מבוא לבינה מלאכותית - תרגיל בית 1

מגישים: יונתן אלבז - 213413586
עמיר בורבין - 325310688

<u>שאלה 1</u>

1. עברנו.
2.

$S\ =\ \{Tuples\ of\ the\ following\ form\ (location,\ Bool,\ Bool)\ W.R.T\ Board\ size\}$
$O\ =\ \{D,\ R,\ U,\ L\}$
$I\ =\ \{(0,\ false,\ false)\}$
$G\ =\ \{Tuples\ of\ the\ following\ form\ (location,\ True,\ True)\ where\ map[location]\ ==\ "G"\}$

3. הפונקציה Domain על האופרטור UP -2 תחזיר:

$\{s \in map\ |\ map[s[0]]\ !=\ "H"\}$

כלומר את כל המצבים בלוח שהם לא hole.

4. הפונקציה Succ על מצב התחלתי 0 תחזיר לנו את הפלט הבא:

$$(0,\ False,\ False),\ (1,\ False,\ False),\ (8,\ False,\ False)$$

5. כן.
לדוגמה לנוע מעלה אינסופית החל מהמצב ההתחלתי.

6. 4, מכל מצב ניתן לבצע 4 פעולות שונות

7. אינסוף, מכיוון שקיימים מעגלים, כלומר סוכן כללי לא בהכרח יגיע למטרה

8. 16, על ידי מסלול שתחילה יורד מטה לאסוף את הכדור השמאלי, ואז ילך לכדור השני בדרך הקצרה ביותר, ואז ימשיך ישירות למטרה.

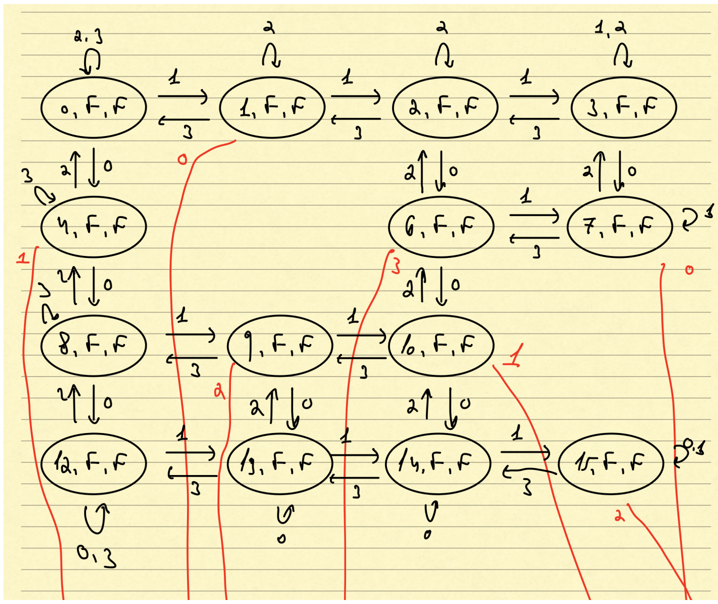
9. לא.
דוגמה נגדית:

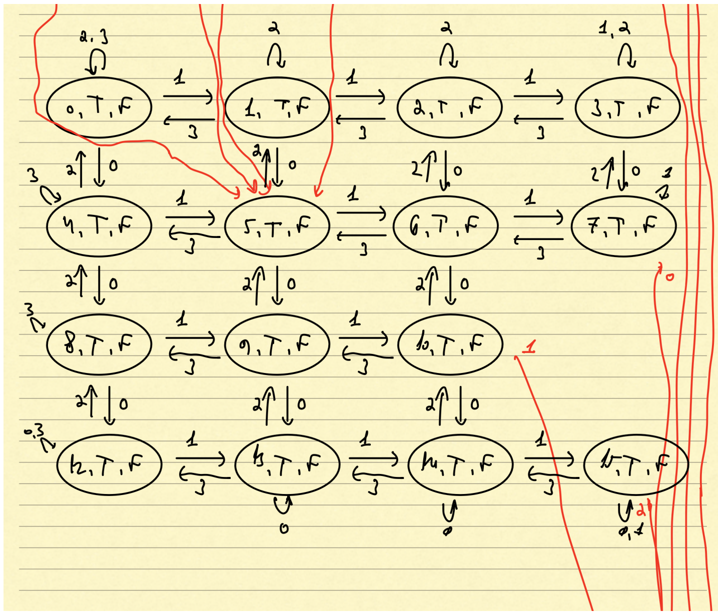
S	H	H	G
F	G	H	D
F	H	H	D
F	F	F	F

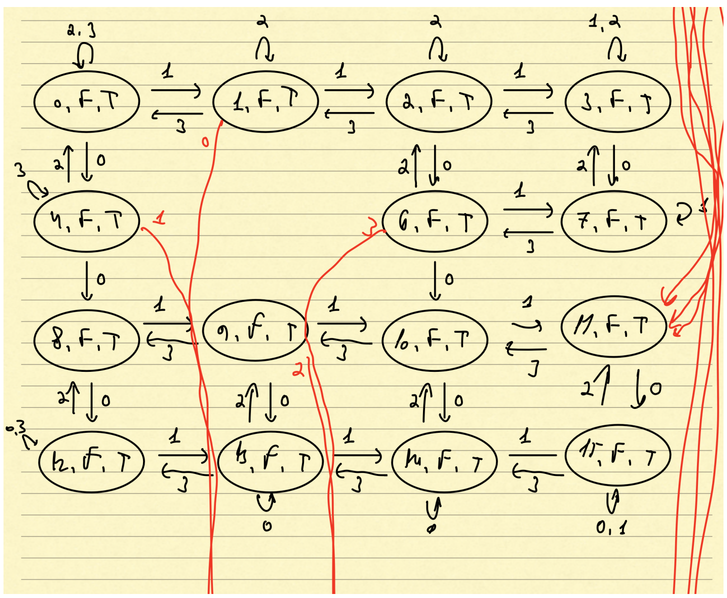
שאלה 2

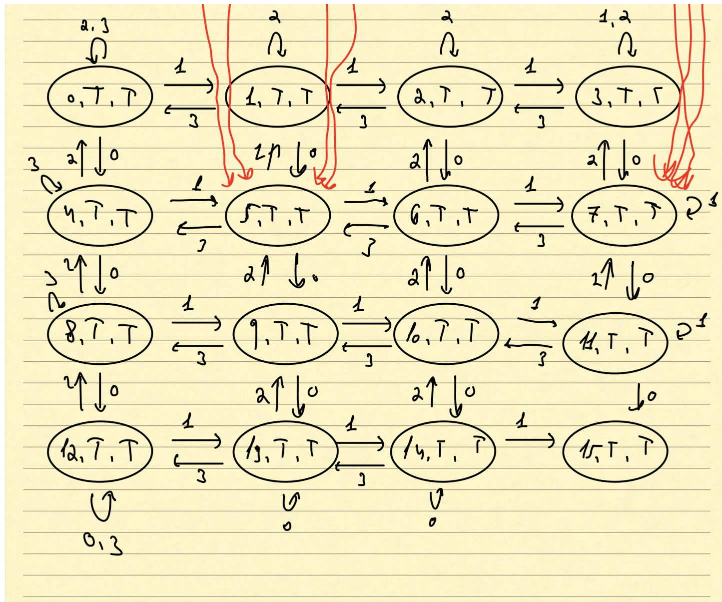
2. שהמרחק (לפי מספר קשתות במסלול) מהשורש למצב מטרה כלשהו קטן מהמרחק מהשורש לכל מעגל (לקשת האחרונה במעגל- זו שסוגרת את המעגל וגורמת לנו לפתח צומת שפותח כבר בעבר). זאת משום שקיים שוני רק ברגע בו נרצה לפתח צומת שכבר ביקרנו בו, ומכיוון שהסדר של פיתוח צמתים בBFS הוא לפי מרחק מהשורש (לפי מספר קשתות).

3. מצורפת תמונה של גרף המצבים:




0, T, F
1, T, F
2, T, F
3, T, F
4, T, F
5, T, F
6, T, F
7, T, F
8, T, F
9, T, F
10, T, F
12, T, F
13, T, F
14, T, F
15, T, F


0, F, T
1, F, T
2, F, T
3, F, T
4, F, T
6, F, T
7, F, T
8, F, T
9, F, T
10, F, T
11, F, T
12, F, T
13, F, T
14, F, T
15, F, T



4. נרצה להגדיר את הפונקציה $T:\ G\ =\ (V, E) \to G'(V',\ E')$ הבאה באופן הבא:

עבור כל 2 צמתים בגרף $v_1,\ v_2\ \in\ V$ אשר מחוברים ע״ קשת $e\ =\ E$ נוסיף לקבוצה V' צמתים $v'_1,\ v'_2\ ,\ \dots\ v'_{w(e)-2}$. ואת הצמתים $v_1,\ v_2$ נחבר לצמתים $v'_1,\ v'_{w(e)-2}$ בהתאמה כך שמספר הקשתות החדשות זהה למשקל הקשתות המחברות בגף G המקורי, בנוסף נוסיף קשתות $e'_1,\ e'_2\ ,\ \dots\ e'_{w(e)-2} \in E'$ המחברות את הצמתים $v'_1,\ v'_2\ ,\ \dots\ v'_{w(e)-2}$ כך שמשקלן 0.

לסיכום עבור כל קשת בגרף המקורי עם משקל w בנינו את הגרף החדש כך שיהיו w צמתים המחוברים ע״י קשתות סרק (בעלות משקל 0) כך שעבור כל 2 צמתים בגרף המקורי המחוברים ע״ קשת. המרחק (מספר הצמתים ביניהם) בין צמתים האלו בגרף החדש יהיה כמשקל הקשת המחברת אותם בגרף המקורי וכך אלגוריתם BFS-G יחזיר את המסלול הקצר ביותר (מבחינת צמתים) אך בעצם המסלול הקצר ביותר בגרף החדש הוא המסלול עם העלות המינימלית.

5. בשאלה לא נתון כי יש משבצות של כדורי דרקון D אז אנחנו מניחים כי הם לא מופיעים בלוח.

