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, 24.5, בשעה 23:59.
- את המטלה יש להגיש <u>בזוגות בלבד</u>.
- יש להגיש מטלות מוקלדות בלבד בעברית או באנגלית. פתרונות בכתב יד לא ייבדקו.
- ניתן לשלוח שאלות בנוגע לתרגיל <u>בפיאצה</u> בלבד.
- המתרגל האחראית על תרגיל: **רון בן שטרית**.
- בקשות דחיה <u>מוצדקות</u> (מילואים, אשפוז וכו') יש לשלוח למתרגל האחראי (**ספיר טובול**) בלבד.
- במהלך התרגיל ייתכן שנעלה עדכונים, למסמך הנ"ל – תפורסם הודעה בהתאם.
- העדכונים הינם <u>מחייבים</u>, ועליכם להתעדכן עד מועד הגשת התרגיל.
- שימו לב, התרגיל מהווה כ- 15% מהציון הסופי במקצוע <u>ולכן העתקות תטופלנה בחומרה!</u>
- 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 שבמקרה 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.

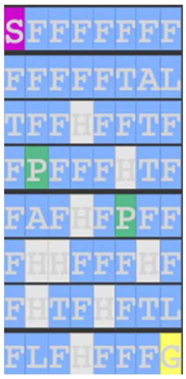
סיפור מסגרת

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.

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, O, I, G) עבור סביבת האגם הקפוא. כאשר S זה מרחב המצבים, O, זה מרחב האופרטורים, I, זה המצב ההתחלתי ו-G הוא קבוצת מצבי המטרה. מה גודל מרחב המצבים S? הסבירו.

$$S = \{0,1,2,\ldots,63\} = \{0\} \cup [63], \qquad O = \{UP, DOWN, LEFT, RIGHT\}, \qquad I = \{0\}, \qquad G = \{63\}$$

$$UP(s) = \begin{cases} s-8, & s>7 \\ s, & otherwise \end{cases}$$

$$DOWN(s) = \begin{cases} s+8, & s<56 \\ s, & otherwise \end{cases}$$

$$LEFT(s) = \begin{cases} s-1, & s \neq (8*n) \mid n \in \mathbb{N} \cup \{0\} \\ & s, otherwise \end{cases}$$

$$RIGHT(s) = \begin{cases} s+1, & s \neq (8*n-1) \mid n \in \mathbb{N} \cup \{0\} \\ & s, \quad otherwise \end{cases}$$

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

$$DOMAIN(UP) = \{s \in S \mid UP(s) \neq \emptyset\} = \{s \in S \mid \ s \neq HOLE\}$$

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

הפעלת פונקצית Succ על המצב ההתלתי תחזיר : {LEFT(0) = UP(0) = 0, DOWN(0) = 8, RIGHT(0) = 1}

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

כן, למשל אפשר ללכת ממצב 0 למצב 8 ואז לחזור בחזרה למצב 0. אפשר גם להתחיל במצב 0 לרדת למצב 8 ואז ללכת ימינה למצב 9 ולעלות למעלה למצה 1 ואז ללכת שמאלה וכך נחזור למצב 0, באופן דומה ניתן לבנות מעגלים אחרים.

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

מקדם הסיעוף בבעיה הינו b = 4 כי לכל צומת ניתן להגיע לארבע צמתים ממנו על ידי הפעלת האופרטורים.

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

במקרה הגרוע ביותר הסוכן ילך במעגלים ולא יגיע ליעד באף שלב לכן אינסוף פעולות.

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

נבחן שתי אופציות :

1) ללכת ימינה עד שמגיעים לסוף השורה ראשונה ואז לרדת למטה עד שמגיעים ליעד, זה דורש סה"כ 14 פעולות.

2) נלך מהמצב ההתחלתי עד ה PORTAL (3פעמים למטה ואז פעם ימינה) במצב 25 שיקח אותנו למצב 37 שדורש ממנו 4 פעולות. ומשם המשיך ליעד הסופי על ידי שנלך ימינה 2 פעמים ואז למטה 3 פעמים ולכן בסה"כ נבצע : 9 פעולות.

קיבלנו שללכת דרך ה PORTAL הינו יותר קצר, ולכן הדרך הקצרה ביותר היא 9 פעולות.

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

לא נכון, למשל בדוגמה של הלוח הבא:

START	PORTAL	FROZEN	FROZEN
HOLE	HOLE	HOLE	HOLE
FROZEN	GOAL_1	FROZEN	FROZEN
HOLE	HOLE	HOLE	HOLE
FROZEN	FROZEN	PORTAL	GOAL_2

מתקיים ש: MD(START,GOAL_1) = 3 ו MD(START,GOAL_2) = 7. אך כפי שמופיע בלוח אי אפשר להגיע ל GOAL_1 כי הוא מסביב לו כל המצבים הם ב HOLE כלומר המרחק אליו הוא אינסוף. לעומת זאת, כן ניתן להגיע ל GOAL_2 ולכן המרחק אליו סופי והמסלול הקל ביותר במקרה הזה הינו למצב המטרה "הרחוק יותר".

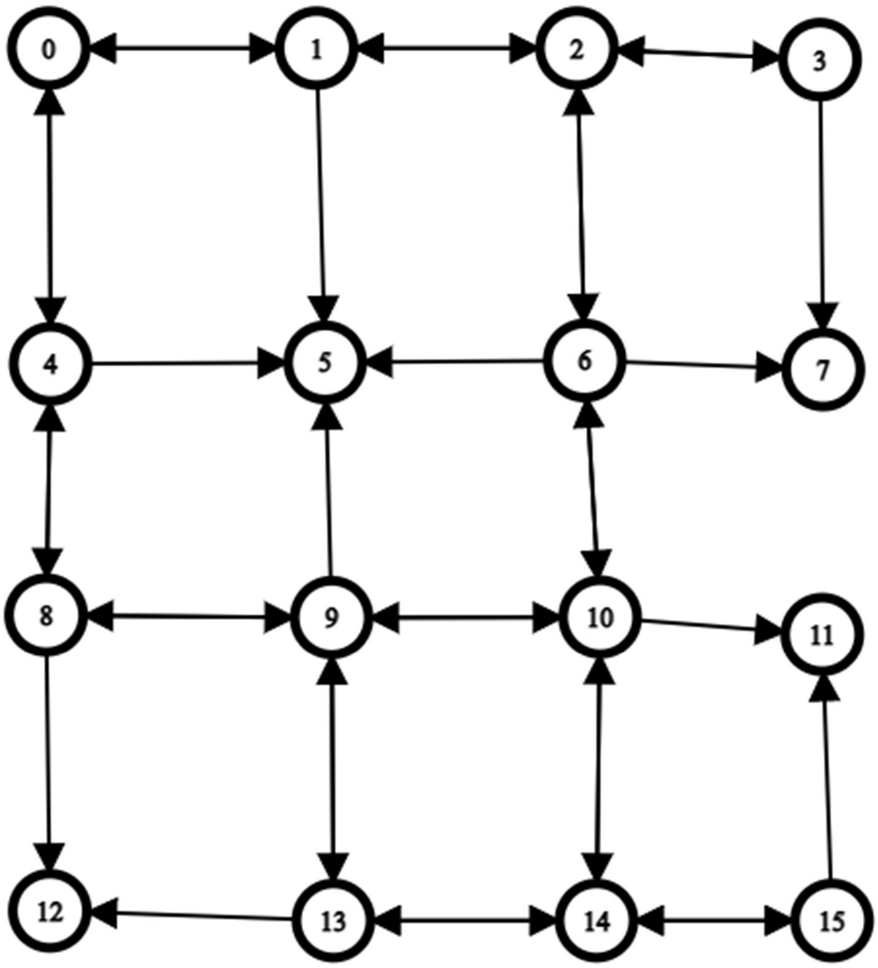
שאלה 2 – Breadth First Search-G (7 נק׳):

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

צריך שהגרף יהייה עץ מכוון כך שגם הוא וגם גרף התשתית שלו לא מכילים מעגלים(כלומר דרגת הכניסה של כל צומת לכל היותר 1).
כי אם הוא מכיל מעגל מכוון אז עץ החיפוש יהייה אינסופי ואם הוא מכיל מעגל לא מכוון אז הצומת שנכנסים אליו יותר מקשת אחת מפתחים אותו לפחות פעמים.

