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: **29.2** יום חמישי, בשעה 23:59.(לשנות מועד ולפתוח פיאצה)
- 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: **שאדי דיאב** .
- 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 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 <u>מסמך זה והמחברת המצורפת</u>. 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:

1. **רטוב**: עברו על המחברת עד שאתם מגיעים לחלק של BFS-G ועצרו שם. בוצע
2. יבש (1 נק׳): תחילה נרצה להגדיר את מרחב החיפוש כפי שנלמד בתרגול. הגדר את (S, O, 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.

   $S = \{(cell\ number, 0/1, 0/1\ )\}$

   S מיוצג על ידי הווקטור (cell_number, bool, bool) כאשר cell_number זה מספר בטווח 63 – 0, וכל אחד מהערכים הבוליאנים יכול לקבל שני ערכים (0/1) בהתאם להאם אספנו את הכדור או לא.

   גודל מרחב המצבים הוא כגודל הobservation space שזה 64 * 2 * 2 = 256

O זה כיוון התנועה של הסוכן, והם {0 = DOWN, 1 = RIGHT, 2 = UP, 3 = LEFT}

I הוא המצב ההתחלתי, והוא (0, 0, 0)

G היא קבוצת מצבי המטרה ותוכנה הוא {(63, 1, 1)}

3. יבש (1 נק׳): מה תחזיר לנו הפונקציה Domain על אופרטור 2 (UP)?
הפונקציה תחזיר את כל המצבים שהם לא חור. מכל מצב אחר ניתן להפעיל את האופרטור (יכול להיות שנישאר במקום, אבל הפעלת האופרטור עדיין חוקית)

4. יבש (1 נק׳): מה תחזיר לנו הפונקציה Succ על המצב ההתחלתי 0?
הפונקציה תחזיר את כל המצבים שאפשר להגיע אליהם מהמצב ההתחלתי, והם כל המצבים המתאימים לתאים 0, במקרה שנשארנו במקום, 1, במקרה שזזנו ימינה, ו8 במקרה שזזנו מטה.
לדוגמא בלוח הספציפי הנתון בשאלה:

$$succ(0) = \{(8,0,0), (1,0,0), (0,0,0)\}$$

5. יבש (1 נק׳): האם קיימים מעגלים במרחב החיפוש שלנו?
כן, לדוגמא, אפשר להיתקע בלולאה אינסופית עם שני מצבים שהם לא חור, כאשר אחד מהם עובר לשני, והשני עובר לראשון.

6. יבש (1 נק׳): מה הוא מקדם הסיעוף בבעיה?
4 – מכל מצב במרחב החיפוש אפשר להגיע לכל היותר ארבעה מצבים

7. יבש (1 נק׳): במקרה הגרוע ביותר, כמה פעולות ידרשו לסוכן כללי להגיע למצב הסופי?
אינסוף פעולות. במקרה הגרוע ניתקע בלופ ונישאר בלולאה אינסופית

8. יבש (1 נק׳): במקרה הטוב ביותר, כמה פעולות ידרשו לסוכן כללי להגיע למצב הסופי?
כמו שלמדנו, מקדם הסיעוף בשאלה הוא סופי ולכן אלגוריתם BFS קביל תחת מחיר אחיד על הקשתות.
לכן אם אנחנו מתייחסים לכל צעד כקשת שמשקלה אחת – ריצת האלגוריתם שממומש בסעיף 2 תביא את הפתרון האופטימלי במספר הצעדים.
הצעדים הינם: $\text{Actions: } [0, 0, 0, 0, 0, 0, 1, 1, 1, 2, 1, 1, 0, 0, 1, 1]$
סהכ מדובר ב16 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.

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

קל לראות שהמסלול הקרוב ביותר במונחים של מרחק מנהטן הוא המסלול הירוק(1), ומשקלו 43. לעומת זאת, המסלול המסומן בוורוד(2), משקלו 9, והוא קל יותר מהאפרסקה.