במהלך ריצתו של אלגוריתם $BFS - G$ יפותחו $N^2 - 2$ צמתים, כולם חוץ ממצב המטרה והמצב אשר מייצגת המשבצת שמעל משבצת המטרה. לפי סדר הפיתוח שמגדירים בתרגיל, סדר פיתוח הצמתים הינו: למטה, ימינה, למעלה, שמאלה. בנוסף, לא נפתח מצב שכבר פיתחנו פעם נוספת. לפי כך אם נסתכל על הריצה של

BFS-G אנחנו נפתח את הצמתים בצורה הבאה:

1	3	6	10
2	5	9	13
4	8	12	15
7	11	14	16

כאשר נגיע לצומת 14 (נגיע אליו לפני צומת 15 לפי סדר הפיתוח שצוין לעיל) ואנחנו ניצור את הצומת שמתחתיו (אין כזה), לאחר מכן ניצור את הצומת שמימינו והוא צומת מטרה ולכן שם נעצור את הריצה. סך הכל קיבלנו כי כל הצמתים למעט 2 פותחו, כלומר N^2-2

שאלה 3

1. האלגוריתם הינו שלם ואינו קביל.

האלגוריתם שלם, כלומר אם קיים פיתרון הוא מחזיר פיתרון כלשהו. מכיוון שמרחב המצבים סופי, ולא חוזרים לפתח מצבים שכבר היינו בהם, בהכרח ריצת DFS תסתיים, ולכן אם קיים פיתרון הוא יוחזר.
אך, האלגוריתם אינו קביל, דוגמה נגדית:

["SFFF",
 "FFFD",
 "FFFD",
 "FFFG"]

DFS יחזיר את המסלול הבא:

$$0,F,F \to 4,F,F \to 8,F,F \to 12,F,F \to 13,F,F \to 14,F,F \to 15,F,F \to 11,F,T \to 15,F,T \to 14,F,T \to 10,F,T \to 6,F,T \to 7,T,T \to 11,T,T \to 15,T,T$$

בעקבות סדר האופרטורים ותכונות DFS-G.
אך, המסלול האופטימלי הינו:

$$0,F,F \to 1,F,F \to 2,F,F \to 3,F,F \to 7,T,F \to 11,T,T \to 15,T,T$$

שהוא קצר יותר.

2. על לוח NXN כללי לא היה מוצא פיתרון כלשהו. לדוגמה עבור הלוח הבא:

["SFFD",
 "FFFD",
 "FFFF",
 "FFFG"]

האלגוריתם היה מתנהג באופן הבא ונתקע בלולאה אינסופית בין המיקומים 11 ו-15- מכיוון שDFS על עץ

מפתח מצבים שכבר היה בהם:

$0, F, F \to 4, F, F \to 8, F, F \to 12, F, F \to 13, F, F \to 14, F, F \to 15, F, F \to 11, F, F \to 15, F, F \to 11, F, F \to 15, F, F \to 11, F, F \to 15, F, F \to 11, F, F \to 15, F, F \to 11, F, F \to 15, F, F...$

.3

1	3		
2	5		
4	7	10	
6	8	9	7

הצמתים המסומנים בכתום יפותחו והצמתים המסומנים בירוק ייוצרו כלומר, נקבל כי $2N-2$ יפותחו, וייווצרו : $2N + 2(N-3) + 1 = 4N-5$ מכיוון שDFS-G יוצא את כל הצמתים שמהם הוא יכול להמשיך אליהם ומפתח אחד מה לי סדר הפיתוח.

.4

1			
2			
4			
6	8	9	7

הצמתים המסומנים בכתום יפותחו והצמתים המסומנים בירוק ייוצרו כלומר, נקבל כי $2N-2$ יפותחו, וייווצרו : $2N-1$, מכיוון ש-backtracking יוצר את הצמתים ברגע הדרישה בלבד.

<u>שאלה 4</u>

.1

a. האלגוריתם שלם, נוכיח:

צ"ל: אם קיים פיתרון, אזי האלגוריתם מחזיר פיתרון כלשהו.

נניח כי קיים פיתרון, צריך להוכיח שהאלגוריתם מחזיר פיתרון כלשהו.

מכיוון שקיים פיתרון, נסמן את אורכו ב-L. לכן, במקרה הגרוע, באיטרציה ה-L של האלגוריתם (יכול להיות שלפני) האלגוריתם ימצא פיתרון- את הפיתרון שהנחנו שקיים באורך L.

.b

נוכיח כי האלגוריתם קביל.

עבור הפתרון האופטימלי, נסמן את אורכו L נניח בשלילה כי האלגוריתם DFS-G החזיר פתרון מאורך $L' > L$ מהגדרת האלגוריתם, לפני שהוא הגיע לפתרון מאורך L' הוא עבר על כל המסלולים מאורך L ובפרט על הפתרון האופטימלי, לכן, היה צריך להחזיר את הפתרון האוטימלי קודם ולכן קיבלנו סתירה.

הוחנו כי האלגוריתם קביל.

c. לפי התשובה לסעיף b, לא ניכר שיש בעיה שצריך לתקן.

.2

.a

דוגמה בה REVERSE עדיף על ID-DFS היא עבור D חסם עליון שהוא בדיוק עומק הפיתרון d, משום ש-REVERSE תוך 2 איטרציות יימצא את הפיתרון האופטימלי (עבור d-1,d ולא ימצא פיתרון באורך זה), לעומת זאת ID-DFS ייקח d איטרציות.

דוגמה בה ID-DFS עדיף על REVERSE היא עבור D חסם עליון גדול מאוד ביחס ל-d, משום ש-REVERSE תוך D-d איטרציות יימצא את הפיתרון האופטימלי, לעומת זאת ID-DFS ייקח d איטרציות, ומההנחה שD גדול מאוד ביחס ל-d, נקבל כי D-d גדול יותר מ-d, ולכן REVERSE פחות עדיף מ-ID-DFS.