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



כאשר למצבים 0,1,2,3,4,8,13,14,15 יש לולאה עצמית.

4. 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 $T: G \rightarrow G'$ המקבלת את גרף המצבים G ויוצרת גרף חדש G' ובעזרתה למצוא את המסלול האופטימלי בגרף G.

רעיון : לכל קשת בגרף G נכניס קשת "שמקבילה לה" בגרף G' כך שאם לקשת יש משקל מסוים, נכניס מספר קשתות כמשקל הקשת שיעברו בצמתים שנוסיף לגרף עד שהקשת האחרונה מגיעה ליעד שהגיעה אליו ב G.

אלגוריתם:

1) $G = (V, E), G' = (V', E'), V' = E' = \emptyset$
2) לכל $e = (u, v) \in E$:
   אם weight(e = (u,v)) = 1:
   $V' = V' \cup \{u, v\},\ E' = E' \cup \{e\}$
   אחרת: (משקלה גדול מ 1)
   נסמן: *weight(e = (u,v)) = c > 1*

נוסיף לגרף החדש *c-1* צמתי ביניים : $u_1, u_2, \dots \dots u_{c-1}$ כך שבין כל 2 צמתי ביניים קשת עם משקל 1,

נחבר אותם באופן הבא : $P : u = u_0 \overset{e'_1}{\rightleftarrows} u_1 \overset{e'_2}{\rightleftarrows} u_2 \cdots u_{c-1} \overset{e'_c}{\rightleftarrows} u_c = v,$

$\forall e_i : weight(e'_i) = 1$

ונוסיף אותם לגרף החדש:

$V' = V' \cup \left(\bigcup_{i=0}^{i=c} u_i\right), E' = E' \cup \left(\bigcup_{i=1}^{i=c} e'_i\right)$

עכשיו בסיום האלגוריתם הנ"ל מתקבל לנו גרף *G'* כך שאם הריץ עליו *BFS-G* מתקבל המסלול האופטימלי כי עכשיו משקל המסלול שווה לאורכו.

5. יבש (2 נק׳): נתון לוח 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?

BFS אף פעם לא יפתח צומת רחוק לפני שהוא מפתח צומת יותר קרוב ממנו. ולכן , היות שצומת היעד הוא הרחוק ביותר מבין כל הצמתים וברגע שמפתחים אחד משני שכניו מגלים אותו ועוצרים אנחנו נפתח בדיוק $N^2 - 2$ צמתים לפני שמגלים את צומת היעד. אך יווצרו N^2 מצבים בגרף כי שני השכנים גם נוצרים.

שאלה 3 – Depth First Search-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 הינו שלם עבור בעיית האגם הקפוא כי מדובר בגרף(לא נלך במעגלים כי שומרים CLOSE) וגם כי הגרף סופי, כי הרי אם לא היה סופי DFS עלול להיתקע בלולאה אינסופית גם אם צומת היעד נמצא במרחק סופי . לעומת זאת, האלגוריתם אינו קביל כי לא בהכרח מוצאים את הפתרון האופטימלי. וזה בגלל ש- DFS מפתח צמתים לפי מדיניות LIFO ולכן אם בריצה הוא קודם מפתח צומת שמוביל למסלול ארוך יותר ליעד אז הוא יחזיר מסלול זה במקום מסלול קל יותר למרות שהוא לא אופטימלי.

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?

אלגוריתם DFS על עץ לא בהכרח היה מוצא פתרון כי בעצם העץ אינסופי בגלל שגרף החיפוש מכיל מעגלים מכוונים . ובגלל שבעץ לא שומרים צמתים שביקרנו בהם ב-CLOSE ה-DFS עלול להיתקע במסלול אינסופי שמקורו במעגל. האלגוריתם היה מנסה לרדת כמה שיותר למטה עד שיתקע או שהגיע לשורה אחרונה וכאשר ינסה להתקדם הוא ייתקע באותו מצב, או בחור ואז יבחר בכיוון אחר להתקדם בו עד שבסופו של דבר יגיע לגבולות הלוח ונתקע או מצליח אם היה לו מזל.

4. יבש (2 נק׳): נתון לוח 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 <u>יפותחו</u> <u>וייווצרו</u> במהלך חיפוש DFS-G? הסבירו?

עכשיו יפותחו 2N – 2 צמתים כי אנחנו נתחיל מצומת ההתחלה ונרד כל הדרך למטה עד השורה האחרונה במסלול ישר ואז נלך ימינה כל הדרך עד צומת היעד (העדיפות באופרטורים היא : למטה, ימינה, למעלה, שמאלה) ולכן אם נספור את המצבים בדרך, נקבל שיש N על העמודה בירידה ו N – 2 אחרים בשורה האחרונה (כי לא מפתחים היעד ויש איד שכבר ספרנו בעמודה). לכן בסך הכל מפתחים 2N – 2 צמתי בדרך.ובדרך יש נצבים שמווצרים אותם אך לא מבקרים בהם כמו למשל המצב מימין למצב ההתחלתי וכנ"ל לכל העמודה הראשוה(חוץ מהמצב האחרון בעמודה הראשונה שכן מבקרים לימינו) ובאופן דומה עבור השורה האחרונה מייצרים את המצב שמעל להם ולכן בסך הכל מייצרים : 4N-5 מצבים.

5. יבש (2 נק׳): נתון לוח 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 <u>יפותחו</u> <u>וייווצרו</u> במהלך חיפוש DFS-G backtracking? הסבירו?

עכשיו מווצרים רק את המצב הבא בפונקציית ()Succ ולכן בעצם לא מייצרים את המצבים "בסביבה הנוספים" ומווצרים רק אלו שבמסלול ולכן מפתחים ומייצרים רק את הצמתים על המסלול שהן 2N – 2 כפי שהסברנו בסעיף הקודם.

שאלה 4 – DFS-L (6 נק׳):

1. יבש (6 נק׳): ג׳רי רוצה למצוא מסלול בסביבת האגם הקפוא עם DFS-L. ידוע כי אורך המסלול הקצר ביותר לצומת מטרה הוא d אך ריק מגביל את החיפוש של ג׳רי לעומק $\frac{d}{2}$.

a. יבש (2 נק׳): הציעו שינוי **לבעיית החיפוש** (S,O,I,G) כך שג׳רי יוכל למצוא פתרון מבלי להפר את הגבלת העומק שריק הטיל עליו. הסבירו למה כעת ניתן למצוא פתרון.

האינטואציה לפתרון היא להגדיר אופרטורים חדשים כך שיתקדמו פי 2 יותר מהר שזה אומר במקום להתקדם משבצת אחת נתקדם שניים אך העומק יגדל רק ב 1. ובזאת כאשר נגיע לעומק $\frac{d}{2}$ אנחנו בפועל הגענו לעומק של d כי כל פעולה שביצענו בדרך מקבילה לשתי פעולות במרחב האוריגינלי. כאשר מתקיים: (S, O_{new}, I, G) ש:
$O_{new} = \{UP, DOWN, LEFT, RIGHT, UU, RR, LL, DD, UR, UL, DL, DR\}$ שאלו האופרטורים שמתקדמים יותר מהר אך גם השארנו האופרטורים המקוריים כי ייתכן שנהיה במצב שצריך לזוז משבצת אחת בלבד כדי שנגיע ליעד. עשיו אנחנו מגלים את הפתרון עם הגבלת העומק ל $\frac{d}{2}$ בגלל שיש לנו צעדים שמתקדמים שניים קדימה אך מגדילים את העומק ב 1 לכן נוכל להגיע לצמתי יעד שבמרחק d מסיבה זאת ולמצוא פתרון.
הסבר אופרטורים:
UU – זז שנים למעלה. *DD* – זז שניים למטה. *UR*- זז אחת למעלה ואחת ימינה.
UL – זז אחת למעלה ו אחת שמאלה. *DR* – זז אחת למטה ואחת ימינה. *DL* – זז אחת למטה ואחת שמאלה.
LL – זז שניים למטה. *RR* – זז שניים ימינה.