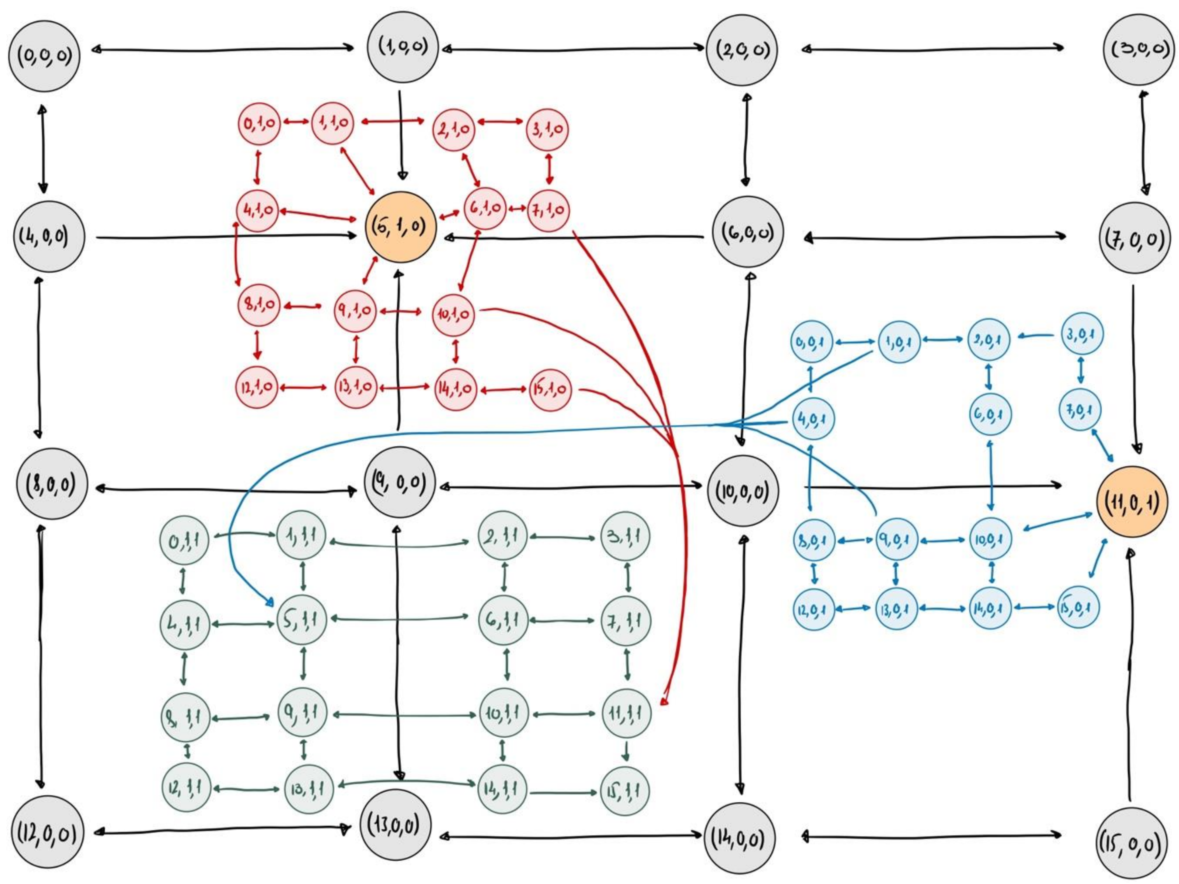
שאלה 2 – Breadth First Search-G (7 נק׳):

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. בוצע

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?
   על מנת שBFS וBFSG ייצרו ויפתחו צמתים זהים באותו הסדר, נדרוש שגרף החיפוש יהיה חסר מעגלים.
   ההבדל בין שני האלגוריתמים הוא תחזוק רשימת הclosed שמוודאת שלא נפתח את אותו מצב פעמיים, ובכך ממונעת מעגלים אינסופיים. לכן אם מלכתחילה בדרך החיפוש אין מעגלים, כלומר אין שני מסלולים שונים לאותה הצומת, האלגוריתמים יהיו זהים.

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



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

   BFS-G מבטיח לנו אופטימליות מבחינת מספר הקשתות, לכן נרצה לתרגם את העלות של כל תא, למספר קשתות שנדרשות על מנת להגיע אליו.
   הפונקציה T מבצעת את השלבים הבאים:
   בהינתן גרף מצבים G המייצג את הלוח,
   1. העתק את הגרף G, סמנו 'G
   2.כל צומת שעלותה 1 בG', או כל צומת שמייצגת חור בG', השאר כמו שהיא.
   3.לכל צומת אחרת, ניצור רכיב שהוא הצומת משורשרת לעצמה במסלול, כמספר פעמים כעלות הצומת. את הצומת הזו נכניס לG' במקום הצומת המקורית.

