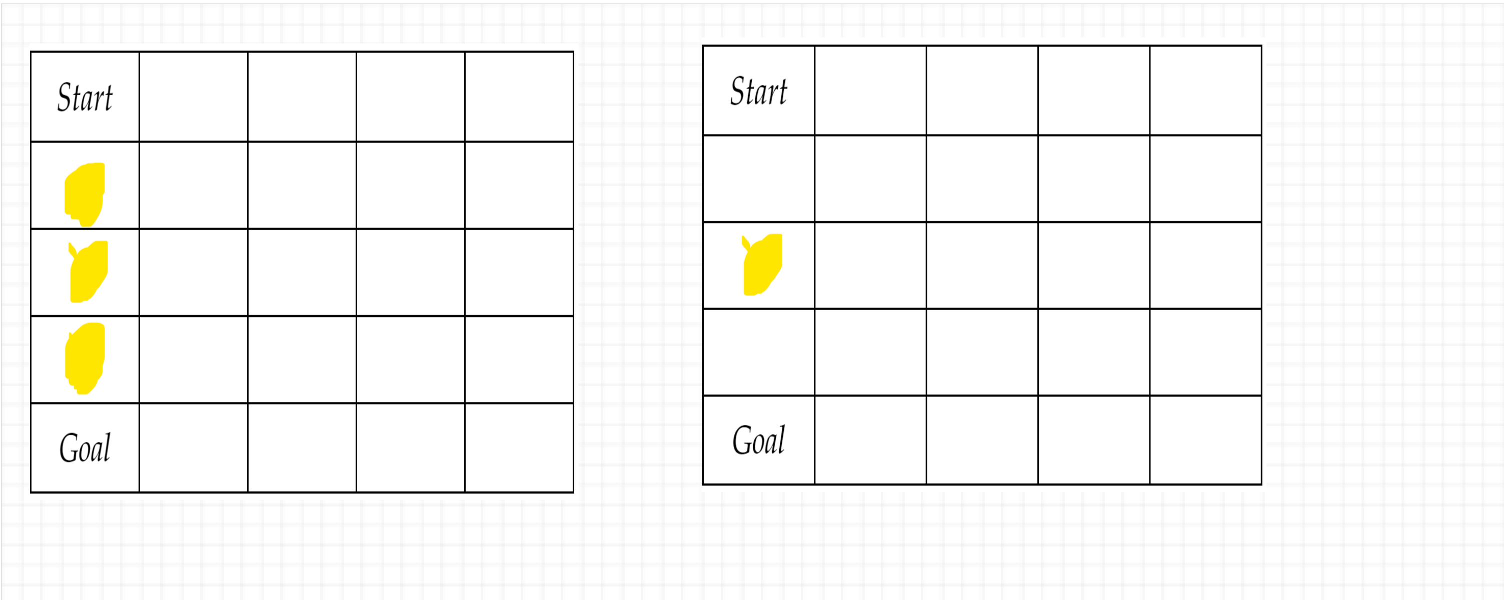
---

4. ספקו דוגמה לבעיה שבה DFS-L במרחב החיפוש החדש (לאחר השינויים שביצעתם בסעיף 5.1) טובה יותר מאשר DFS-L במרחב החיפוש הקודם, ודוגמה לבעיה שבה DFS-L במרחב המקורי עדיף. בתשובתכם התייחסו למספר הצמתים שפותחו. דוגמות יכולות להיות כלליות ולא בהכרח מסביבת הקמפוס. (2 נק')

דוגמה לכך שהמימוש הישן יותר טוב מאשר בהחדש כי הוא פיתח 8 צמתים אבל המימוש החדש פיתח 12 צמתים .:



דוגמה שניה , בה המימוש החדש יותר טוב כי הוא פיתח 2 צמתים אבל המימוש החדש פיתח 4 צמתים:



שאלה 4:

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

כן, האלגוריתם בבעיית החיפוש שלנו הוא גם שלם וגם קביל.
האלגוריתם שלם מכיוון שפונקציית המחירים חסומה ע"י 1, ובמקרה של בעיית החיפוש שלנו פונקציית המחירים היא פונקציית המשקלים, והמשקל הקל ביותר של קשת הוא 1.
והאלגוריתם קביל מכיוון שהגרף שלנו הוא סופי.