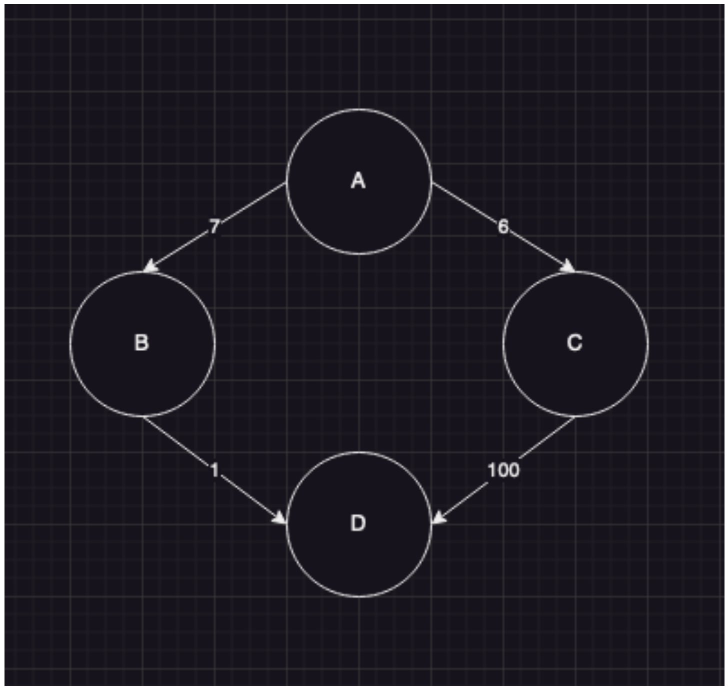
b. נציע את היעול הבא: בכל איטרציה נשמור את ה L – של האיטרציה הקודמת. בנוסף, במקום לעדכן את L באופן הבא: $L = L - 1$ נוכל לבצע חיפוש בינארי, כלומר, במידה ומצאנו פתרון נעדכן את L להיות $L = \frac{L}{2}$ ובמידה ולא מצאנו פתרון נעדכן את L להיות $L = \frac{curr\ L + prev\ L}{2}$

<u>שאלה 6</u>

1.עבור בעיות חיפוש בהן מחיר כל הפעולות/קשתות אחיד.

2. האלגוריתם הינו שלם והינו קביל. ננמק מדוע הוא קביל, ומכך נסיק כי הוא שלם. ננמק מדוע אם קיים פיתרון, UCS מחזיר פיתרון אופטימלי. זאת משום שהסדר שבו UCS עובר על הגרף הוא בסדר עולה לפי מחיר, בפרט, צומת שמוצא מ-OPEN הוא בעל המחיר המינימלי מכל אלה שבתור, ומכיוון שכל הקשתות במשקל אי שלילי, נקבל כי זהו המשקל הקטן ביותר שיינתן מאותו רגע בריצת האלגוריתם.

3. נראה דוגמה בה הטעות של שאדי גוררת שהאלגוריתם יחזיר מסלול שהוא לא הקל ביותר:

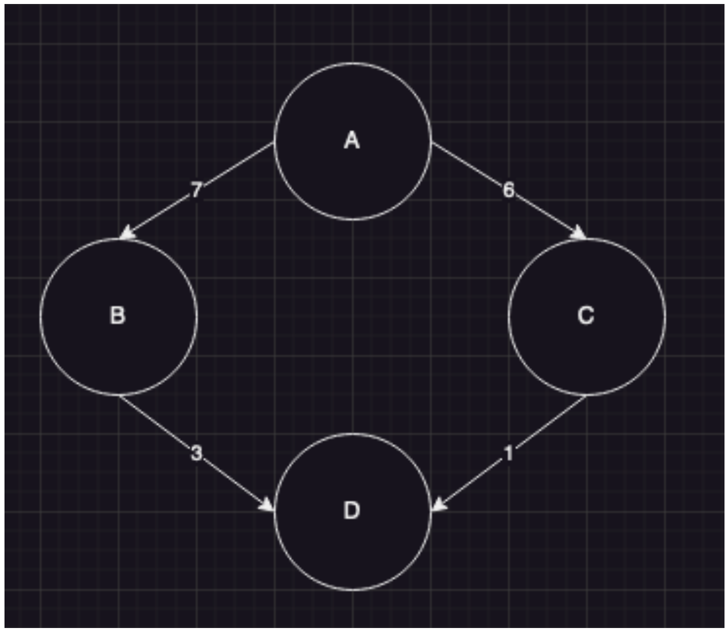


בדוגמה זו, מהלך האלגוריתם הוא כלהלן:

A יהיה ב-OPEN.

A יצא מ-OPEN, ו-B,C יהיו ב-OPEN עם ערכים 7,6 בהתאמה.

C יצא מ-OPEN, ואז יוחזר המסלול $A \to C \to D$, עם מחיר 106, למרות שהמסלול בעל המחיר המינימלי הוא דרך B עם מחיר 8.



עבור הדוגמה הזו, אלגוריתם UCS יצור את הצומת C ולאחר מכן ייצור את הצומת D וירשום עבורה מחיר 7 ויחזיר את המסלול הנ״ל, זה אכן המסלול בעל המחיר המינימלי ולכן הטעות של שאדי לא תבוא לידי ביטוי במקרה זה והאלגוריתם אכן יחזיר את המסלול הקל ביותר.