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

מקדם הסיעוף השתנה עכשיו כיוון שבפונקצית Succ() עכשיו יהיו לנו 12 בנים במקום 4 בנים. ועכשיו מתקיים שמקדם הסיעוף החדש מקיים: $b' = 3 * b$

c. יבש (1נק׳): מה סיבוכיות הזמן והמקום החדשים? ענו במונחים של b, d והשוו את התשובה לDFS-L רגיל עם עומק d. כיצד תשובתכם היתה משתנה עם היינו משתמשים בDFS-L עם בקטרקינג?

כרגע סיבוכיות הזמן הינה: $O((3b)^{\frac{d}{2}})$ וסיבוכיות המקום הינה: $O(b' \cdot d) = O(b \cdot d)$.
נשים לב שזו סיבוכיות זמן יחסית יותר טובה מ $O(b^d)$– אם מניחים כי $b > 3$ וזה בגלל ש –

$$\frac{b}{3} > 1 \rightarrow \lim_{d \to inf} \frac{b^{\frac{d}{2}}}{3^{\frac{d}{2}}} = \text{inf}, \rightarrow \ (3b)^{\frac{d}{2}} = o(b^d)$$

והעובדה שמריצים אותו עם חסם על העומק ששווה ל $\frac{d}{2}$ היא שקובעת את הסיבוכיות באלגוריתם זה.
כדי לקבל את היתרון בצמצום סיבוכיות המקום שמאפיין את שיטת הבקטרקינג נצטרך להריץ את האלגוריתם על עץ החיפוש ולא על הגרף כי אז לא נצטרך לשמור בצד כל צומת שמבקרים בו ב-*CLOSED*, ואז אם נשתמש בשיטה זו נקבל בכל איטירציה של *DFS-L* סיבוכיות מקום של $O(d/2) = O(d)$ וזה בגלל שאלגוריתם עם בקטרגינג אינו מפתח את הכל הבנים האחרים כשיורד ברקורסיה באחד מהם, מפתח כל בן רק כשמחליט לרדת בו, ולכן סיבוכיות המקום מוגבלת לעומק הרקורסיה שכמבן חסום ע"י עומק החיפוש של האיטירציה האחרונה.

d. יבש (2 נק׳): ספקו דוגמא לבעיה שבה DFS-L במרחב החיפוש החדש (לאחר השינויים שביצעתם בa) יותר טובה מאשר DFS-L במרחב החיפוש הקודם ודוגמה לבעיה שבה DFS-L במרחב המקורי עדיף. בתשובתכם התייחסו למספר צמתים שפותחו. דוגמאות יכולות להיות כלליות ולא בהכרח מסביבת הfrozen lake .

דוגמה שבה המרחב החדש יותר טוב:
נסתכל על הלוח הבא:-

S = 0	Hole = 1	F	3
F = 4	Hole = 5	F = 6	..
G	F = 9	F = 10	..

..	..	..	..

ונניח כי סדר הפעולות בו עובר DFS על הבנים הוא : ...... -> DD -> UR -> DL
וכי בגרף המקורי הסדר הוא: ... D -> R -> L -> U
במקרה הזה צומת היעד G הוא בעצם צומת שכן למצב ההתחלתי בגרף של המרחב החדש ורחוק ממנו צעד אחד אמנם הוא רחוק 2 צעדים בגרף המקורי כלומר אם נריץ DFS-2 למשל נגיע ישר לצומת היעד בגרף החדש אך בגרף המקורי נצטרך לפתח 2 צמתים לפחות עד שנגיע.

דוגמה שבה המרחב הישן עדיף:
נסתכל על הלוח הבא:-

S = 0	F = 1	F = 2
G = 3	F = 4	F = 5
F = 6	F = 7	F = 8
F = 9	F = 10	F = 11

ונניח כי סדר המעבר על הבנים בשני האלגוריתמים הוא אותו סדר מהדוגמה הקודמת, אזי האלגוריתם על הגרף המקורי יפתח צומת ויגלה דרכה ישירות את צומת היעד,
לעומת זאת, האלגוריתם החדש לפחות יצטרך לפתח את צומת 6 ו-4 שזה כבר יותר מהמקורי..

שאלה 5 – ReverseDFS (6 נק׳):