כלומר:

T(G) = G’

T(V,E) = V, E’

כאשר T פועלת על כל קשת $(u,v) \in E$ בצורה הבאה:

$T((u,v)) = (u_1, u_2, \ldots, u_i, v)$ s.t $i = cost(v)$

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?
יצירת צומת – קריאה לmake_node
פיתוח צומת – קריאה לexpand(node).
BFS עובר על הגרף לפי 'שכבות', כאשר הוא תחילה מבקר בצמתים שהכי קרובים לשורש, וכל איטרציה מתרחק לשכבה רחוקה יותר.
מכיוון שאין לנו חורים או כדורים, ונקודת ההתחלה שלנו היא בפינה הימנית למעלה, ונקודת הסיום היא בפינה השמאלית למטה, אלגוריתם הBFS יתקדם על הלוח באלכסונים, כאשר כל הצמתים הם צמתים שחוקי לדרוך עליהם, והמטרה היחידה היא להגיע לצומת המטרה, וכל צעד של האלגוריתם הוא למטה או ימינה.
בגלל שכל משבצת בלוח היא משבצת חוקית לדריכה, והאלגוריתם יכסה את כל משבצות הלוח – סהכ לכל משבצת על הלוח נקרא לפונקציה make_node – כלומר **ייווצרו N^2 צמתים**.
ישנם רק שני צמתים שעליהם לא נבצע expand – צומת המטרה, כי האלגוריתם יעצור באחד האבות שלו, ואחד מצמתי האב של המטרה, כי האלגוריתם יעצור באב הראשון, ויגלה שהילד שלו הוא פיתרון, ולכן לא ימשיך לאב השני.
לכן **סך כמות הצמתים המפותחים היא $N^2 - 2$**.

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?

DFS-G מחזיק רשימת close, ובכך מבטיח שלא ניכנס למעגלים אינסופיים, לכם אם קיים פיתרון האלגוריתם ימצא אותו ועל כן האלגוריתם שלם (בהנתן שN סופי).

האלגוריתם לא קביל, שכן לא מובטח שהפתרון של DFS-G יהיה מינימלי תחת פונקציית המחיר הנתונה בבעיה שלנו.

דוגמא נגדית: עבור הלוח הבא:

S	L	D
F	D	L
F	F	G

ריצת DFS-G תהיה כדלקמן:

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

עלות המסלול הנל הינה: $10+10+10+1+1+1+1+1+10+1+1+1=48$

בעוד שקיים מסלול: $(0,0,0) \rightarrow (1,0,0) \rightarrow (2,1,0) \rightarrow (5,1,0) \rightarrow (4,1,1) \rightarrow (5,1,1) \rightarrow (8,1,1)$

שמשקלו 6, ולכן כבר רואים שDFS-G לא החזיר מסלול אופטימלי.

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?

לא מובטח שהאלגוריתם ימצא פתרון. מכיוון שאלגוריתם לא מחזיק ברשימת close, הוא יכול לבקר באותו המצב פעמיים ולהיתקע בלולאה.

לדוגמא, באותו הלוח מהסעיף הקודם, האלגוריתם ירוץ במסלול ... $0 \rightarrow 3 \rightarrow 6 \rightarrow 7 \rightarrow 8 \rightarrow 5 \rightarrow 8 \rightarrow 5 \rightarrow 8$

(תחת ההנחה שניסיון לרדת למטה ממשבצת 6 לא יתקע אותו בלולאה סופית בתוך המשבצת עוד קודם)

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 <u>יפותחו</u> <u>וייווצרו</u> במהלך חיפוש **DFS-G**? הסבירו?

כעת אין לנו דרגונבולים לאסוף ואין חורים, ולכן המשימה היחידה שלנו היא להגיע מS לG.

האלגוריתם ינוע במסלול אנכי שיורד מטה מS עד לצומת התחתונה בפינה הימנית, ואז יפנה שמאלה עד שיגיע לG.

לכן הוא יפתח (expand) כל צומת במסלול הנל, מלבד הצומת הסופית(ראינו בתרגול שצומת המטרה לא מפותח)

- **יפותחו** $\boldsymbol{2N-2}$ $N+(N-1)-1=$ **צמתים שונים**.