---

3. עבור אילו בעיות חיפוש אלגוריתם UCS ואלגוריתם BFS-G יפעלו באותו האופן? הסבירו. (1 נק׳)

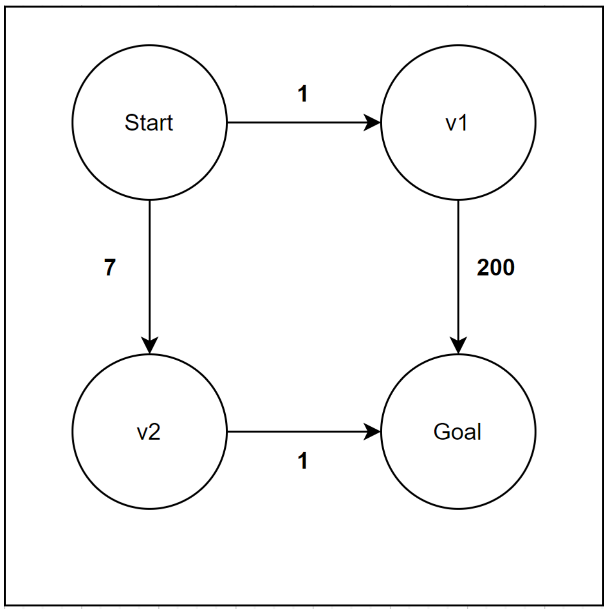
עבור בעיות חיפוש שבהן המחיר על הקשתות שיוצאות מצמתים באותה רמה יהיה זהה, כך שהמחיר של הקשתות היוצאות מהצמתים ברמה מסויימת יהיה זהה g, ויהיה קטן יותר מזה של כל הרמות הנמצאות מתחתה.
במצב הזה, אלגוריתם UCS יפתח את הצמתים לפי המחיר הקטן ביותר, ולכן יפתח את הצמתים רמה רמה, ובצורה זו הוא פועל כמו אלגוריתם $BFS - G$.

---

4. איימי טעתה במימוש של אלגוריתם UCS ובדקה בטעות בזמן <u>יצירת</u> הצומת האם הוא צומת מטרה במקום בזמן <u>הפיתוח</u> שלו (כלומר, לאחר הוצאתו מתור העדיפויות). תנו דוגמה לגרף חיפוש שעבורו האלג׳ שאיימי מימשה יחזיר בכל זאת את המסלול הקל ביותר, ודוגמה לגרף חיפוש שעבורו האלג׳ לא יחזיר את המסלול הקל ביותר. (2 נק׳)

- 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 הניווט הקמפוס – ניתן לתת דוגמה לגרף שמייצג בעיית חיפוש אחרת.
- על הגרפים המוצעים להכיל קשתות מכוונות וכן את העלות של כל קשת.

1) דוגמא לגרף שעבורו האלגוריתם של איימי מחזיר תשובה לא נכונה:



אלגוריתם UCS המקורי מחזיר את המסלול הבא שמשקלו הוא 8:

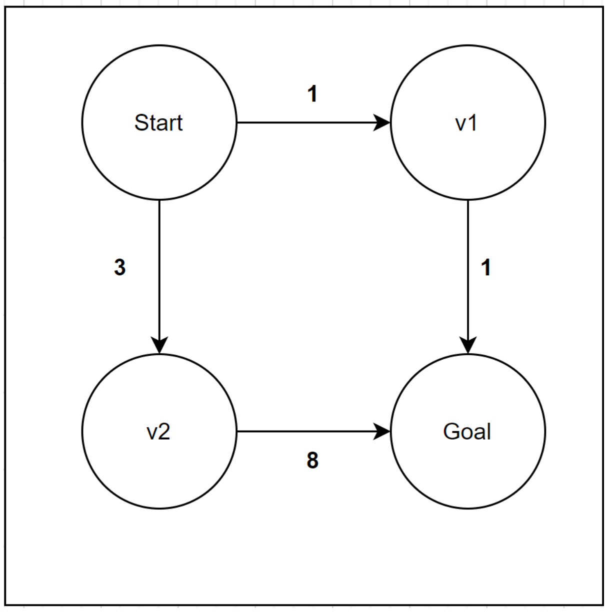
$$Start \rightarrow v_2 \rightarrow Goal$$