1. יבש (6 נק׳): הניחו כי יש לנו ידע מקדים על חסם עליון למרחק למצב מטרה, נסמן 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
```

בשאלות הבאות הניחו כי יש מספיק זמן לסיום האיטרציה הראשונה.

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

עכשיו אם מניחים שהחסם על עומק הפתרון הינו נכון אז נקבל שכאשר נריץ DFS-L עם עומק D נמצא פתרון ועצם זה שאנחנו מקטינים את העומק ומחפשים שוב אינו פוגע בשלמותו כי כאשר מגיעים לעומק שלא מוצאים בו פתרון מחזירים הפתרון הקודם שמצאנו ומובטח שהוא קיים מכיוון שיש לנו ידע קודם על עומק הפתרון שנמצא עד עומק D.(כמובן כל זה בהנחה שיש מספיק זמן ריצה עבור האיטרציה הראשונה לסיים כי אחרת האלגוריתם נקטע לפני שהוא מוצא פתרון לבעיה)

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

אם נניח שיש לנו מספיק זמן ריצה, אז נקבל שהאלגוריתם הינו אופטימלי כי כל פעם אנחנו מנסים למצוא את הפתרון שנמצא בעומק L-1 ולכן אם קיים פתרון אז נמצא אותו וכיוון שנעשה זאת עד שנגיע לעומק שאי אפשר למצוא בו פתרון אנחנו בהכרח נמצא הפתרון האופטימלי.

c. (2 נק׳) ספקו דוגמה בה ReverseDFS עדיף על ID-DFS ודוגמה בה ID-DFS עדיף על ReverseDFS. הדוגמאות יכולות להיות כלליות ולא בהכרח מסביבת התרגיל.

1. דוגמה שבה ReverseDFS יותר יעיל: נניח כי הפתרון נמצא בעומק 1000 וכי ReverseDFS מתחיל מ- L = 1000 אז הוא יצטרך להריץ רק 2 איטירציות של DFS שהן: DFS 999 <- DFS -1000
לעומת זאת, IterativeDFS יצטרך להריץ 1000 איטירציות שכמובן בהן יפתחו יותר צמתים מ-Reverese
2 . דוגמה מקבילה היא כשהפתרון נמצא בעומק 2 למשל ואז iterative ימצא אותו יותר מהר מאשר Reverse יריץ 1000 איטירציות.

d. (2 נק׳) הציעו כיצד ניצן לייעל את האלגוריתם. רמז: האם אתם יכולים לחשוב על צעד עדכון עדיף לL?

חיפוש בינארי

נקצה מערך בגודל L (העומק ההתחלתי) שיעזור לנו לחפש את העומק האופטימלי באופן לדומה לחיפוש בינארי. אנחנו מחפשים את העומק 'L האופטימלי שבו 'DFS-L מוצא את הצומת ומחזיר מסלול אך 'DFS-L-1 אינו מוצא את הצומת (שזה כמובן גורר שהמרחק האופטימלי הוא 'L), נשים לב שאם הפתרון נמצא בעומק 'L אז הוא נמצא בכל עומק גדול מ-'L , ולכן כדי למצוא את העומק שמקיים את זה נתחיל מ- L/2 (אמצע המערך)

בכל שלב נבדוק אם L הנוכחי :

- צומת היעד נמצא בו:
    - נבדוק אם הוא לא נמצא ב-(L-1) , אם כן נעצור.

- אחרת, העומק שאנחנו מחפשים הוא בוודאי קטן מ-L , נחפש בחצי התחתון
- צומת היעד לא נמצא בו:
  - נבדוק אם צומת היעד נמצא ב- L+1 , אם כן נעצור.
  - אחרת, העומק האופטימלי בוודאי נמצא בחצי העליון , נמשיך משם.

- בדרך הזו נוכל למצוא את העומק האופטימלי אחרי $O(\log_2 L)$ איטרציות, ולכן זו שיטה יותר יעיל מפשוט לרדת כל פעם ב-1.

שאלה 6 - 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.

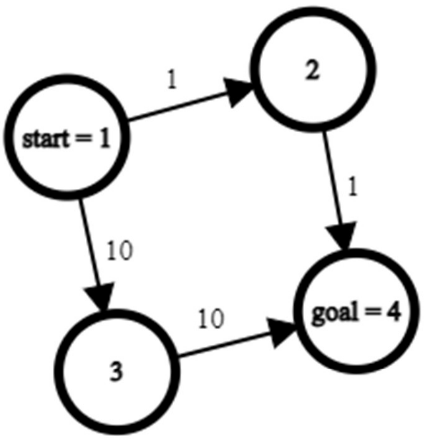
עבור בעיות בהן משקל הקשתות הינו אחיד בכל רמה מרמות הגרף. UCS תמיד יבחר בקשת הזולה ביותר ברמה וכיוון שכולם באותו משקל אז נעבור על כולן כמו BFS.

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

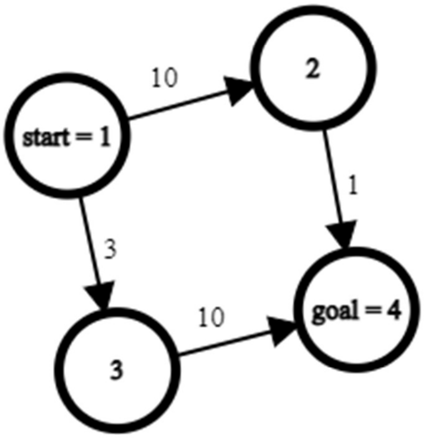
האלגוריתם הינו שלם כיוון שכל המשקלים על הקשתות בגרף שלנו הינם חיוביים ממש וחסומים מלמטה על ידי 1 נקבל שלפי מה שראינו בכיתה האלגוריתם הוא שלם וגם הגרף שלנו הוא סופי. והוא קביל גם כן כי מחזיר המסלול הזול ביותר כאשר מגלה צומת היעד.

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.

ניקח את הדוגמה הבאה:



כך שנתחיל מצומת 1 ועבור שני האלגורתמים יוחזר שאורך המסלול הקל ביותר הינו 2 ויוחזר המסלול מ 1 ל 2 ל 4. אך זה רק במקרה כי למשל בגרף הבא:



הפעלת UCS תחזיר לנו שהמסלול הקל ביותר הינו מ 1 ל 2 ל 4 ומשקלו שווה ל 11. אך הפעלת אלגורימו של דן יחזיר לנו את המסלול 1 ל 3 ל 4 עם משקל מסלול 13. וזה בגלל שהוא לא בודק אם הוא גילה שיפור למסלול ו בודק אם הוא הגיע למצב יעד כאשר פיתח אותו.

שאלה 7 - יוריסטיקות (8 נק׳):

1. יבש (1 נק׳): בהינתן שתי יוריסטיקות קבילות h_1, h_2. האם $h = \min\{h_1, h_2\}$ קבילה? אם כן, הוכיחו. אם לא, הפריכו.

יוריסטיקה h קבילה אם: $\forall s \in S: 0 \le h(s) \le h^*(s)$
כיוון ש: h_1, h_2 קבילות מתקיים: $0 \le h_1(s) \le h^*(s), 0 \le h_2(s) \le h^*(s)$ ובפרט מתקיים ש :
$0 \le \min\{h_1(s), h_2(s)\} \le h_1(s) \le h^*(s)$ לכן נתקבל ש: $\min\{h_1, h_2\}$ קבילה.

2. יבש (1 נק׳): בהינתן שתי יוריסטיקות קבילות h_1, h_2. האם $h = \max\{h_1, h_2\}$ קבילה? אם כן, הוכיחו. אם לא, הפריכו.

יוריסטיקה h קבילה אם: $\forall s \in S: 0 \le h(s) \le h^*(s)$
כיוון ש: h_1, h_2 קבילות מתקיים: $0 \le h_1(s) \le h^*(s), 0 \le h_2(s) \le h^*(s)$ ונניח בה"כ שמתקיים : $h_1(s) \ge h_2(s)$ עבור s
כלשהו, $0 \le h_1(s) = \ max\{h_1(s), h_2(s)\} \le h^*(s)$ והמקרה השני שבו $h_1(s) \le h_2(s)$ מקבלים בו :
$0 \le h_2(s) = \ max\{h_1(s), h_2(s)\} \le h^*(s)$ לכן נקבל ש: $\max\{h_1, h_2\}$ קבילה בכל מקרה.

3. יבש (1 נק׳): בהינתן שתי יוריסטיקות עקביות h_1, h_2. האם $h = \min\{h_1, h_2\}$ עקבית? אם כן, הוכיחו. אם לא, הפריכו.

היוריסטיקה h המינימלית אכן עקבית:
תהי קשת (v,u), צ"ל: $\forall u \in S, \forall v \in S: h(v) - h(u) \le cost((v,u))$
h_1, h_2 עקביות ולכן מתקיים: $h_1(v) - h_1(u) < cost \ and \ h_2(u) - h_2(v) < cost$
נניח בה"כ כי $h(v) = \min\{h_1(v), h_2(v)\} = h_1(v)$ וכי $h(u) = \min\{h_1(u), h_2(u)\} = h_2(u)$