לכל צומת במסלול, האלגוריתם יוצר את הצמתים השכנים לו,

לכן לכל צומת במסלול, הוא ייצור את הצומת שמתחתיה ומימינה אם הוא במסלול מטה,

או רק את הצומת שמשמאל אם הוא בצומת השמאלית התחתונה ביותר, או רק את הצמתים שמעל ומשמאל עבור המסלול שהולך ימינה, עד לצומת הסופית, לא כולל.

כלומר **ייווצרו** $\boldsymbol{4N-5}$ $=(N-1)*2+1+(N-2)*2$**צמתים שונים**.

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 <u>יפותחו</u> <u>וייווצרו</u> במהלך חיפוש DFS-G backtracking? הסבירו?

ההבדל הוא שכעת האלגוריתם יוצר צומת בצורה עצלה, כלומר הוא תיצור צומת רק כאשר היא תידרש 'לדרוך' עליו, ולא יוצר לכל צומת את כל בניו מההתחלה.

לכן מספר הצמתים **המפותחים** יהיה כמו בסעיף קודם - $\boldsymbol{2N-2}$,

אבל כעת לא ניצור את הצמתים השכנים המיותרים שיצרנו קודם, ומספר הצמתים הנוצרים יהיה כמספר הצמתים במסלול - $\boldsymbol{2N-1}$

שאלה 4 – ID-DFS (6 נק׳):

1.

a. (1 נק׳) האם האלגוריתם שלם? אם כן, הוכיחו. אם לא, ספקו דוגמה נגדית.
כן, מכיוון שפתרון קיים והוא בעומק סופי, אז האלגוריתם ימצא אותו. זאת מכיוון שהאלגוריתם מריץ את DFS-L כל פעם עם עומק שונה, מה שמונע ממנו להיתקע בלולאה אינסופית, ולחפש בכל הכיוונים ולא רק באחד.

b. (1 נק׳) נניח כי עלות כל פעולה היא 1, האם האלגוריתם קביל? אם כן, הוכיחו. אם לא, הסבירו**.**
כן, המסלול המוחזר הוא תמיד הקצר ביותר, וזאת מכיוון שבכל איטרציה של ID-DFS עם עומק כלשהו d, אנחנו בודקים את כל המסלולים האפשריים בעומק d, ולכן אם קיים פתרון, אנחנו נמצא אותו. ורק אחרי סריקת כל המסלולים בעומק הזה ממשיכים להבא. ואם קיים פתרון בעומק קטן dm < d’, אז כשהיינו מריצים את האלגוריתם על העומק הזה, היינו מוצאים.

2. הניחו כי יש לנו ידע מקדים על חסם עליון למרחק למצב מטרה, נסמנו 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.
כאשר החסם העליון D הדוק, כלומר קרוב מאוד לעומק האמיתי d, והפתרון נמצא בעומק גדול מאוד, אז עדיף לנו להריץ את ReverseDFS, כי הוא ייבצע כמה איטרציות וימצא את הפיתרון. לעומת זאת, ID-DFS יריץ הרבה איטרציות עד שימצא אותו.
במקרה ההפוך, כאשר הפתרון נמצא בעומק לא גדול מאוד, וD הוא לא חסם הדוק, עדיף יהיה להריץ את ID-DFS, כי ReverseDFS יתחיל להריץ עם D גדול, וייתכן שייקח לו הרבה זמן להריץ כל פעם עם עומק גדול.

b. (2 נק׳) הציעו כיצד ניצן לייעל את האלגוריתם. רמז: האם אתם יכולים לחשוב על צעד עדכון עדיף לL?
נשנה את האלגוריתם בצורה הבאה :
ReverseDFS מריץ DFS-L בכל פעם עם L שקטן ב1, כלומר עבור קלט D האלגוריתם יריץ עד שניכשל.
נניח שמצאנו פיתרון בעומק S, במקום להריץ שוב עם L-1, נריץ את האלגוריתם עם S-1, כי ידוע לנו שכל עומק בין [S, L-1] ימצא פיתרון, לכן מיותר להריץ עם עומק זה.

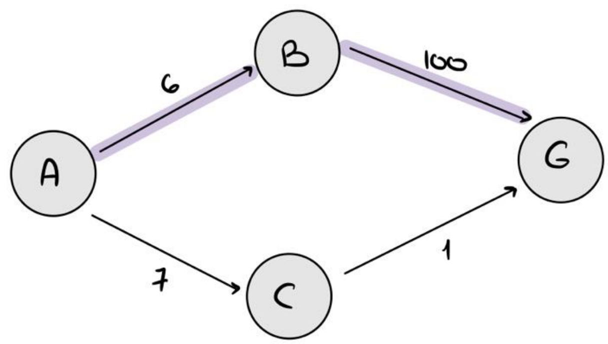
שאלה 6 - UCS (4 נק׳):

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.
אם המחיר על הקשתות אחיד. זאת מכיוון שההבדל הוא שבUCS תור הOPEN ממוין, ובBFS לא, לכן אם המחיר על הקשתות הוא אחיד, אז אין משמעות למיון.