אבל האלגוריתם השגוי של איימי מחזיר את המסלול שמשקלו הוא 201:

$$Start \rightarrow v_1 \rightarrow Goal$$

כי הוא יפתח תחילה את הצומת v_1 והצומת v_2, ואז ממשיך בצומת v_1 כי המחיר אליה נמוך יותר, ואז יפתח את הצומת $Goal$ ויגלה כי זו צומת מטרה, ואז יחזיר את המסלול הנזכר לעיל.

2) דוגמא לגרף שעבורו האלגוריתם של איימי כן מחזיר תשובה נכונה:



אלגוריתם UCS המקורי מחזיר את המסלול הבא שמשקלו הוא 2:

$$Start \rightarrow v_2 \rightarrow Goal$$

אבל האלגוריתם השגוי של איימי מחזיר את המסלול שמשקלו הוא 2:

$$Start \rightarrow v_2 \rightarrow Goal$$

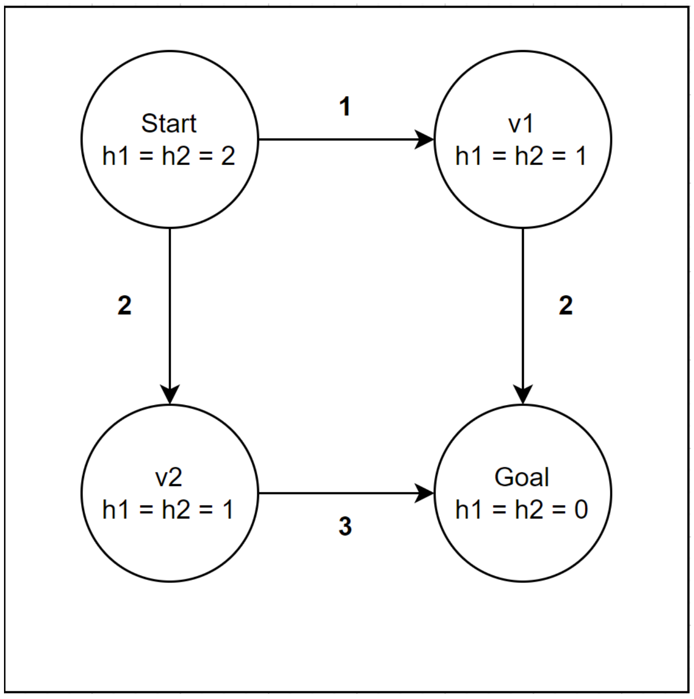
כי הוא יפתח תחילה את הצומת v_1 והצומת v_2, ואז ממשיך בצומת v_1 כי המחיר אליה נמוך יותר, ואז יפתח את הצומת $Goal$ ויגלה כי זו צומת מטרה, ואז יחזיר את המסלול הנזכר לעיל.

<u>שאלה 5:</u>

1. תהיינה שתי יוריסטיקות קבילות h_1, h_2.

1. הוכיחו/הפריכו : היוריסטיקה $h = h_1 + h_2$ קבילה. (1 נקי)

הפרכה: נסתכל על הדוגמה הנגדית הבאה:



כאשר הגדרת ההיורסטוקות $h_1,\ h_2$ הן כמו שמופיע בציור, עכשיו עבור היוריסטיקה החדשה מתקיים כי:

$$h(start) = h_1(start) + h_2(start) = 4$$
$$h(v_1) = h_1(v_1) + h_2(v_1) = 2$$
$$h(v_2) = h_1(v_2) + h_2(v_2) = 2$$
$$h(Goal) = h_1(Goal) + h_2(Goal) = 0$$

מתקיים כי העלות של המסלול מ- $Start$ ל- $Goal$ הוא 3, אבל עבור היוריסטיקה החדשה מתקיים כי:

$$h(Start) = 4 > 3$$

ולכן היוריסטיקה אינה קבילה.