<u>שאלה 7</u>

1. קיים $\epsilon \geqslant 1$ כך שהיוריסטיקה היא ϵ – קבילה. זה מתקיים עבור $\epsilon = 1$, ברור למה הוא ההדוק ביותר- מכיוון שהוא המינימלי האפשרי. זהו ϵ מתאים, משום שמרחק מנהטן בעצם משרה מחיר 1 על כל קשת בין מצבים סמוכים, ובבעיה שלנו מחיר כל קשת הוא 1 (לפי התיקון בפיאצה- שמבהיר כי אורך כל כביש נקבע לפי מספר הפעולות, לכן עלות בין כל 2 צמתים סמוכים היא 1), מכך נסיק כי מחיר כל מסלול, שהוא סכום מחירי קשתות, בהכרח קטן שווה (בדיוק שווה בפרט) במרחק מנהטן ביחס לh^*.

2. קיים $\epsilon \geqslant 1$ כך שהיוריסטיקה היא ϵ – קבילה. זה מתקיים עבור $\epsilon = 1$, ברור למה הוא ההדוק ביותר- מכיוון שהוא המינימלי האפשרי. זהו ϵ מתאים, משום שניתן להשתכנע בצורה אלגברית שהיוריסטיקה המוגדרת סעיף זה תמיד קטנה או שווה ליורסטיקה המוגדרת בסעיף 1, ולכן נקבל כי:
יהי $s \in S$,
מתקיים:
$$h(s) \ \leqslant \ h_{MD}(s) \leqslant 1 * h^*(s)$$
כאשר מעבר מספר 1 נובע מכך שבסעיף 1 היורסטיקה סוכמת שני גדלים אי שליליים-a,b נסמנם, ובסעיף זה אנו לוקחים את המינימלי מבין שני איברים, c,d נסמנם,כך ש:$c \leqslant a,\ d \leqslant b$, והמעבר השני נובע מסעיף א'.

3. נראה כי $L^3 \ < \ MD$ נסתכל על הביטוי הבא:

$$|P-G|_3 \ = \ \sqrt[3]{|G_x-P_x|^3 \ + |G_y-P_y|^3} \ \leqslant_{1}$$
$$\sqrt[3]{|G_x-P_x|^3 \ + |G_y-P_y|^3 \ + \ 3|G_x-P_x|^2|G_y-P_y| + \ 3|G_y-P_y||G_x-P_x|^2}$$
$$=_{2} \sqrt[3]{(|G_x-P_x| \ + |G_y-P_y|)^3} \ = \ |G_x-P_x| \ + |G_y-P_y| \ = \ MD$$

הסבר מעברים:

1. הכל בערך מוחלט ולכן הכל חיובי ולכן כל מה שנוסיף רק יגדיל את הביטוי.
2. לפי הנוסחה: $(a+b)^3 \ = \ a^3 \ + \ 3a^2b + \ 3b^2a + b^3$

קיבלנו כי הנורמה L^3 קטנה ממרחק מנהטן, מכיוון שהוכחנו כי החסם ההדוק שלמרחק מנהטן הוא 1:
$h_{MD}(s) \ \leqslant \ h^*(s)$, נקבל כי גם עבור הנורמה L^3 החסם ההדוק הוא 1: $h_{L^3}(s) \ \leqslant \ h^*(s)$

4. יהיו $h_1, \ h_2$ שהן $\epsilon_1, \ \epsilon_2$ קבילות. נראה כי $h_3 \ = h_1 + \ h_2$ היא ϵ_3 קבילה כך ש - $\epsilon_3 \ = \ \epsilon_1 + \ \epsilon_2$
יהי $s \in S$,
מתקיים:

$$h_1(s) \ \leqslant \ \epsilon_1 h^*(s), \ h_2(s) \ \leqslant \ \epsilon_2 h^*(s) \ \Longrightarrow$$
$$h_3(s) \ = h_1(s) + \ h_2(s) \ \leqslant \ \epsilon_1 h^*(s) \ + \ \epsilon_2 h^*(s) \ = \ (\epsilon_1 + \ \epsilon_2) h^*(s)$$

קיבלנו כי $h_3 \ = h_1 + \ h_2$ היא ϵ_3 קבילה כך - $\epsilon_3 \ = \ \epsilon_1 + \ \epsilon_2$, ו-$\epsilon_3$ הוא החסם ההדוק ביותר בעקבות הנתון ש-$\epsilon_1, \ \epsilon_2$ הם החסמים ההדוקים ביותר, והמעבר היחיד של אי שיוויון שביצענו בחישוב הוא חסימה על ידי $\epsilon_1, \ \epsilon_2$.

5. כן.
זאת משום שהיורסטיקה המושלמת, כפי שהסברנו לעיל היא מרחק מנהטן מ-s למטרה G, והיוריסטיקה כאן היא מינימום בין איברים, שביניהם נמצא מרחק מנהטן מ-s למטרה, ולכן נקבל כי לכל s היורסטיקה קטנה או שווה מהיוריסטיקה המושלמת (וגם גדולה שווה ל-0 מהגדרת מרחק מנהטן).