2. יבש (1 נק׳): האם בבעיית החיפוש שלנו, עבור לוח NxN, האלגוריתם הוא שלם? האם הוא קביל?
פונקציית המחיר בבעיית החיפוש שלנו חסומה מלמטה על ידי 1 > 0, ולכן כמו שלמדנו, האלגוריתם שלם.
האלגוריתם קביל, כפי שראינו בהרצאה ובתרגול

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.
דוגמא לגרף שבו המסלול המוחזר לא יהיה אופטימלי הוא:



במקרה זה המסלול המוחזר הוא המסלול המסומן, $A \to B \to G$ ומשקלו 106, בפועל המסלול הקל ביותר הוא $A \to C \to G$
דוגמא לגרף שעבורו הפתרון של שאדי יחזיר את המסלול הזול ביותר הוא שרוך לדוגמא, וזאת מכיוון שיש רק מסלול יחיד לG

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.

יהיה חסם תחתון $0 < \delta$ על אורך הכבישים

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 : $h_{MD}(p) = |P - G|_1 = |G_x - P_x| + |G_y - P_y|$

היוריסטיקה קבילה עם $\epsilon = \sqrt{2}$. ממשפט פיתגורס, $a^2 + b^2 = c^2$, החסם העליון עבור c מתקבל עבור משולש ישר זווית שווה שוקיים. לכן מתקיים ש $c = \sqrt{2} * a$. במקרה הכי גרוע, נקבל שהיוריסטיקה נותנת לנו סכום של שני מסלולים שווים שמקבילים לצירים, והמסלול האמיתי הוא מסלול ישר, לכן יווצר לנו משולש ישר זווית שווה שוקיים. במקרה הזה מתקיים

$$h_{md}(s) = 2a = \sqrt{2}\,\sqrt{2}\,a = \sqrt{2}\,c = \sqrt{2}\,h^*(s)$$

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

היוריסטיקה היא אפסילון קבילה.

במקרה שייתכנו מסלולים שהם רק מקבילים לצירים, אז היוריסטיקה מחזירה ערך שהוא לכל היותר כגודל המסלול האמיתי, אם למשל המסלול הוא קו ישר מאונך לציר. אחרת, תחזיר תחת מסלול ישר קטן יותר מהמסלול האמיתי.

במקרה שייתכנו מסלולים שאינם מקבילים, אז נקבל מעין משולש ישר זווית, כאשר היתר הוא המסלול האמיתי. ולכן אורך הצלעות קטן יותר מאורך היתר.

היוריסטיקה קבילה עם אפסילון שווה 1

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

לפי משפט הנורמות, נורמה L2 תמיד גדולה שווה לנורמה L3, *h היא נורמת L2, לכן תמיד מובטח לנו ש

$$h(p) = L3(p) \leq L2(p) = h^*(p)$$

לכן קבילה עם אפסילון שווה 1

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 .

נתון: $h_1(p) \leq \epsilon_1 \times h^*(p)$ וגם: $h_2(p) \leq \epsilon_2 \times h^*(p)$ לכן:

$$h_3(p) = h_1(p) + h_2(p) \leq \epsilon_1 * h^*(p) + \epsilon_2 * h^*(p) = (\epsilon_1 + \epsilon_2) * h^*(p) = \epsilon_3 * h^*(p)$$

לכן החסם $\epsilon_3 = \epsilon_1 + \epsilon_2$

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, *h מחזיר את האורך של המסלול האופטימלי שעובר דרך שני הכדורים ומגיע ליעד, לעומת זאת, Hmsap מחזיר אורך של מסלול כלשהו, לנקודת יעד כלשהי, כאשר *H סוכם את אורכי המסלולים לכל נקודות היעד, לכן $h_{MSAP}(p) \leq h^*(p)$

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, עקבית. נסמן:

$$h(s) = d$$
$$h(s') \leq d + 1$$
כי, או שיש צומת מטרה שמרחקה מ’s קטן יותר מd+1, או שאין, ואז צומת המטרה הקרובה ביותר ל’s היא אותה צומת מטרה של s, ולכן מרחקו מ’s הוא d+1

בנוסף, s ’s הם צמתים עוקבים, והמחיר המינימלי של הפעולה הוא 1

לכן, $h(s') - h(s) \leq d + 1 - d = 1 \leq cost(s, s')$

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.