אזי : $h(v) - h(u) = h_1(v) - h_2(u) < h_2(v) - h_2(u) < cost$ וזה מה שרצינו להוכיח.

4. יבש (1 נק׳): בהינתן שתי יוריסטיקות עקביות h_1, h_2. האם $h = \max\{h_1, h_2\}$ עקבית? אם כן, הוכיחו. אם לא, הפריכו.

היוריסטיקה h המקסימלית אכן עקבית:
תהי קשת (v,u), צ"ל: $\forall u \in S, \forall v \in S: h(v) - h(u) \le cost((v,u))$
h_1, h_2 עקביות ולכן מתקיים: $h_1(v) - h_1(u) < cost \ and \ h_2(u) - h_2(v) < cost$
נניח בה"כ כי $h(v) = \max\{h_1(v), h_2(v)\} = h_1(v)$ וכי $h(u) = \max\{h_1(u), h_2(u)\} = h_2(u)$
אזי : $h(v) - h(u) = h_1(v) - h_2(u) < h_1(v) - h_1(u) < cost$ וזה מה שרצינו להוכיח.

נשים לב כי במקרה בו בשני הצמתים היוריסטיקה המקסימלית\המינימלית היא אחת היוריסטיקות אז היוריסטיקה החדשה תקיים את התנאי כי אז תהייה שווה לאחת מהן.

נגדיר יוריסטיקה חדשה עבור בעיות עם מצב מטרה יחיד $|G| = 1$:

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

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

5. יבש (1 נק׳): האם היוריסטיקה h_{SAP} קבילה על כל לוח? אם כן הסבר, אם לא הבא קודמה נגדית.
היוריסטיקה אכן קבילה כי בהינתן צומת v כלשהו בגרף, קיימים 2 מקרים:
- או שהמסלול האופטימלי בינו לבין g עובר דרך פורטל ואז הוא לפחות 100 ואז זה כמובן גדול\שווה לערך היוריסטי של v,
- או שאינו עובר דרך פורטל, ואז המשקל שלו גדול או שווה ל- manhattan distance כי בבעיה שלהו אין מעבר על האלכסונים וכי המחיר של כל קשת גדול\שווה ל- 1. ואז בפרט, היוריסטיקה שלנו קטנה ממשקל זה ולכן קבילה.

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

נחלק הבעיה לתתי מקרים:

1) $h_{Manhatan}(u,g) = h_{SAP}(u) < 100, h_{Manhatan}(v,g) = h_{SAP}(v) < 100$ נקבל

נניח בשלילה שהיוריסטיקה אינה עקבית, נקבל שקיימים $u,v \in S$ כך ש: $h_{SAP}(u) - h_{SAP}(v) > cost(\ (u,v)\ )$

נחלק למקרים לפי $cost(\ (u,v)\ )$:

1) $cost(\ (u,v)\ ) = 1$: מתקיים ש מצב *v* הוא משבצת מסוג *L* ו *u* היא שכנה שלה (מעליה, מתחתיה, מימינה, משמאלה) כי ניתן להגיע למשבצת מסוג זה רק על ידי שכניה. ולכן ההפרש בין מרחקי המנהטן שלהן הוא 1. ובמקרה שהיוריסיטה שווה לעלות ה *PORTAL* ההפרש שווה לאפס, ובמקרה שהיוריסטיקה של אחת מחושבת לפי מרחק מנהטן והשני לפי עלות ה *PORTAL* ההפרש המתקבל הינו לכל היותר 1 כי ההפרש במרחקי מנהטן ביניהם שווה ל 1 ומתקבל ש:
$1 = h_{SAP}(u) - h_{SAP}(v) > cost(\ (u,v)\ ) = 1$ וזו כמובן סתירה להנחת השלילה ולכן היורסטיקה עקבית.

2) $cost(\ (u,v)\ ) = 2$: מתקיים ש מצב *v* הוא משבצת מסוג *A* ו *u* היא שכנה שלה (מעליה, מתחתיה, מימינה, משמאלה) כי ניתן להגיע למשבצת מסוג זה רק על ידי שכניה. ולכן ההפרש בין מרחקי המנהטן שלהן הוא 1. ובמקרה שהיוריסיטה שווה לעלות ה *PORTAL* ההפרש שווה לאפס, ובמקרה שהיוריסטיקה של אחת מחושבת לפי מרחק מנהטן והשני לפי עלות ה *PORTAL* ההפרש המתקבל הינו לכל היותר 1 כי ההפרש במרחקי מנהטן ביניהם שווה ל 1 ומתקבל ש:
$1 = h_{SAP}(u) - h_{SAP}(v) > cost(\ (u,v)\ ) = 2$ וזו כמובן סתירה להנחת השלילה ולכן היורסטיקה עקבית.

3) $cost(\ (u,v)\ ) = 3$: מתקיים ש מצב *v* הוא משבצת מסוג *T* ו *u* היא שכנה שלה (מעליה, מתחתיה, מימינה, משמאלה) כי ניתן להגיע למשבצת מסוג זה רק על ידי שכניה. ולכן ההפרש בין מרחקי המנהטן שלהן הוא 1. ובמקרה שהיוריסיטה שווה לעלות ה *PORTAL* ההפרש שווה לאפס, ובמקרה שהיוריסטיקה של אחת מחושבת לפי מרחק מנהטן והשני לפי עלות ה *PORTAL* ההפרש המתקבל הינו לכל היותר 1 כי ההפרש במרחקי מנהטן ביניהם שווה ל 1 ומתקבל ש:
$1 = h_{SAP}(u) - h_{SAP}(v) > cost(\ (u,v)\ ) = 3$ וזו כמובן סתירה להנחת השלילה ולכן היורסטיקה עקבית.

4) $cost(\ (u,v)\ ) = 10$: מתקיים ש מצב *v* הוא משבצת מסוג *F* ו *u* היא שכנה שלה (מעליה, מתחתיה, מימינה, משמאלה) כי ניתן להגיע למשבצת מסוג זה רק על ידי שכניה. ולכן ההפרש בין מרחקי המנהטן שלהן הוא 1. ובמקרה שהיוריסיטה שווה לעלות ה *PORTAL* ההפרש שווה לאפס, ובמקרה שהיוריסטיקה של אחת מחושבת לפי מרחק מנהטן והשני לפי עלות ה *PORTAL* ההפרש המתקבל הינו לכל היותר 1 כי ההפרש במרחקי מנהטן ביניהם שווה ל 1 ומתקבל ש:
$1 = h_{SAP}(u) - h_{SAP}(v) > cost(\ (u,v)\ ) = 10$ וזו כמובן סתירה להנחת השלילה ולכן היוריסטיקה עקבית.

5) $cost(\ (u,v)\ ) = 100$: מתקיים ש מצב *v* הוא משבצת מסוג *P* ו *u* היא הקצה השני של ה *PORTAL* ולכן:
אם לאחד מהם יש $h_{Manhatan}(u) < 100, h_{Manhatan}(v) > 100$ נקבל ש:
$h_{SAP}(u) = h_{Manhatan}(u) < 100, h_{SAP}(v) = 100$ ולכן כיוון שהיוריסטקה שלנו היא אי-שלילית ההפרש ביניהם הינו קטן / שווה ל 100 כלומר קיבלנו: $100 \geq h_{SAP}(u) - h_{SAP}(v) > cost(\ (u,v)\ ) = 100$ וזו כמובן סתירה להנחת השלילה ולכן היוריסטיקה עקבית.
ואם לשניהם יש $h_{Manhatan}(u) < 100, h_{Manhatan}(v) < 100$ אז מתקבל (מארתמיטקה של שני מספרים):
$100 \geq h_{SAP}(u) - h_{SAP}(v) > cost(\ (u,v)\ ) = 100$ וזו כמובן סתירה להנחת השלילה ולכן היורסטיקה עקבית.
ואם לשניהם יש $h_{Manhatan}(u) > 100, h_{Manhatan}(v) > 100$ אז מתקבל:
$h_{SAP}(u) = 100, h_{SAP}(v) = 100$ לכן, $0 = h_{SAP}(u) - h_{SAP}(v) > cost(\ (u,v)\ ) = 100$ וזו כמובן סתירה להנחת השלילה ולכן היוריסטיקה עקבית.

6) $cost(\ (u,v)\ ) = infinity$ זה מקרה שלא ייתכן כיוון שזה אומר שאנחנו במצב הבא:
$infinity > h_{SAP}(u) > infinity = cost(\ (u,v)\ ) + h_{SAP}(v)$ וזו כמובן סתירה להנחת השלילה ולכן היורסטיקה עקבית.