2. הוכיחו/הפריכו : היוריסטיקה $h = \frac{h_1+h_2}{2}$ קבילה. (1 נקי)

הוכחה: מתקיים כי: $0 \leqslant h_1,\ h_2 \leqslant h^*$

$$h = \frac{h_1 + h_2}{2} \leqslant \frac{h^* + h^*}{2} = \frac{2 \cdot h^*}{2} = h^*$$

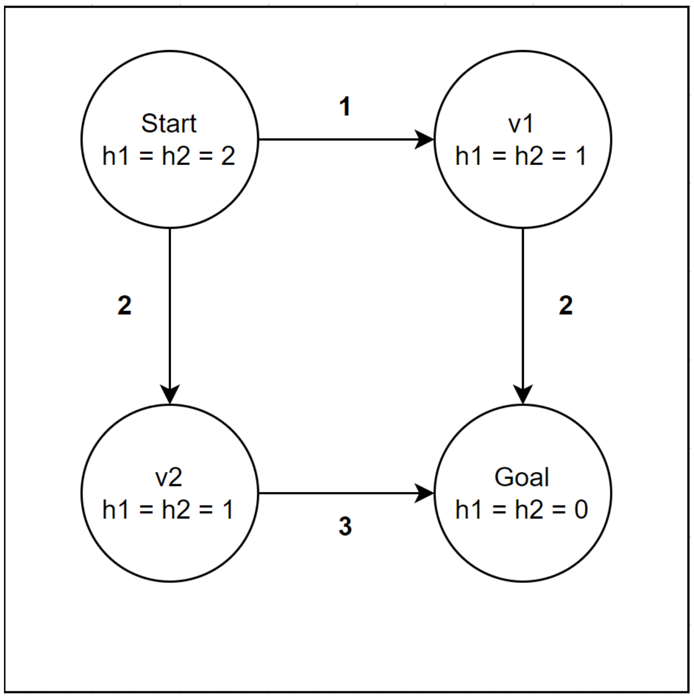
ולכן מתקיים לפי ההגדרה של יוריסטיקה קבילה, כי לכל $s \in S$, $h \leqslant h^*$ ולכן היוריסטיקה בסעיף זה קבילה.

---

2. תהיינה שתי יוריסטיקות עקביות h_1, h_2.

1. הוכיחו/הפריכו : היוריסטיקה $h = h_1 + h_2$ עקבית. (1 נקי)

הפרכה: נסתכל על אותה הדוגמה מסעיף 1.2:



היוריסטיקה עקבית כי אכן מתקיים לפי ההגדרה של יוריסטיקה עקבית כי $h_1,\ h_2$ אכן עקביות:

$$\forall v_1,\ v_2\ \in\ S \mid e = \left(u_1 \to v_2\right) \in E:\ h_1(v_1) - h_1(v_2)\ \leqslant\ cost(e)$$
$$\wedge\ h_2(v_1)\ -\ h_2(v_2)\ \leqslant\ cost(e)$$

אבל עבור ההיוריסטיקה החדשה $h = h_1 + h_2$ ההגדרה לא מתקיימת כי:

$$h(Start) - h(v_1) = 4 - 2 = 2 > 1 = cost\left(Start \to v_1\right)$$

2. הוכיחו/הפריכו : היוריסטיקה $h = \frac{h_1+h_2}{2}$ עקבית. (1 נק')

נתון כי היוריסטיקות $h_1,\ h_2$ עקביות, ולכן לפי מה שלמדנו בתרגול הן גם קבילות, ומתקיים לפי הגדרת יוריסטיקה עקבית כי:

$$\forall v_1,\ v_2\ \in\ S \mid e = \left(u_1 \to v_2\right) \in E:\ h_1(v_1) - h_1(v_2)\ \leqslant\ cost(e)$$
$$\wedge\ h_2(v_1)\ -\ h_2(v_2)\ \leqslant\ cost(e)$$
$$0\ \leqslant\ h_1,\ h_2\ \leqslant\ h^*$$

