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

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

סיפור מסגרת

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

שאלה 1 – מבוא (8 נק׳):

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

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

$$S = \{(s, d1, d2) |\ s \in [0,63], d1, d2 \in \{True, False\}\}$$
$$O = \{Down, Up, Right, Left\}$$
$$I = \{(0, False, False)\}$$
$$G = \{(63, True, True)\}$$

גודל מרחב המצבים S יהיה $2*2*rowsNum*colsNum$. מספר השורות והעמודות יהיה בדומה לנלמד בכיתה, נכפול גם ב2*2 על מנת לייצג את האפשרויות של כל מצב עבור d1,d2.

3. יבש (1 נק׳): מה תחזיר לנו הפונקציה Domain על אופרטור 2 (UP)?
בהנחה ומתייחסים אל החורים כמצב שאפשר להפעיל עליו אופרטור (כלומר אנחנו כוללים אותם בחישוב שלנו), המצבים שעליהם ניתן להפעיל את האופרטור UP הינם כל קבוצה S. נשים לב כי במקרה של חור,לא יקרה דבר (כלומר הסוכן ישאר בחור), ובמקרה של השורה העליונה במפה, נשאר באותה משבצת גם לאחר הפעלת האופרטור.
במידה ולא ניתן להפעיל אף אופרטור על הסוכן לאחר שהגיע לחור, אזי $Domain(Up)=S/Holes$

4. יבש (1 נק׳): מה תחזיר לנו הפונקציה Succ על המצב ההתחלתי 0?
נשים לב כי אם נפעיל על המצב ההתחלתי את הפעולות $Up, Left$, נשאר באותה המשבצת אבל נשלם על ה"מעבר" למצב הבא, שהוא בעצם השארות באותה המשבצת. התשלום יהיה 1 עבור כל "מעבר" כזה, כמחיר של S הנתון בשאלה.
בשאר המקרים (Down,Right), נשלם את המחיר של המשבצת אליה נתקדם, במקרה זה המחיר יהיה 10 עבור משבצת F.
$$Succ(0, False, False) = \{((8, False, False), 10, False), ((1, False, False), 10, False), ((0, False, False), 1, False), ((0, False, False), 1, False)\}$$

5. יבש (1 נק׳): האם קיימים מעגלים במרחב החיפוש שלנו?
כן, הסוכן יבחר להתקדם למצב הבא, ללא תלות במצבים הקודמים שהוא עבר (כל עוד הוא לא בחור), אלא כתלות במצב הנוכחי בו הוא נמצא. למשל, עלול לקרות מצב שבו הסוכן יעשה באופן איטרטיבי את סדר הפעולות Down,Right,Up,Left שוב ושוב, כך שיווצר מעגל.

6. יבש (1 נק׳): מה הוא מקדם הסיעוף בבעיה?
4, מפני שקיימים מצבים במפה שבהם הסוכן יכול לבחור להתקדם ל1 מבין 4 מצבים אפשריים (מעלה,מטה,ימינה,שמאלה). זאת כמובן בהנחה שהסוכן לא נמצא במצב נוכחי של חור.

7. יבש (1 נק׳): במקרה הגרוע ביותר, כמה פעולות ידרשו לסוכן כללי להגיע למצב הסופי?
אינסוף, כי עלולים להיות מעגלים כמתואר בסעיף 5.

8. יבש (1 נק׳): במקרה הטוב ביותר, כמה פעולות ידרשו לסוכן כללי להגיע למצב הסופי?
סהכ 16 פעולות. נרצה לעבור דרך 2 הכדורים כדי להגיע למצב הסופי, לכן סהכ נאלץ לבצע 16 פעולות.

9. יבש (1 נק׳): עבור לוח כללי, המסלול הקל ביותר הוא המסלול שמגיע למצב מטרה שהכי קרוב למצב ההתחלתי (במונחים של manhattan distance)? אם כן, הוכיחו. אם לא, ספקו דוגמא נגדית.
לא.נסתכל על המפה הנתונה ההתחלתית. נניח כי קיים מצב מטרה נוסף לקיים, במשבצת מספר 1 (כלומר זו שמימין לS), נסמנה G1. מרחק מנהטן למטרה המקורית (נסמנה G2), הוא רחוק הרבה יותר מהמרחק לG1. בפועל בהרצת האלגוריתם של הסוכן, הוא חייב לעבור דרך d1,d2 על מנת לסיים את הריצה, ולכן אם כוללים את המעבר דרכם, המסלול הקל ביותר הוא דווקא המסלול ל-G2 ולא לG1.

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

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

1. **רטוב**: ממשו את אלג׳ BFS-G (על גרף) במחברת ע״פ ההנחיות המופיעות שם.

2. יבש (1 נק׳): מה צריך להיות התנאי על גרף החיפוש (לא בהכרח בבעיית כדורי הדרקון) כך שBFS על גרף ו-BFS על עץ ייצרו ויפתחו צמתים זהים באותו הסדר?