נכליל את היוריסטיקה לבעיות עם מספר מצבי מטרה על ידי:

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

שימו לב שבמקרה זה אנחנו לוקחים את המינימום על פני כל צמתי היעד.

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

מתקיים שהיוריסטיקה הינה קבילה, נחלק ההוכחה למקרים:

1) $\exists g \in G: h_{MSAP}(s) = h_{Manhatan}(s,g)$ ולכן ניתן להסיק שמתקיים : $h_{MSAP}(s) \leq 100$ לפי הגרתה של $h_{MSAP}(s)$ ולפי הגרת מרחק מנהטן מתקיים שהוא אי-שלילי. בנוסף לכך כדי לעבור ממצב s למצב g בלוח ללא PORTALS צריך לעבור בדרך לפחות משבצות כמו המרחק מנהטן (כאשר ערך כל משבצת חסום מלמטה ע"י 1). ובלוח עם PORTAS צריך לעבור מרחק לפחות כמו מרחק מנהטן או לעבור דרך PORTAL ואז לעבור מספר כלשהו של צעדים עד היעד כאשר העלות של מעבר ב PORTAL הינו 100 לכן מתקבל : $\exists g \in G: 0 \leq h_{MSAP}(s) = h_{Manhatan}(s,g) \leq h^*(s)$ כלומר היא קבילה.

2) $h_{MSAP}(s) = Cost(P) = 100$ ואז ניתן להסיק שמתקיים: $h_{Manhattan}(s,g) \geq 100$ לכל $g \in G$. לכן נקבל שבלוח ללא PORTALS צריך לעבור מספר משפצות לפחות כמו מרחק המנהטן (כאשר ערכה של כל משבצת חסום מלמטה על ידי 1). ובלוח עם PORTALS צריך לעבור ב PORTAL שהעלות שלה היא 100 ואז עוד כמה צעדים עד שמגיעים ליעד. או לעבור מספר צעדים לפחות כמו מרחק המנהטן שהוא גדול/שווה ל 100 ולכן בכל מקרה מקבלים ש:

$0 \leq h_{MSAP}(s) = Cost(P) = 100 \leq h^*(s)$ כלומר שהיוריסטיקה קביל.

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

מתקיים שהיוריסטיקה הינה עקבית, נניח בשלילה שהיא לא עקבית כלומר קיימים:
$u \in S, v \in Succ(u): h_{MSAP}(u) - h_{MSAP}(v) > Cost\ (\ (u,v)\ )$ נחלק למקרים לפי $Cost\ (\ (u,v)\ )$:

1) $cost(\ (u,v)\ ) = 1$: מתקיים ש מצב *v* הוא משבצת מסוג *L* ו *u* היא שכנה שלה (מעליה, מתחתיה, מימינה, משמאלה) כי ניתן להגיע למשבצת מסוג זה רק על ידי שכניה. אם לשניהם מחזבים את היורסטיקה לפי העלות של ה *PORTAL* אז ההפרש שווה לאפס, ואם לאחד מהם מחשבים את העלות לפי ה *PORTAL* ולשני לפי מרחק מנהטן אז מתקיים שההפרש ביניהם שווה ל 1 כיוון שהם שכנים על הלוח ולכן מרחק מנהטן של השני חייה להיות 99. ואם מחשבים לשניהם לפי מרחק מנהטן מתקיים שקיימים $g_1, g_2 \in G$ כך ש : $h_{MSAP}(u) = h_{Manhattan}(u,g_1), h_{MSAP}(v) = h_{Manhattan}(v,g_2)$ מתקבל שההפרש : $|h_{Manhattan}(u,g_2) - h_{Manhattan}(u,g_1)| \leq 1$ ו $|h_{Manhattan}(v,g_1) - h_{Manhattan}(v,g_2)| \leq 1$
מתקבל ש: $h_{Manhattan}(u,g_1) - h_{Manhattan}(v,g_2) \leq 1$ לכן בסך הכל:
$1 \geq h_{MSAP}(u) - h_{MSAP}(v) > cost(\ (u,v)\ ) = 1$ וזו כמובן סתירה להנחת השלילה ולכן היורסיטקה עקבית.

2) $cost(\ (u,v)\ ) = 2$: מתקיים ש מצב *v* הוא משבצת מסוג *A* ו *u* היא שכנה שלה (מעליה, מתחתיה, מימינה, משמאלה) כי ניתן להגיע למשבצת מסוג זה רק על ידי שכניה. אם לשניהם מחזבים את היורסטיקה לפי העלות של ה *PORTAL* אז ההפרש שווה לאפס, ואם לאחד מהם מחשבים את העלות לפי ה *PORTAL* ולשני לפי מרחק מנהטן אז מתקיים שההפרש ביניהם שווה ל 1 כיוון שהם שכנים על הלוח ולכן מרחק מנהטן של השני חייה להיות 99. ואם מחשבים לשניהם לפי מרחק מנהטן מתקיים שקיימים $g_1, g_2 \in G$ כך ש : $h_{MSAP}(u) = h_{Manhattan}(u,g_1), h_{MSAP}(v) = h_{Manhatta}\ (v,g_2)$ מתקבל שההפרש : $|h_{Manhatta}\ (u,g_2) - h_{Manhattan}(u,g_1)| \leq 1$ ו $|h_{Manhatta}\ (v,g_1) - h_{Manhatta}\ (v,g_2)| \leq 1$
מתקבל ש: $h_{Manhatta}\ (u,g_1) - h_{Manhatta}\ (v,g_2) \leq 1$ לכן בסך הכל:
$1 \geq h_{MSAP}(u) - h_{MSAP}(v) > cost(\ (u,v)\ ) = 2$ וזו כמובן סתירה להנחת השלילה ולכן היורסיטקה עקבית.

3) $cost(\ (u,v)\ ) = 3$: מתקיים ש מצב *v* הוא משבצת מסוג *T* ו *u* היא שכנה שלה (מעליה, מתחתיה, מימינה, משמאלה) כי ניתן להגיע למשבצת מסוג זה רק על ידי שכניה. אם לשניהם מחזבים את היורסטיקה לפי העלות של ה *PORTAL* אז ההפרש שווה לאפס, ואם לאחד מהם מחשבים את העלות לפי ה *PORTAL* ולשני לפי מרחק מנהטן אז מתקיים שההפרש ביניהם שווה ל 1 כיוון שהם שכנים על הלוח ולכן מרחק מנהטן של השני חייה להיות 99. ואם מחשבים לשניהם לפי מרחק מנהטן מתקיים שקיימים $g_1, g_2 \in G$ כך ש : $h_{MSAP}(u) = h_{Manhattan}(u,g_1), h_{MSAP}(v) = h_{Manhatta}\ (v,g_2)$ מתקבל שההפרש : $|h_{Manhattan}(u,g_2) - h_{Manhatta}\ (u,g_1)| \leq 1$ ו $|h_{Manhatta}\ (v,g_1) - h_{Manhatta}\ (v,g_2)| \leq 1$
מתקבל ש: $h_{Manhattan}(u,g_1) - h_{Manhattan}(v,g_2) \leq 1$ לכן בסך הכל:
$1 \geq h_{MSAP}(u) - h_{MSAP}(v) > cost(\ (u,v)\ ) = 3$ וזו כמובן סתירה להנחת השלילה ולכן היורסיטקה עקבית.

4) $cost(\ (u,v)\ ) = 10$: מתקיים ש מצב *v* הוא משבצת מסוג *F* ו *u* היא שכנה שלה (מעליה, מתחתיה, מימינה, משמאלה) כי ניתן להגיע למשבצת מסוג זה רק על ידי שכניה. אם לשניהם מחזבים את היורסטיקה לפי העלות של ה *PORTAL* אז ההפרש שווה לאפס, ואם לאחד מהם מחשבים את העלות לפי ה *PORTAL* ולשני לפי מרחק מנהטן אז מתקיים שההפרש ביניהם שווה ל 1 כיוון שהם שכנים על הלוח ולכן מרחק מנהטן של השני חייה להיות 99. ואם מחשבים לשניהם לפי מרחק מנהטן מתקיים שקיימים $g_1, g_2 \in G$ כך ש : $h_{MSAP}(u) = h_{Manhatta}\ (u,g_1), h_{MSAP}(v) = h_{Manhattan}(v,g_2)$ מתקבל שההפרש : $|h_{Manhatta}\ (u,g_2) - h_{Manhatta}\ (u,g_1)| \leq 1$ ו $|h_{Manhattan}(v,g_1) - h_{Manhattan}(v,g_2)| \leq 1$