נסכום את שני האי-שוויונים הללו ונקבל את הדרוש:

$$h_1(v_1) - h_1(v_2) + h_1(v_1) - h_1(v_1) \leqslant 2 \cdot cost(e)$$

נארגן את אי-השוויון מחדש ונקבל כי h היא עקבית לפי ההגדרה של יורסטיקה עקבית:

$$h_1(v_1) + h_2(v_1) - (h_2(v_1) + h_2(v_2)) \leqslant 2 \cdot cost(e)$$

$$\frac{h_1(v_1) + h_2(v_1)}{2} - \frac{h_2(v_1) + h_2(v_2)}{2} \leqslant cost(e)$$

$$h(v_1) - h(v_2) \leqslant cost(e)$$

---

3. נגדיר יוריסטיקה חדשה עבור בעיית הניווט בקמפוס :

$$h_{CAMPUS}(s) = min\{\ min\{h_{Manhatan}(s,g) \mid g \in G\ \}, C_{portal}\}$$

כאשר הביטוי $h_{Manhatan}(s,g)$ מתאר את מרחק מנהטן מהמצב הנוכחי למצב סופי, והביטוי C_{portal} מתאר את מחיר השימוש ב-Portal (לדוגמה, 100 בבעיית הניווט במקפוס). שימו לב כי היוריסטיקה מחשבת את מרחקי מנהטן מהמצב הנוכחי למצב סופי על פני כל צמתי היעד.

4. האם היוריסטיקה h_{CAMPUS} קבילה עבור כל מפה בבעיית הניווט בקמפוס? אם כן, הסבירו בקצרה. אם לא, ספקו דוגמה נגדית. (2 נק')

כן, היוריסטיקה קבילה בכל מפה בבעיית הניווט בקמפוס, נסביר ע"י חילוק למקרים של הפתרון האופטימלי:

❖ עובר ב- $portal$: אז מתקיים כי מחיר המסלול הכולל הוא גדול או שווה למחיר השימוש ב- $portal$, כלומר מתקיים:

$$h^*(s) \geqslant C_{portal} = 100 \geqslant h_{CAMPUS}(s) = min\{min\{h_{Manhatan}(s,g)|g \in G\}, C_{portal}\}$$

ולכן נקבל כי במקרה הזה היוריסטיקה כן קבילה.

❖ לא עובר ב- $portal$: מתקיים במקרה הזה כי צריך לבצע כמרחק $manhatan$ צעדים בשביל להגיע לצומת מטרה, ולכן לפי ההגדרה של היוריסטיקה החדשה h_{CAMPUS} מכיוון שהיא בוחרת במינימלי, אז נקבל כי:

$$h_{CAMPUS} \leqslant h_{Manhatan}(s) \leqslant h^*(s)$$

כלומר נקבל כי היוריסטיקה אכן קבילה לפי ההגדרה.

★ הערה חשובה עבור שני המקרים הללו, היא כי המחירים של $L,\ F,\ Portal$ גדולים או שווים לאפס, וצריכים את זה בשביל הגדרת יוריסטיקה קבילה.

---

5. האם היוריסטיקה h_{CAMPUS} עקבית עבור כל מפה בבעיית הניווט בקמפוס? אם כן, הסבירו בקצרה. אם לא, ספקו דוגמה נגדית. (2 נק')

נכון, היוריסטיקה h_{CAMPUS} עקבית עבור כל מפה בבעיית הניווט בקמפוס, נוכיח זאת לפי ההגדרה:
תהי קשת $(s_1,\ s_2) \in S$ כאשר $s_1,\ s_2\ \in\ S$ נרצה להוכיח את הגדרת העקביות כלומר נרצה שיתקיים:

$$h_{CAMPUS}(s_1) - h_{CAMPUS}(s_2) \leqslant cost(s_1 \rightarrow s_2)$$

נחלק למקרים לפי המחיר של הקשת:

❖ אם משתמשים ב- $portal$ אז $cost(s_1 \rightarrow s_2) = 100$, ומתקיים כי $\forall s \in S: h_{CAMPUS}(s) \leqslant 100$ ולכן ההפרש בערך היוריסטיקה של שני צמתים כלשהם תמיד מקיים: $h_{CAMPUS}(s_1) - h_{CAMPUS}(s_2) \leqslant 100$ כלומר קיבלנו את הדרוש.