התנאי החשוב ביותר – שלא יהיו מעגלים בגרף. כל עוד יש מעגלים, יהיו הבדלים בין עץ לגרף (זאת מכיוון שכנלמד, בעץ אין מעגלים).
תנאים נוספים:
   1. ששובר השיוויון יהיה זהה, כלומר הבחירה של הצומת הבאה שאותה נרצה לפתח, תעשה באופן זהה.
   2. להתחיל מאותה הצומת בגרף ובעץ.

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

False, False
True, False
False, True
True, True

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

על מנת להשתמש בBFS לקבלת פתרון אופטימלי, בהתחשב בעלות של כל צומת, נרצה להגיע למצב שבו עלות כל צומת וכל קשת הוא זהה, במקרה שלנו נבחר ב1. לשם כך, כל צומת שעלותה גדולה מ-1, נסמן את העלות שלה c, נפצל ל-c צמתים(כולל המקורית) כך שנוצרת שורה של צמתים שמחוברים זה לזה באופן חד כיווני, כך שעלות כל צומת וכל קשת, יהיה 1. פיצול זה יאפשר הרצה של BFS כנדרש.

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

יווצרו N^2 צמתים, ויפותחו $N^2 - 2$ צמתים.
הסיבה לכך היא שברגע שנגיע ל"שכבה" האחרונה של צמתים, שעבור 2 הצמתים שנמצאים בשכבה זו, היעד הוא שכן שלהם. ניצור את שני הצמתים הללו, ואז נפתח אחד מהם, נגיע לG ונעצור (נזהה שזוהי צומת המטרה). לא נפתח את G, ובנוסף לא נפתח את הצומת השניה שנמצאת באותה השכבה. לכן סהכ לא פתחנו 2 צמתים מתוך סך הצמתים שיצרנו.
כפי שלמדנו, BFS עובדת בשכבות ולכן האלגוריתם יפעל כך.

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

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

האלגוריתם שלם מפני שבG-DFS אנו שומרים רשימות OPEN,CLOSE ובכך דואגים שלא לפתח כל צומת יותר מפעם אחת, לכן במקרה של הלוח שלנו, בהנחה של N סופי, אם קיים פתרון, האלגוריתם יחזיר אותו (ובפרט לא יכול לקרות מצב שבו נקלע למעגל או שרוך).

האלגוריתם אינו קביל, כי הוא מאפשר קבלת פתרון שאינו אופטימלי (מתייחס למסלול הקצר ביותר ולא לעלות הנמוכה ביותר בהתחשב בעלויות הצמתים).

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 בהכרח. DFS על עץ לא מתחשב ברשימות הOPEN & CLOSE, וכתוצאה מכך, לא בודק האם כבר פיתח צומת שהוא נמצא בה. לכן עלול להווצר מצב שבו הסוכן מבצע את פעולת down עד שיגיע למשבצת התחתונה ביותר מצד שמאל, החל מרגע זה יבצע באופן רקורסיבי אינסופי את פעולה זו, למרות ש"אין לו עוד לאן לרדת", וכבר פיתחנו את צומת זו. למעשה, בהסתכלות על עץ מדובר בפיתוח של שרוך אינסופי. לכן לא בהכרח ימצא פתרון.

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

סהכ יפותחו $2N - 2$ צמתים, שכן הסוכן יתעדף את פעולת down ולכן ירד עד לשורה התחתונה ביותר (N צעדים). לאחר מכן יתעדף את פעולת right וילך ימינה עד שיגיע לצומת המטרה, סהכ N-2 צעדים. במהלך הריצה, כל צומת שנפתח, תיצור את כלל צמתי ההמשך, במקרה של הגרף הנתון מדובר על 2 צמתים בכל פעם (חוץ מצומת אחת). סהכ נקבל את 2 השורות התחתונות ביותר ו2 העמודות השמאליות ביותר. הצומת היחידה שלא תיווצר הינה הצומת שבדיוק מעל צומת המטרה.

$$(N-1) * 4 - 1 = 4N - 5$$

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

יפותחו $2N - 2$ צמתים, כפי שתיארנו בסעיף 3. כעת לא יווצרו הצמתים הנוספים אותם תיארנו, ניצור רק את הצמתים הרלוונטים למסלול (פיתוח עצל כנלמד בתרגול). לכן הפעם, נייצר $2N - 1$ צמתים (כמספר הצמתים הכולל במסלול).