לא, דוגמא נגדית.

דוגמא נגדית:

נניח והלכנו במסלול המצורף, אז אנחנו כרגע במצב $(6, T, T)$,

אספנו את שני הכדורים, ולכן עבור המצב $(6, T, T)$, המרחק האמיתי אל צומת המטרה הוא ירידה מטה אל צומת מספר 10, שכן כבר אספנו את שני הכדורים, ולכן $h^*(6, T, T) = 1$, בעוד ש$h_{new}(6, T, T) = 4$.

(מרחק מנהטן עבור הצומת G במשבצת 12)

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

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

לא, כפי שלמדנו, עקביות גוררת קבילות, ולכן בגלל שהיוריסטיקה לא קבילה, היא גם לא עקבית.

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, וזאת מכיוון שמרחב המצבים שלנו סופי וקשיר, והיוריסטיקה רק קובעת סדר.
ראינו בתרגול שהאלגוריתם אינו קביל.
2. יבש (2 נק׳): תנו יתרון וחיסרון של אלגוריתם Greedy Best first Search לעומת Beam Search.
יתרון של Greedy BFS – טיב הפתרון לא נפגע כי כל המצבים יפותחו.
חיסרון: צורך יותר זיכרון וזמן ריצה מחיפוש אלומה

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 $f = g + w \cdot h$ 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 נכון, דוגמא נגדית:
   נתבונן ביוריסיטיקת האפס – כפי שלמדנו היוריסטיקה קבילה, אבל מתקיים כי:
   יהיו w_1, w_2 כנדרש, מתקיים:
   $$f_1 = g + w_1 * h = g + w_1 * 0 = g + 0 = \ g + w_2 * 0 = g + w_2 * h = f_2$$
   כלומר בפועל אנחנו מריצים את wA^* עם אותה הפונקציה, ולכן נקבל $cost(p_1) = cost(p_2)$ ולכן אותו מחיר
   b. יוריסטיקה כללית (לא בהכרח קבילה) h. אם כן הסבירו. אם לא, ספקו דוגמה נגדית.
   הדוגמא מסעיף קודם היא דוגמא 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: IDA* צורך פחות זיכרון מA*, בIDA* צריכת הזיכרון היא ליניארית באורך המסלול, בעוד בA*, צריכת הזיכרון היא פרופורציונלית במספר הצמתים שנוצרו
חסרון: בIDA* מפתחים מצבים שכבר פותחו, מבלי לדעת שביקרנו בהם בעבר, בעוד בA* אנחנו נמנעים מפיתוחים חוזרים של מצבים כשאין שיפור

2. יבש (1 נק׳): ספק המחשה שלב אחר שלב של אלגוריתם IDA* על הלוח (4x4) שמופיע במחברת, המראה כיצד החיפוש מתקדם באמצעות העמקה איטרטיבית ?
(השאלה מוגדרת על פונקצית היוריסטיקה $h_{MSAP}(s)$)
אלגוריתם IDA* מריץ איטרציות של DFS על עץ, כלומר אין לו זיכרון של המצבים שהוא פיתח בעבר, כאשר כל ריצה היא ריצת DFS, שמתחשבת בסדר הצעדים המוגדר בשאלה. הגרף שעליו אנחנו מריצים את האלגוריתם הוא הדרך שסופק בשאלה 2 סעיף 3.
כפי שלמדנו, האלגוריתם שלם וקביל, כי פונקציית המחיר שלנו חסומה מלמטה ע"י 1 > 0, וכפי שהוכחנו בשאלה 7, היוריסטיקה קבילה.
לכן האלגוריתם יחזיר לנו את המסלול האופטימלי (מבחינת מחיר) בגרף, תוך כדי שהוא נותן קדימות לקשתות לפי סדר ריצת הDFS בכל איטרציה. (כלומר למטה, ימינה, מעלה, שמאלה)
לכן המסלול שמוחזר בהרצת האלגוריתם על דרך המצבים הינו הינו:
$$(0,0,0) \to (4,0,0) \to (5,1,0) \to (9,1,0) \to (10,1,0) \to (11,1,1) \to (15,1,1)$$
כלומר המסלול הוא $0 \to 4 \to 5 \to 9 \to 10 \to 11 \to 15$

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}.
בוצע

2. יבש (2 נק׳): תנו יתרון וחיסרון של A*-epsilon לעומת A*.
יתרון של A*-epsilon: מאפשר בחירה לא אופטימלית בתקווה למצוא פיתרון מהיר יותר
חסרון: האופטימליות לא מובטחת, טיב הפתרון נפגע.