❖ אם לא משתמשים ב- $portal$ אז $h_{CAMPUS}(s)$ הוא מחושב לפי מרחק $Manhatan$ ואז מתקיים לפי הגדרת h_{CAMPUS} כי מרחק $Manhatan$ קטן או שווה ל- C_{Portal}, במקרה הזה מחיר הקשת הוא אחד מהבא: $1,\ 2,\ 3,\ 10$, ועבור המחירים האלה, $s_1,\ s_2$ הפ בהכרח שכנים, ולכן מרחק $Manhatan$ שלהם שווה ל- 1.
אם מתקיים שערך היוריסטיקה שווה ל- C_{portal} אז נקבל כי $h_{CAMPUS}(s_1) - h_{CAMPUS}(s_2) = 0$ ונקבל את הדרוש.
מכיוון ש- h_{CAMPUS} מחושב לפי מרחק $Manhatan$ ולפי ההסבר שהזכרנו קודם נקבל כי:

$$h_{CAMPUS}(s_1) - h_{CAMPUS}(s_2) \leqslant 1, 2, 3, 10 = cost(s_1 \rightarrow s_2)$$

וגם במקרה הזה נקבל את הדרוש.

שאלה 6:

1. האם האלגי Greedy Best First Search, על מפה כללית עבור בעיית הניווט בקמפוס בגודל NxN, הוא שלם? האם הוא קביל? (1 נקי)

מכיוון שמרחב החיפוש הוא סופי עבור בעיית הניווט בקמפוס, אזי לפי מה שלמדנו אלגוריתם *Greedy Best First Search* שלם במקרה הזה, אבל הוא לא קביל כי עבור היוריסטיקה h_{campus} שיכולה לספק תשובות שגויות ועבור הדוגמה הבאה הקבל כי אכן האלגוריתם אינו קביל:

Start	*F*	*F*	*Goal*
L	*L*	*L*	*L*
L	*L*	*L*	*L*
L	*L*	*L*	*L*

לפי היורסטיקה h_{campus} הצומת הראשון שנפתח הוא F ואז שוב F כי היא מסתכלת על ה- *Manhattan distance*, ואז יפתח G, למרות שיש מסלולים אחרים שעוברים רק דרך L עם עלות יותר נמוכה מזה שנבחר ביורסטיקה, ולכן האלגוריתם *Greedy Best First Search* אינו קביל.

---

2. תנו יתרון וחיסרון של האלגי Greedy Best first Search ביחס ל-Beam Search. בתשובה התייחסו להגדרות השלמות והעקביות ולסיבוכיות הזמן והזיכרון. תוכלו להתייחס לבעיית חיפוש כללית, ולא ספציפית עבור בעיית הניווט בקמפוס. (2 נקי)

יתרון: אלגוריתם *Greedy Best First Search* לא פוגע בפתרון, להיפך מאלגוריתם *Beam Search* שמגביל את כמות הצמתים הנפתחים לגודל מסויים, וזורק צמתים כאשר עוברים את הגודל הזה.
חסרון: אלגוריתם *Greedy Best First Search* עלול להשתמש ביותר זיכרון מאלגוריתם *Beam Search* שמגביל את כמות הצמתים הנפתחים, שאכן בצורה זו משתמש בפחות זיכרון.

שאלה 7:

2. בכיתה הגדרנו את פונקציית ההערכה עבור הצומת הבא לפיתוח במהלך ריצת האלג׳ A* באופן הבא:

$$f(v) = h(v) + g(v)$$

איימי טוענת כי אפשר להשתמש בפונקציה $f'(v) = \frac{h(v)+g(v)}{2}$ במקום ב-$f(v)$, ותוצאת הריצה של האלג׳ תהיה שקולה.