מתקבל ש: $h_{Manhattan}(u,g_1) - h_{Manhatta}\ (v,g_2) \leq 1$ לכן בסך הכל:
$10 = cost(\ (u,v)\ ) > h_{MSAP}(u) - h_{MSAP}(v) \geq 1$ וזו כמובן סתירה להנחת השלילה ולכן היורסטיקה עקבית.

5) $cost(\ (u,v)\ ) = 100$: מתקיים שמצב *v* הוא משבצת מסוג *P* ו *u* היא הקצה השני של ה *PORTAL* ולכן:
אם לשניהם מחשבים את היורסטיקה לפי העלות של ה*PORTAL* מקבלים שההפרש הינו 0. זו כמובן סתירה להנחת השלילה.
ואם מחשבים לשניהם לפי מרחק מנהטן, מתקיים שקיימים $g_1, g_2 \in G$ כך ש :
$h_{MSAP}(u) = h_{Manhatta}\ (u,g_1), h_{MSAP}(v) = h_{Manhattan}(v,g_2)$ מתקבל שההפרש :
$|h_{Manhattan}(v,g_1) - h_{Manhattan}(v,g_2)| \leq 1$ ו $|h_{Manhatta}\ (u,g_2) - h_{Manhattan}(u,g_1)| \leq 1$
מתקבל ש: $h_{Manhattan}(u,g_1) - h_{Manhatta}\ (v,g_2) \leq 1$ לכן בסך הכל:
$100 = cost(\ (u,v)\ ) > h_{MSAP}(u) - h_{MSAP}(v) \geq 1$ זו כמובן סתירה להנחת השלילה.
ואם מחשבים לאחד מהם לפי מרחק מנהטן ולשני לפי העלות של ה *PORTAL* מקבלים שההפרש כם הינו לכל היותר 1 מאותה סיבה על ההפרשים למעלה ולכן בסך הכל מקבלים : $100 = cost(\ (u,v)\ ) > h_{MSAP}(u) - h_{MSAP}(v) \geq 1$ ובכל מקרה מגיעים לסתירה להנחת השלילה לכן היורסטיקה עקבית!

6) $cost(\ (u,v)\ ) = infinity$ זה מקרה שלא ייתכן כיוון שזה אומר שאנחנו במצב הבא:
$h_{MSAP}(v) + cost(\ (u,v)\ ) = infinity > h_{SAP}(u) > infinity$ וזו כמובן סתירה להנחת השלילה ולכן היוריסטיקה עקבית.

שאלה 8 – 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?

בבעיה שלנו האלגוריתם הוא שלם כי יש מרחב חיפוש סופי אך הוא אינו קביל כי היוריסטיקה לא בהכרח תהייה מושלמת ויכולה להטעות אותנו.

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

<u>יתרון של Greedy לעומת Beam:</u> Greedy עלול לקבל פתרון יותר טובמזה של Beam כי הוא לא זורק צמתים "מיותרים" ומשאיר את כולם גם אם הערך היוריסטי שלהם גבוה.
<u>חסרון של Greedy לעומת Beam:</u> Greedy עלול להשתמש ביותר זכרון כי הוא לא זורק את הצמתים שהוערכו כרחוקים מהפתרון וכתוצאה מזאת יבזבז יותר זכרון במקרה שיש הרבה צמתים רחוקים ששמורים וגם greedy פחות יעיל בזמן הריצה כי בכל צעד הוא סורק ערימה גדולה יותר.

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 $w_1 < w_2 \leq 1$, נסמן את המסלולים המחוזרים על ידי W-A* תחת הפורמולציה $f = g + w \cdot h$ ב p_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 נכון, למשל ניקח את היוריסטיקה $\forall s \in S: h_{Zero}(s) = 0$ אשר תקיים שהיא קבילה כי בעצם מתקיים $\forall s \in S: 0 \leq h_{Zero}(s) = 0 \leq h^*(s)$ כי כל המעברים משקליהם חיוביים ממש וחסומים מלמטה על ידי $\delta = 1$. אך מתקיים ש: $f = g + w \cdot h \overset{h=0}{\cong} g$ לכן האלגוריתם במקרה הזה מתנהג כמו UCS וכתוצאה מכך שני המסלולים p_1, p_2 הם באותו משקל מינימלי שאלגוריתם UCS מוצא בסתירה לטענה לכן הטענה שגויה.

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

   אנו יודעים שאם h עקבית אז לפי ההגדרה לכל קשת (v,u): $h(v) - h(u) \leq cost(v,u)$,
   ובעצם לכל $0 \leq a \leq 1$ מתקיים ש: $a * \left(h(v) - h(u)\right) = ah(v) - ah(u) \leq cost(v,u)$
   ולכן גם האלגוריתם עם w_1 וגם זה עם w_2 יתנהגו כמו A* שמחזיר פתרון אופטימלי ולכן בפרט:
   $cost(p_1) = cost(p_2)$
   - הדוגמא שמצאנו בסעיף קודם בפרט תקפה לסעיף זה גם.

שאלה 10 – IDA* (2 נק׳):

1. **רטוב**: ממשו את החלקים החסרים באלג׳ IDA* בקובץ ע״פ ההנחיות המופיעות שם. עליכם להשתמש ביוריסטיקה h_{MSAP}.
2. יבש (2 נק׳): ספקו יתרון וחסרון של IDA* ביחס לA*. באילו מקרים הייתם מעדיפים להשתמש בכל אחד מהם?

<u>יתרון של IDA:</u> IDA צורך הרבה פחות זכרון לעומת A* כי הוא לא מחזיק רשימת OPEN גדולה כמו A*.

<u>חסרון של IDA:</u> IDA צורך יותר זמן כך שבעצם בכל איטירציה שלו הוא מתעלם מכל המידע שמצא באיטרציה הקודמת חוץ מ f_limit ולכן בסך הכל צורך יותר זמן מ A* כדי למצוא פתרון טוב.

לסיכום, אם היה לנו משאבי זכרון רבים אך מוגבלים בזמן היינו מעדיפים להשתמש ב A*. לעומת זאת אם היו משאבי הזכרון שלנו מוגבלים ו יש לנו זמן רב לחישוב אז היינו מעדיפים להשתמש ב IDA.

שאלה 11 – A* epsilon (6 נק׳):

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

<u>יתרון של A*-epsilon:</u> הוא לפעמים ימצא פתרון יותר מהר כיוון שהוא מאפשר לבחור צומת מתוך מבחר גדול יותר של צמתים ולהסתמך יותר על יוריסטיקות.
<u>חסרון של A*-epsilon:</u> הוא עלול להחזיר פתרון תת-אופטימלי בפקטור של $(1+\varepsilon)$ מהפתרון של A*..

2. יבש (4 נק׳): תנו הצעה ליוריסטיקה כדי לבחור את הצומת הבאה לפיתוח מתוך FOCAL. תארו את היוריסטיקה והציגו השוואה בין השימוש ביוריסטיקה זו לעומת השימוש ב-$g(v)$, מבחינת מספר פיתוחים, מסלול שנבחר ועלות המסלול שנבחר.

נציע להשתמש ביורסטקה: $h_{FOCAL}=w_t*g(v)+(1-w_t)*h(v)$ כאשר $w_t=\frac{1}{t}$ כאשר t זה משתנה אשר שומר את הזמן שעבר מתחילת החיפוש שלנו (נניח שזה מספר טבעי שאחרי כל פיתוח של צומת גדל ב 1 ומאתחלים אותו ב-1) , וכך ככל שעובר הזמן של החיפוש והאלגוריתם עדיין לא מצא פתרון -> אנחנו נוטים להאמין ליורסטקה h יותר ויותר כדי להאיץ את האלגוריתם ולהתקרב מהר יותר לצומת היעד.
אם נשווה שימוש ביוריסטיקה זו לעומת שימוש בערך g(v) נקבל שבהתחלה שניהם מתנהגים כמו בחירה לפי g(v) ועם הזמן מתחילים להאמין יותר ויותר ל h ולכן בוחרים יותר לפי h.