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.
נבחר להשתמש ביוריסטיקה h_{MSAP}
היוריסטיקה יותר מיודעת מפונקציית המחיר, ולכן ההערכה היא שבשימוש יוריסטיקה קבילה זו, יפותחו פחות צמתים מאשר שימוש בפונקציית המחיר, זאת משום ששימוש ביוריסטיקה הופכת את החיפוש למיודע יותר וכך בסבירות גבוהה נפתח צמתים ״טובים יותר״, ולכן קטן הסיכוי שנפתח צמתים מיותרים.
ייתכן שנקריב את טיב הפתרון בשימוש היוריסטיקה, בגלל שהיא נותנת הערכה בניגוד לg שנותן מחיר אמיתי.

4. יבש (1נק׳): אם נגדיר שאפסילון שווה לאינסוף איך תהיה ההתנהגות של האלגוריתם עם סביבת כדורי הדרקון.
במקרה זה, הFOCAL, לכל צומת, הוא כל הצמתים בOPEN, לכן תמיד בחירת הצומת הבאה לפיתוח היא הצומת בOPEN בעלת g מינימלי. כלומר במקרה זה האלגוריתם יבחר צמתים רק לפי מחיר מינימלי, וזה זהה להתנהגות של אלגוריתם UCS

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).

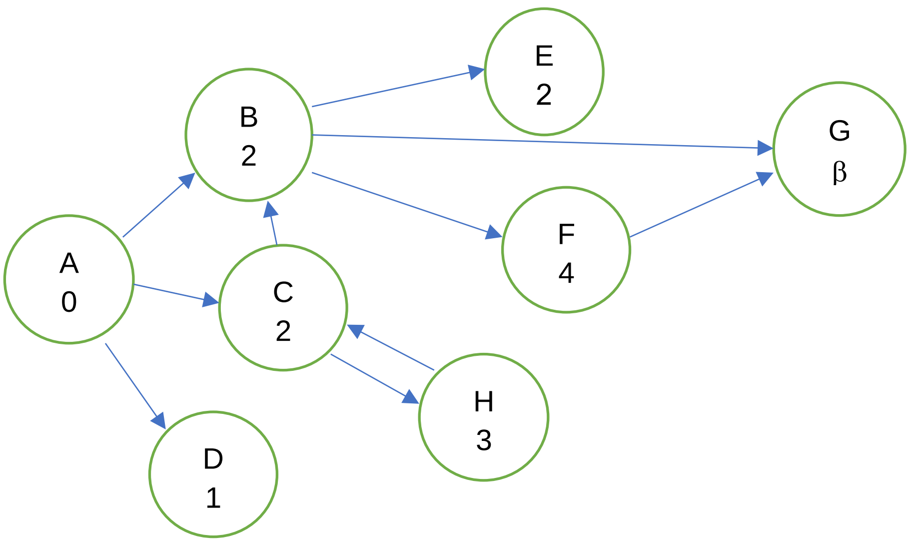
WA* (0.9) expanded	WA* (0.9) cost	WA* (0.7) expanded	WA* (0.7) cost	WA* (0.5) expanded	WA* (0.5) cost	BFS-G expanded	BFS-G cost	map
240	118	200	118	224	118	445	140	map12x12
707	195	604	178	651	178	858	215	map15x15
1002	188	587	188	684	188	1045	203	map20x20

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.
קצת תואם וגם קצת לא תואם.
BFS-G מבטיח להחזיר לנו פתרון אופטימלי כאשר המשקל על הקשתות אחיד. מכיוון שזה לא המקרה, קיבלנו פתרון שמשקלו לא אופטימלי, בהתאם למצופה.
כמות הצמתים המפותחים בBFS-G גדולה יותר מהכמות בWA* , וזה מתאים לציפיות שלנו, שכן השימוש ביוריסטיקה מטרתו "להאיץ" את החיפוש ולהגיע למטרה מהר יותר.