הסבירו בקצרה מדוע איימי צודקת (בתשובתכם התייחסו לסדר פיתוח הצמתים, המסלול המוחזר ועלות המסלול המוחזר מהאלג׳ בעת השימוש בפונקציה $f'(v)$). (2 נק׳)

איימי צודקת מכיוון שאלגוריתם A^* פותח את הצמתים לפי ערכי ה- f של כל צומת, ולכן אם היה מתקיים עבור שני צמתים כי:

$$f(v_1) \leqslant f(v_2) \Longrightarrow \frac{f(v_1)}{2} \leqslant \frac{f(v_2)}{2}$$

כי מבצעים חלוקה במספר שלם חיובי גדול מ- 1, ולכן הסדר נשמר, ובחירת הצמתים, פתיחתם $OPEN$ וגם סגירתם $CLOSE$ נשמרת ולכן המסלול המוחזר מהאלגוריתם עם הפונקצייה f החדשה לא ישתנה, ובנוסף לזה שהמסלול עצמו נשמר, אנחנו לא משנים את העלויות ולכן נקבל את אותה התוצאה.

---

4. תנו יתרון וחסרון של האלג׳ ID-A* ביחס ל-A*. באילו מקרים הייתם מעדיפים להשתמש בכל אחד מהם? (2 נק׳)

יתרון: אלגוריתם $ID-A^*$ משתמש בפחות זיכרון מאשר אלגוריתם A^*, וסיבוכיות הזיכרון שלו לינארית באורך המסלול, אבל אלגוריתם A^* תלוי במספר הצמתים שנפתחו.

חסרון: אלגוריתם $ID-A^*$ פותח את הצמתים הרבה פעמים, בניגוד להתנהגות של אלגוריתם A^* שפותח כל צומת רק פעם אחת, שעלול לגרום לכך ש- $ID-A^*$ לוקח זמן יותר מ- A^*.

במקרים שיש בהם הגבלת זיכרון, נעדיף להשתמש ב- $ID-A^*$ כי משתמש בפחות זיכרון מ- A^*.

ובמקרים שיש בהם הגבלת זמן, נעדיף להשתמש ב- A^* כי זמן הריצה שלו פחות מזה של $ID-A^*$, כפי שהסברנו לעיל.

---

5. תנו יתרון וחיסרון של האלג׳ A*-epsilon ביחס ל-A*. באילו מקרים הייתם מעדיפים להשתמש בכל אחד מהם? (2 נק׳)

יתרון: אלגוריתם $A^*-epsilon$ מוצא פתרון בצורה מהירה יותר מ- A^*, כי מאפשר בחירה לא אופטימלית ובמקביל לוקח זמן יותר קצר.

חסרון: אלגוריתם $A^*-epsilon$ עלול להחזיר פתרון שהוא לא אופטימלי, כי מאפשר סטייה מהפתרון האופטימלי עד כדי $epsilon$ שנבחר או נקבע לאלגוריתם.

במקרים שאנו רוצים להעדיף את זמן הריצה על אופטימליות הפתרון, עדיף לנו להשתמש באלגוריתם $A^*-epsilon$.

ובמקרים שהאופטימליות היא העדיפות הראשונה לעומת זמן הריצה, אז נעדיף את אלגוריתם A^*, וזה לפי מה שהסברנו לעיל

שאלה 8:

W-A* (0.9)	W-A* (0.9)	W-A* (0.7)	W-A* (0.7)	W-A* (0.3)	W-A* (0.3)	A* num of	A* cost	UCS num	UCS cost	DFS-G nu	DFS-G co	map
26	89	82	87	94	87	92	87	97	87	33	121	map12x12
42	129	150	106	167	106	167	106	167	106	47	181	map15x15
56	202	298	175	308	175	308	175	309	175	57	282	map20x20

לפי מה שאנו רואים, אלגוריתם $DFS-G$ הוא פתרון לא הכי יעיל ולא אופטימלי.
UCS כן משפר את עלות כך שהוא מקטין את העלות אבל הוא מפתח יותר צמתים כי לפי מה שראינו ה- UCS כי הוא בודק יותר צמתים עד שימצא את המסלול בעל העלות המינימלית.