מבחינת עלות הפתרונות שיניבו שתי שיטות אלו, הן יניבו פתרונות עם עלות דומה אשר מובטח שתהיה חסומה על ידי פקטור של $(1+\varepsilon)$ מהפתרון האופטימלי בשני המקרים, עבור ε קטן הפתרונות יהיו קרובים אם לא זהים זה לזה.
לגבי מספר הצמתים אשר יפותחו בשתי השיטות, אם משתמשים בערך g(v) אז ייתכן שנפתח יותר צמתים כיוון ששיטה זו נוטה לבחור צמתים שיותר קרובים לנקודת ההתחלה אשר "פותחו היטב" ולכן אנחנו עלולים לפתח עוד כמה צמתים מיותרים. ואם משתמשים ביוריסטיקה h_{FOCAL}אז נוטים להאמין יותר ויותר ליוריסטיקה h עם הזמן ולבחור המצב הבא להיות זה שמוערך להיות הכי קרוב ליעד ולכן לפתח קצת פחות צמתים. נציין שלמרות שייתכנו הבדלים במספר הצמתים שיפותחו, וא כנראה יהיה הבדל קטן.

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.

- קיבלנו את התוצאות הבאות אחרי הרצת האלגוריתמים במקביל:-

map	BFS-G cost	BFS-G expanded	DFS-G cost	DFS-G expanded	UCS cost	UCS expanded	Greedy cost	Greedy expanded	WA* (0.5) cost	WA* (0.5) expanded	WA* (0.7) cost	WA* (0.7) expanded	WA* (0.9) cost	WA* (0.9) expanded
map12x12	109	141	121	33	87	97	227	32	87	92	87	82	89	26
map15x15	121	223	181	47	106	167	188	53	106	167	106	150	129	42
map20x20	256	390	282	57	175	309	268	45	175	308	175	298	202	56

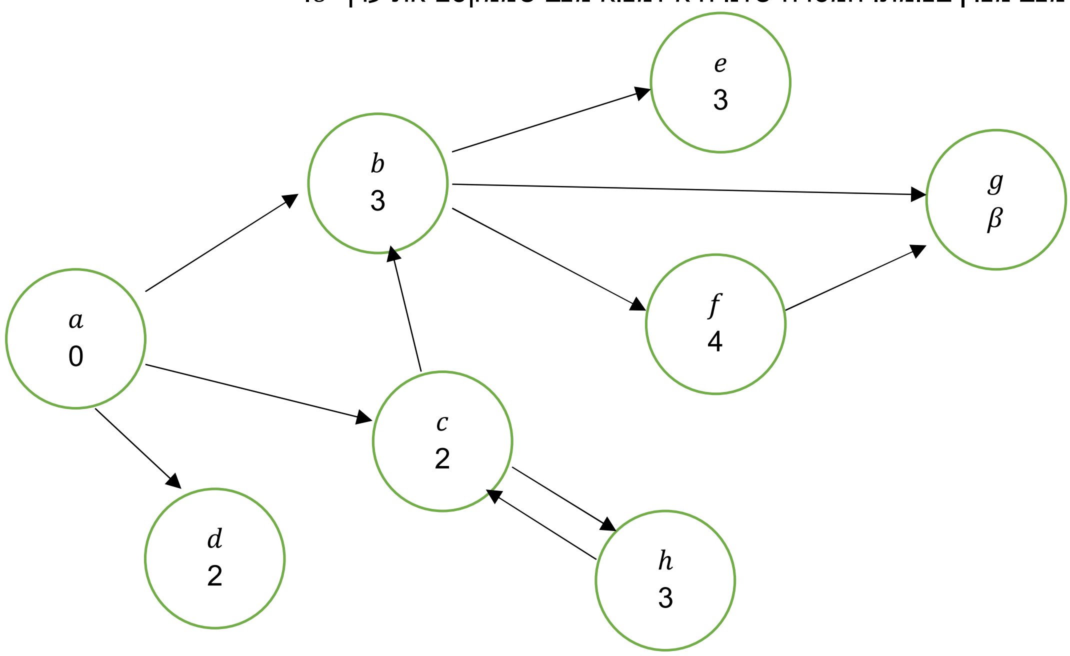
- ניתן לראות ש-BFS לא מחזיר פתרון אופטימלי מבחינת עלות כי אצלנו לא כל הקשתות עם משקל = 1 ויש גם חורים ויש portals , וכנ"ל לגבי ה-DFS כי הוא מחזיר את המסלול הראשון שדרכו מגלה את היעד שזה כמובן לא בהכרח האופטימלי.
- שלוש המפות ניתן לראות את ה-tradeoff בין הזמן שלוקח כל אלגוריתם לבין איכות הפתרון שהוא מציע: כך שככל שהאלגוריתם מסתמך יותר על יוריסטיקות ככל שהוא מגיע יותר מהר לפתרון (זאת אומרת, מפתח פחות צמתים עד שיגיע לצומת היעד) אבל מצד אחר מקבל איכות פתרון גרועה יותר) מחשב מסלול לא אופטימלי) ,
אחת הדוגמאות לזה היא ההבדל בין ריצת UCS לבין ריצת GREEDY על "map12x12":-
  - UCS החזיר לנו פתרון אופטימלי (ה- cost מינימלי ושווה ל- 87.0) אמנם לקח הרבה זמן עד שחישב אותו (כי פיתח 97 צמתים)
  - Greedy שמסתמך אך ורק על יוריסטיקות החזיר לנו פתרון רע מאוד (227.0) אמנם פיתח רק 32 צמתים ובגלל זה לא צרך הרבה זמן.

  וכאן נכנס לתמונה אגלוריתם ה- *weightedA שמנסה לאזן את ה-tradeoff ולמצות את הטוב מהשניים ואכן כפי שרואים בריצת *weightedA עם משקל 0.9 פיתחנו רק 26 צמתים ולרות זה קיבלנו פתרון מאווד קרוב לאופטימלי שזה 89.0 – כלומר התנהג יותר טוב מהשניים.

- יוריסטיקה יותר מיודעת כמובן תשפר את הביצועים של אלגוריתמי ה-Greedy וגם ה-*WeightedA כי היא יותר קרובה ליוריסטיקה המושלמת (שזה המרחק האמיתי מהיעד) וזה גם חוסך במספר הפיתוחים עד שנגיע לפתרון וגן מכוון אותנו לפתרון יותר טוב וקרוב לאופטימלי..

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 $\boldsymbol{\beta > 4}$.

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

מתקיים ש: $\Delta_{a\to d} = 2, \Delta_{a\to c} = 2, \Delta_{a\to b} = 3, \sum_{x\in Neighbors(a)} \Delta_{a\to x} = 7$ לכן:

$$p(d|a) = \frac{2}{7}, p(c|a) = \frac{2}{7}, p(b|a) = \frac{3}{7}$$

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

מספר הצעדים המקסימלי שאפשר לבצע הוא 4 שמתקבל עבור המעבר מ a ל c ל b ל f ל g.

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

נכון, כיוון שבוחרים את הצעד שהכי משפר בכל צעד, נבחר להתקדם ל g בהסתברות יותר גבוהה משאר השכנים (f,e) כאשר יש לצומת g את הערך של המקסימום הגלובלי כי כל שאר הצמתים יש להם ערך קטן/שווה ל 4 ולצומת g יש ערך $\beta > 4$. אך ייתכן גם שנעבור ל f ומשם מובטח שנתקדם אל g כי יש רק קשת לשם אשר משפרת בערך חיובי ממש. לעומת זאת אף פעם לא נתקדם אל e כי הוא לא משפר את הערך של b כלומר מתקיים $\Delta_{b\to e} = 0$.

וגם מתקיים: $\Delta_{b\to e}=3-3=0, \Delta_{b\to f}=\ 4-3=1, \Delta_{b\to g}=\beta-3, \sum_{x\in Neighbors(b)}\Delta_{b\to x}=\beta-2$

וכך קיבלנו שמתכנסים אל הפתרון $p(e|b)\ =\ \frac{3-3}{\beta-2}=0\ , p(f|b)\ =\ \frac{4-3}{\beta-2}=\frac{1}{\beta-2}\ , p(g|b)\ =\ \frac{\beta-3}{\beta-2}$ האופטימלי.

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

האופציות שלא מתכנסים בהם אל הפתרון האופטימלי הן:

1) נתקעים בצומת d.

האפציות לזה הן שמתקדמים מ a ל d בסיכוי $\frac{2}{7}$.

2) נתקעים בצומת h

האופציות לזה הן, להתקדם מ a ל c ואז ל h. שזה מתרחש בסיכוי : $p(c|a)\cdot p(h|c)=\frac{2}{7}\cdot\frac{1}{2}=\frac{1}{7}$

3) נתקעים בצומת e.

שהאופציות לזה הן, להתקדם מ a ל b ל e.

שזה מתרחש בסיכוי : $p(b|a)\cdot p(e|b)=\frac{3}{7}\cdot 0=0$

ולכן בסך הכל, הסיכוי להתכנס לפתרון <u>לא</u> אופטימלי הוא : $0+\frac{1}{7}+\frac{2}{7}=\frac{3}{7}$.

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

נגיע לצומת g שהוא המקסימום הגלובלי בתוך 3 צעדים בדיוק אם הולכים בשני מסלולים :

- מ a ל b ל f ל g : זה מתרחש בסיכוי $\frac{3}{7}\cdot\frac{1}{\beta-2}\cdot 1=\frac{3}{7(\beta-2)}$
- $a\to c\to b\to g$: זה מתרחש בסיכוי $\frac{2}{7}*\frac{1}{2}*\frac{\beta-3}{(\beta-2)}$
- ולכן מה שנרצה זה ש-

$$\frac{1}{5}<\frac{3}{7(\beta-2)}+\frac{2}{7}*\frac{1}{2}*\frac{\beta-3}{(\beta-2)}$$
$$\to\frac{7}{5}*(\beta-2)<3+\beta-3$$
$$\to\beta<7$$

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.