עבור WA*: כאשר המשקל=0.5 – האלגוריתם מתנהג כמו A* ולכן משקל המסלול המוחזר הינו המסלול האופטימלי – כמצופה.
ככל שאנחנו מגדילים את המשקל, נצפה לפגיעה בטיב הפתרון, ולהקטנה במספר הצמתים המפותחים.
כפי שאנחנו רואים בתוצאות שלנו, ככל שאנחנו מעלים את המשקל, עדיין אנחנו מקבלים את הפתרון האופטימלי מבחינת מחיר המסלול, אבל עם כמות צמתים גדולה יותר, מה שלא תואם להתנהגות המצופה מהאלגוריתמים, אבל זה לא מפחיד אותנו, כי בסוף התנהגות מצופה היא לא התנהגות מובטחת, ולכן יכול לצאת מצב שלוחות מסוימים פוגעים בביצועים, ולא מניבים תוצאה שמתאימה להתנהגות המצופה.
נוכל לצפות (וגם ניסינו וזה מה שיצא לנו) שבהינתן יוריסטיקה 'מוצלחת' יותר, לדוגמא יוריסטיקה שמחשבת את המרחק למצב המטרה רק לאחר שאספנו את שני הכדורים, כמות הצמתים המפותחים תהיה קטנה יותר.

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 מהמצב ההתחלתי לכל אחד מהמצבים b, c, d. רשמו את $p(d|a).p(b|a), p(c|a)$

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

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

נחלק למקרים לפי הערך של β

$\beta > 4$:

במקרה הזה, אנחנו נבצע 3 צעדים, $A \to B \to F \to G$, בגלל שהערך של U בF הוא 4, אז צריך שהערך U בG יהיה גדול מ4. (כשאין מצבים משפרים אנחנו נתקעים)

$\beta < 4$:

במקרה זה, מספר הצעדים המקסימלי יהיה 2, עד שנגיע לצומת שאים ממנה שיפור

$A \to B \to F$ או $A \to C \to H$

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

לא, האלגוריתם יתכנס למקסימום לוקלי. מכיוון שמ c אין לאן להתקדם חוץ מH, אז הערך המקסימלי יהיה 3, כמו הערך בH. לעומת זאת, יש לנו צומת עם ערך 4, לכן המקסימום הגלובלי הוא לכל הפחות 4

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

יש שני מקרים :

$\beta > 4$:

במקרה הזה, המקסימום הגלובלי הוא בצומת G. הסיכוי להגיע למקסימום זה הוא:

$$\frac{2}{5} * \frac{2}{\beta} * 1 + \frac{2}{5} * \frac{\beta - 2}{\beta}$$

לכן ההסתברות שנגיע לפתרון הלא אופטימלי הוא $1-(\frac{2}{5}*\frac{2}{\beta}*1+\frac{2}{5}*\frac{\beta-2}{\beta})$

$\beta < 4$:

במקרה הזה המקסימום הגלובלי הוא 4, בצומת F, ההסתברות להגיע למקסימום זה היא:

$$\frac{2}{5}*\frac{2}{\beta}$$

לכן ההסתברות שנגיע לפתרון הלא אופטימלי הוא $1-\frac{2}{5}*\frac{2}{\beta}$

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

אף β.

אם $\beta \leq 4$, המקסימום הגלובאלי של הגרף נמצא בצומת *F* (האלגוריתם לא ימשיך אל צומת *G* גם אם הערך שלה הוא 4, כי זה לא שיפור לעומת *F*), אבל האלגוריתם יכול להגיע אליו רק תוך 2 צעדים בדיוק, ולכן עבור מקרה זה ההסתברות להגיע אל *F* תוך 3 צעדים היא 0, ואינה גדולה מ$\frac{1}{5}$.

אחרת, $\beta > 4$, והמקסימום הגלובאלי נמצא בצומת *G*.

על מנת להגיע אל צומת *G* ב3 צעדים בדיוק, האלגוריתם יבחר במסלול $A \to B \to G$ ולכן נקבל:

$$\frac{2}{5}*\frac{2}{\beta}*1 > \frac{1}{5}$$

$$\Rightarrow \frac{4}{\beta} > 1 \Rightarrow 4 > \beta$$

בסתירה לכך ש$4 < \beta$.

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.