$W-A^*$:

עבור $W=\frac{1}{2}$ קיבלנו כי הוא הפתרון האופטימלי שהוא כן שווה לעלות של UCS אבל פותח פחות צמתים מ- UCS. אבל כל עוד מעלים את המשקל מקבלים פחות צמתים לעומת זאת שהעלות עולה שכן לא מקבלים פתרון אופטימלי וזה לפי כך שעבור $W-A^*$ כאשר מגדילים את המשקל איכות הפתרון יורדת בניגוד לכך שהפתרון יהיה מהיר יותר.

<u>שאלה 9:</u>

1. כיצד יש להגדיר את המצבים במרחב החיפוש? (2 נק׳)

יש להגדיר כי כל מצב הוא בעצם פרמוטציה של המילים, כלומר כל מצב הוא מסמך או ווקטור באורך n שהוא סידור שונה של המילים, ובמקום ה- i בווקטור נמצאת מילה במקום ה- i במסמך.

---

2. מהו מספר המצבים במרחב החיפוש? (1 נק׳)

מספר המצבים במרחב החיפוש הוא $n!$, שזה שווה למספר הסידורים השונים או במילים אחרות מספר הפרמוטציות של n מילים בווקטור.

---

3. אתם יודעים כי איימי קיבלה 100 בתרגיל בית 1 של ״מבוא לבינה מלאכותית״, ולכן מבקשים את עזרתה גם בהפעלת האלגוריתם. היא מציעה לכם להשתמש ב-SAHC (Steepest Ascent Hill Climbing), על מנת למצוא פתרון.
האם האלגוריתם Steepest Ascent Hill Climbing בהכרח ימצא פתרון? (2 נק׳)

כן, אלגוריתם *Steepest Ascent Hill Climbing* בהכרח ימצא פתרון, כי האלגוריתם הזה חמדני, ובכל צעד בוחר את השכן המשפר ביותר כפי שלמדנו בתרגול, ולכן בכל צעד בריצת האלגוריתם או שיש אפשרות לשיפור ויכול להחליף בין מילים ואז בוחר את הטובה ביותר, או שאין שיפור ואז בעצם הגענו למצב מקבל או מצב יעד, כלומר בעצם תמיד ימצא פתרון כי ממשיך בריצה עד שאין אפשרות כלשהי לשיפור.

---

4. איימי סיפרה לכם שאלון המתרגל יודע גם הוא קצת סינית. בשעת הקבלה אתם נעזרים בו בפתרון הבעיה, והוא מייעץ לכם להשתמש ב-SAHC with sidesways steps.

   1. האם האלגוריתם SAHC with sidesways steps בהכרח ימצא פתרון? (2 נק׳)

כן, בסעיף הקודם הסברנו כי בכל צעד יש אפשרות לבצע שיפור, ולכן בכל מצב שנעבור בו מלבד מצב היעד שבו אנחנו בעצם עוצרים ומגיעים לפתרון, אף פעם לא נגיד למצב שאין אפשרות לשיפור, ולכן ריצת האלגוריתם *SAHC with sideways steps* זהה לחלוטין לריצת האלגוריתם *SAHC* במקרה הזה.

---

   2. אלון טוען כי עבור המקרים בהם SAHC ו-SAHC with sideways steps מוצאים פתרון, עדיף להשתמש ב- SAHC with sideways steps כי ייתכן שהוא ימצא פתרון בפחות צעדים. האם אלון צודק? (3 נק׳)

אלון טועה, כי לפי מה שהסברנו בסעיף הקודם ריצת שני האלגוריתמים במקרה הזה זהה לחלוטין, ולכן מספר הצעדים של שניהם זהה.

---

5. אחרי דיון סוער בין אלון לאיימי, הגעתם לפשרה ומחליטים להשתמש באלגוריתם Stochastic Hill Climbing. האם האלגוריתם Stochastic Hill Climbing בהכרח ימצא פתרון? (2 נק׳)

כן, אלגוריתם *Stochastic Hill Climbing* בהכרח ימצא פתרון, כי בבעייה המתוארת אין חשיבות לסדר השיפור וההחלפות, ולכן למרות שהאלגוריתם רק מבטיח בחירת צומת משפר כלשהו ולאוו דווקא את המשפר ביותר, הוא בכל זאת יצליח למצוא פתרון.