6. כן.
ראשית נציין כי $cost(s,s') \ = \ 1$ לכל $s \in S, \ s' \in succ(s)$, מכיוון שנדרשת פעולה אחת על מנת לעבור בין מצב לסמוכו. טענה: בתזוזה של משבצת יחידה ניתן לשפר את המרחק מנהטן למשבצת אחרת בלכל היותר 1. זה מתקיים מכיוון שניתן לזוז רק לאורך הצירים, לכן תזוזה לכיוון נכון באחד הצירים אכן תקצר את המרחק ב-1 בלבד. לכן, המרחק הקטן ביותר שנקבל הוא לכל הפחות קטן ב-1 מהמרחק שהיה לפני, ולכן גם ערך היורסטיקה שהגדרנו קטן ב-1 לכל היותר. לכן הסברנו.

7. לא.

דוגמה נגדית:

["SFDD",
"FFFF",
"FFFF",
"FFFG"]

ראינו בתרגולים כי היוריסטיקה המושלמת מקיימת כי:

$$\forall g \in G:\ h(g) \ = \ 0$$

עבור $s \ = \ (15, T, T)$ ערך היורסטיקה הוא 4, וזהו מצב מטרה.

8. לא, בדומה ל-7.

<u>שאלה 8</u>

1. האלגוריתם הינו שלם, מכיוון שמרחב המצבים סופי וקשיר, בהתאם למה שראינו בתרגול.
אך, האלגוריתם אינו קביל, מכיוון שאין שום הגבלה לגבי היוריסטיקה. אם לדוגמה נבחר את היוריסטיקה להיות $\forall s \in S:\ -h^*(s)$, נקבל כי היוריסטיקה תטעה את האלגוריתם ותוביל אותו לפיתרון לא אופטימלי.

2. יתרון: אלגוריתם Beam חוסך בזיכרון מכיוון שהוא מגביל את רשימת הצמתים ב-open למספר קבוע K.
חסרון: עלול לפגוע בטיב בפתרון ואפילו במציאותו, יכול להיות שהיוריסטיקה שהגדרנו אינה מושלמת ועבור צומת שהכרחי לעבור בו על מנת להגיע לפתרון יקבל ערך יוריסטי גבוה ולא ייכנס לרשימת open ולכן לא נגיע לפתרון לעולם.

<u>שאלה 9</u>

2.
a. לא. דוגמה נגדית: h=0, נקבל כי ללא תלות ב-w, האלגוריתם מתנהג בצורה זהה ולכן נקבל כי:
$cost(p_1) \ = \ cost(p_2)$. בנוסף, h היא קבילה לפי הגדרה.
b. לא, כמו ב-a.

<u>שאלה 10</u>

1. יתרון של IDA^* ביחס ל-A^* הוא צריכת הזיכרון. כפי שראינו בתרגול, צריכת הזיכרון של IDA^* ליניארית באורך המסלול לפיתרון, בעקבות כך שזה מריץ DFS, בעוד שצריכת הזיכרון של A^* ליניארית בכמות הצמתים שפותחו. חיסרון של IDA^* ביחס ל-A^* הוא שמכיוון שהוא לא מתחזק closed, עלול לחזור למצבים שהוא כבר פיתח- וזה פוגע בזמן הריצה.

2. להשלים!
אני משלים!

איטרציה ראשונה:

Iteration	State	f - limit	new - limit	g	h	f = g+h

1	(0,F,F)	2	inf	0	2	2
1	(0,F,F)→ (0,F,F)	2	3	1	2	3
1	(0,F,F) → (1,F,F)	2	3	1	1	2
1	(1,F,F) → (1,F,F)	2	3	2	1	3
1	(1,F,F) → (2,F,F)	2	3	2	2	4
1	(1,F,F) → (0,F,F)	2	3	2	2	4
1	(1,F,F) → (5,T,F)	2	3	2	0	2
1	(5,T,F) → (6,T,F)	2	3	3	1	4
1	(5,T,F) → (9,T,F)	2	3	3	1	4
1	(5,T,F) → (4,T,F)	2	3	3	1	4
1	(5,T,F) → (1,T,F)	2	3	3	1	4
1	(0,F,F) → (4,F,F)	2	3	1	1	2
1	(4,F,F) → (4,F,F)	2	3	2	1	3
1	(4,F,F) → (8,F,F)	2	3	2	2	4
1	(4,F,F) → (5,T,F)	2	3	2	0	2
1	(5,T,F) → (6,T,F)	2	3	3	1	4
1	(5,T,F) → (9,T,F)	2	3	3	1	4
1	(5,T,F) → (4,T,F)	2	3	3	1	4
1	(5,T,F) → (1,T,F)	2	3	3	1	4
1	(4,F,F) → (0,F,F)	2	3	2	2	4

נתאר את האיטרציה השניה:

Iteration	State	f - limit	new - limit	g	h	f = g+h
2	(0,F,F)	3	inf	0	2	2
2	(0,F,F)→ (0,F,F)	3	inf	1	2	3
2	(0,F,F)→ (0,F,F)	3	4	2	2	4
2	(0,F,F) → (1,F,F)	3	4	2	1	3
2	(1,F,F) → (1,F,F)	3	4	3	1	4
2	(1,F,F) → (2,F,F)	3	4	3	2	5
2	(1,F,F) → (0,F,F)	3	4	3	2	5
2	(1,F,F) → (5,T,F)	3	4	3	0	3
2	(5,T,F) → (6,T,F)	3	4	4	1	5
2	(5,T,F) → (9,T,F)	3	4	4	1	5
2	(5,T,F) → (4,T,F)	3	4	4	1	5
2	(5,T,F) → (1,T,F)	3	4	4	1	5
2	(0,F,F) → (4,F,F)	3	4	2	1	3
2	(4,F,F) → (4,F,F)	3	4	3	1	4
2	(4,F,F) → (8,F,F)	3	4	3	2	5
2	(4,F,F) → (5,T,F)	3	4	3	0	3
2	(5,T,F) → (6,T,F)	3	4	4	1	5
2	(5,T,F) → (9,T,F)	3	4	4	1	5
2	(5,T,F) → (4,T,F)	3	4	4	1	5
2	(5,T,F) → (1,T,F)	3	4	4	1	5

2	(4,F,F) → (0,F,F)	3	4	2	2	5
2	(0,F,F) → (1,F,F)	3	4	1	1	2
2	(1,F,F) → (1,F,F)	3	4	2	1	3
2	(1,F,F) → (1,F,F)	3	4	3	1	4
2	(1,F,F) → (2,F,F)	3	4	3	2	5
2	(1,F,F) → (0,F,F)	3	4	3	2	5
2	(1,F,F) → (5,T,F)	3	4	3	0	3
2	(5,T,F) → (6,T,F)	3	4	4	1	5
2	(5,T,F) → (9,T,F)	3	4	4	1	5
2	(5,T,F) → (4,T,F)	3	4	4	1	5
2	(5,T,F) → (1,T,F)	3	4	4	1	5
2	(1,F,F) → (2,F,F)	3	4	2	2	4
2	(1,F,F) → (0,F,F)	3	4	2	2	4
2	(1,F,F) → (5,T,F)	3	4	2	0	2
2	(5,T,F) → (6,T,F)	3	4	3	1	4
2	(5,T,F) → (9,T,F)	3	4	3	1	4
2	(5,T,F) → (4,T,F)	3	4	3	1	4
2	(5,T,F) → (1,T,F)	3	4	3	1	4
2	(0,F,F) → (4,F,F)	3	4	1	1	2
2	(4,F,F) → (4,F,F)	3	4	2	1	3

2	(4,F,F) → (8,F,F)	3	4	3	2	5
2	(4,F,F) → (5,T,F)	3	4	3	0	3
2	(5,T,F) → (6,T,F)	3	4	4	1	5
2	(5,T,F) → (9,T,F)	3	4	4	1	5
2	(5,T,F) → (4,T,F)	3	4	4	1	5
2	(5,T,F) → (1,T,F)	3	4	4	1	5
2	(4,F,F) → (0,F,F)	3	4	3	2	5
2	(4,F,F) → (4,F,F)	3	4	3	1	4
2	(4,F,F) → (8,F,F)	3	4	2	2	4
2	(4,F,F) → (5,T,F)	3	4	2	0	2
2	(5,T,F) → (6,T,F)	3	4	3	1	4
2	(5,T,F) → (9,T,F)	3	4	3	1	4
2	(5,T,F) → (4,T,F)	3	4	3	1	4
2	(5,T,F) → (1,T,F)	3	4	3	1	4
2	(4,F,F) → (0,F,F)	3	4	2	2	4

<u>שאלה 11</u>

2. חיסרון של A^*_ϵ ביחס ל-A^* הוא שבהינתן פונק' מחיר חסומה מלמטה וחיובית ויוריסטיקה קבילה, A^* מבטיח פיתרון אופטימלי, אך A^*_ϵ מבטיח פיתרון אופטימלי עד כדי פקטור $1+\epsilon$, ולכן חלש יותר.
יתרון של A^*_ϵ ביחס ל-A^* הוא יותר גמישות- מאפשר לדוגמה שילוב של יוריסטיקה אחרת, או שימוש בהתפלגות מה שגורם לאקראיות בין ריצות שונות של האלגוריתם ולעיתים זה חיובי, על ידי אופי הבחירה של הצומת מתוך focal.

3. יוריסטיקה לדוגמה- h_{MSAP} כפי שהוגדרה בשאלה 7.
עבור הרצת האלגו' עם $g(v)$ קיבלנו:

```python
AStarEpsilon_agent = AStarEpsilonAgent()
actions, total_cost, expanded = AStarEpsilon_agent.search(env, epsilon=100)
print(f"Total_cost: {total_cost}")
print(f"Expanded: {expanded}")
print(f"Actions: {actions}")
```
✓ 0.0s

```
Total_cost: 103.0
Expanded: 79
Actions: [0, 0, 0, 0, 0, 0, 1, 1, 1, 2, 1, 1, 0, 1, 1, 0]
```

עבור הרצה עם h_{MSAP}:

```python
AStarEpsilon_agent = AStarEpsilonAgent()
actions, total_cost, expanded = AStarEpsilon_agent.search(env, epsilon=100)
print(f"Total_cost: {total_cost}")
print(f"Expanded: {expanded}")
print(f"Actions: {actions}")
```
✓ 0.0s

```
Total_cost: 474.0
Expanded: 64
Actions: [0, 1, 0, 0, 3, 0, 0, 0, 2, 2, 2, 1, 2, 2, 2, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 3, 3, 2, 3, 3, 0, 3, 3, 3, 2, 2, 2, 1, 2, 2, 2, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0]
```

ניתן לראות שעלות הפיתרון כאן הרבה יותר גבוהה- אבל עדיין בטווח של פקטור $1 + \epsilon$, עבור $\epsilon = 100$, מהמחיר האידיאלי שהוא 103, אך פותחו פחות nodes.

4. להגדיר $\epsilon = \infty$, גורר את זה שכל הnodes נמצאים ב-focal תמיד. כלומר, הדבר היחיד שמשנה הוא איך שבוחרים מתוך ה-focal. לדוגמה, שימוש ב-g לבחירה מה-focal היא שקולה להרצת האלגוריתם של bfs, שימוש ב-h_{MSAP} שקול להרצת אלגוריתם greedy best first search, עם היוריסטיקה הזו (h_{MSAP}).

<u>שאלה 12</u>

2. צירוף תצלום של התוצאות:

map	BFS-G cost	BFS-G expa	WA* (0.5)	WA* (0.5)	WA* (0.7)	WA* (0.7)	WA* (0.9)	WA* (0.9) expanded
map12x12	140	445	118	221	118	214	118	259
map15x15	215	858	178	664	178	646	178	679
map20x20	203	1045	188	723	188	645	188	1111

התוצאות די מפתיעות.
ראשית אתייחס לשינוי הweight- היינו מצפים לאיכות פיתרון הולכת ויורדת ומהירות הולכת וגדלה ככל שהמשקל של היוריסטיקה גדל. כאן אנו רואים שאיכות הפיתרון זהה ללא תלות במשקל, וכי יש עלייה במהירות בין 0.5 ל0.7 אך ירידה במהירות בין ה-0.7 ל-0.9.
בהשוואה ל-BFS-G התוצאות שקיבלנו הגיוניות, שכן BFS הולך שרירותית בסדר פעולות מסוים (קודם למטה וכן הלאה), ולכן הגיוני שאיכות הפיתרון שלו תהיה פחותה, וגם שכמות הצמתים שפיתח תהיה גדולה, משום שהוא סורק רוחבית.

עבור יוריסטיקה יותר מיודעת הייתי מצפה לשינוי בכמות הצמתים שפותחו- כך שתקטן, משום שנלך יותר ממוקד ישר למטרה.

<u>שאלה 13</u>

.1

$$p(d|a) = 0.2 \, , p(b|a) = 0.4 \, , p(c|a) = 0.4$$

.2

3, על ידי המסלול: $A \to B \to F \to G$, במקרה בו: $\beta > 4$.

.3

לא, מכיוון שהמקסימום הגלובלי הוא F או G כתלות בערך β, ומ-C אי אפשר להגיע לאף אחד משניהם, אלא מגיעים בהכרח ל-H.

.4

הצעד $A \to D$ מתבצע בהסתברות של 0.2, וגורר שבהכרח לא נגיע לפיתרון האופטימלי.
הצעד $A \to C$ מתבצע בהסתברות של 0.4, וגורר שבהכרח לא נגיע לפיתרון האופטימלי.
נפרק את הצעד $A \to B$, שמתבצע בהסתברות 0.4 לחלקים לפי המשך הריצה של האלגו':
אם $\beta \geqslant 4$: מ-B האלגוריתם ימשיך או ל-G, ואז נגיע לערך האופטימלי, או ל-F ואז משם בהכרח ימשיך ל-G, לכן בכל מקרה נגיע לפיתרון האופטימלי במקרה הזה.
לסיכום המקרה הזה, לא נגיע לפיתרון האופטימלי בהסתברות של 0.6.

אם $3 < \beta < 4$: מ-B האלגוריתם ימשיך או ל-G, ואז לא נגיע לערך האופטימלי, או ל-F ושם נסיים עם הערך האופטימלי. ההסתברות ללכת מ-B ל-F היא: $\frac{2}{\beta}$, ול-G היא: $\frac{\beta - 2}{\beta}$

לסיכום המקרה הזה, לא נגיע לפיתרון האופטימלי בהסתברות של $0.6 + 0.4 \cdot \frac{\beta - 2}{\beta}$.

5. אין ערכים כאלה.
ראינו לעיל שהמסלול הארוך ביותר של האלגוריתם הוא 3, ומסתיים ב-G. על מנת שנוכל להגיע לפיתרון האופטימלי תוך 3 צעדים בדיוק, מתחייב שהפיתרון האופטימלי יהיה ב-G, כלומר $\beta > 4$.
ראינו לעיל שההסתברות להגיע ל-B מההתחלה היא 0.4. לכן, על מנת שבאמת נגיע לפיתרון האופטימלי ב-3 צעדים בדיוק, נידרש לעבור דרך F ולא ישר מ-B ל-G. נשים לב שאם נגיע ל-F במקרה בו $\beta > 4$ בהכרח נמשיך ל-G, ולכן נסתכל על ההסתברות להגיע מההתחלה ל-F:
עד ל-B זה 0.4, וכעת מ-B ל-F זה: $\frac{2}{\beta}$, ולכן סה"כ מההתחלה ל-F: $\frac{2}{\beta} \cdot \frac{2}{5} = \frac{4}{5\beta} < \frac{1}{5}$
כאשר המעבר האחרון נובע מכך ש-$\beta > 4$. לכן אין מקרה בו ההסתברות להגיע לפיתרון האופטימלי בדיוק ב-3 צעדים היא גדולה מ-$\frac{1}{5}$.