שאלה 4 – ID-DFS (6 נק׳):

.1

a. (1 נק׳) האם האלגוריתם שלם? אם כן, הוכיחו. אם לא, ספקו דוגמה נגדית.
כן. נניח בשלילה שקיים פתרון נמוך ביותר בעומק d אך האלגוריתם לא יחזיר אותו.
למדנו כי DFS-L שלם כל עוד הפתרון נמצא בשכבות שמתחת ל-L הסופי שנבחר.
כנלמד, ID-DFS משתמש בDFS-L במימוש שלו.
כאשר נגיע לאיטרציה בה L=d, מהנחת השלמות של DFS-L נקבל פתרון עבור הבעיה, בסתירה להנחת השלילה.

b. (1 נק׳) נניח כי עלות כל פעולה היא 1, האם האלגוריתם קביל? אם כן, הוכיחו. אם לא, הסבירו.
כן. המחיר הינו אחיד על הקשתות, לכן DFS יחזיר מסלול קצר ביותר, כלומר יחזיר מסלול בעל עומק קטן ביותר, שכן הפתרונות שנמצאים בשכבה ה-d, נבדקים לפני הפתרונות שנמצאים בשכבה d+1 (לכל d) ולכן מדובר בפתרון אופטימלי, כלומר קביל.

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.
ID-DFSעדיף על Reverse : למשל כאשר יש עץ וצומת המטרה היא בן של השורש (שממנו מתחילים את הריצה). במקרה זה נוכל לרוץ רק פעמיים על האלגוריתם ונקבל את הפתרון. אם נשתמש בreverse, נריץ אותו מספר פעמים כעומק העץ (פחות 1).
Reverse עדיף על ID-DFS: למשל כאשר יש שרוך וצומת המטרה היא הצומת בעומק הגבוה ביותר (כלומר הצומת הרחוקה ביותר מצומת ההתחלה). במקרה זה Reverse ירוץ פעם אחת בלבד ואילו ID ירוץ מספר פעמים כעומק השרוך.

b. (2 נק׳) הציעו כיצד ניתן לייעל את האלגוריתם. רמז: האם אתם יכולים לחשוב על צעד עדכון עדיף לL?
נרצה לעדכן את L בכל איטרציה, להיות אורך הפתרון הקצר ביותר שאלגוריתם DFS שהורץ החזיר פחות 1. נשים לב כי DFS מחזיר את המסלול הקצר ביותר, גם במקרה שבו D או L הם גדולים מאורך זה. לכן אם נעדכן בכל פעם את L להיות בגודל של אורך המסלול הקצר ביותר, נוכל לוותר על איטרציות שהן "לא רלוונטיות", כלומר איטרציות בהן לא ימצא פתרון או ימצא פתרון שאינו אופטימלי (ובהמשך הריצה יתוקן).

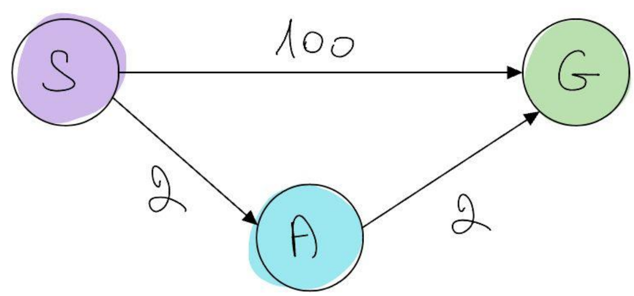
שאלה 6 - UCS (4 נק׳):

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

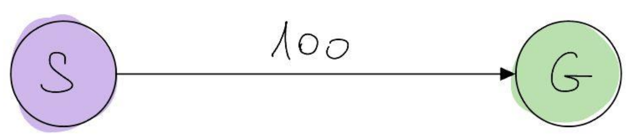
1. יבש (1 נק׳): עבור אילו בעיות חיפוש אלגוריתם UCS ואלגוריתם BFS יפעלו באותו האופן? הסבירו.
כאשר הקשתות בגרף הן במחיר אחיד. נשים לב כי ברשימת הOPEN בBFS, האלגוריתם מוציא איברים מהרשימה לפי סדר FIFO, ואילו UCS ממיין את הרשימה לפי גובה המחיר של כל קשת, ויוציא אותם מהרשימה לפי מיון זה. ברגע שמחיר כל הקשתות הינו זהה, UCS יוציא איברים מהרשימה בצורה זהה לBFS.

2. יבש (1 נק׳): האם בבעיית החיפוש שלנו, עבור לוח NxN, האלגוריתם הוא שלם? האם הוא קביל?
האלגוריתם שלם. נשים לב כי פונקציית המחיר של הקשתות (שמחושבת לפי צומת היעד של התנועה של הסוכן) חסומה מלמטה, כך שהחסם התחתון הוא 1. UCS מפתח את כל הצמתים אליהם ניתן להגיע (כל הצמתים שברכיב הקשירות), ועל כן אם קיים פתרון, הוא ימצא אותו.
האלגוריתם קביל, שכן הוא מחזיר מסלול שהמחיר שלו הוא הקטן ביותר, כלומר מסלול קל ביותר (בדומה לנלמד באלגו1).

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.
UCS השגוי יטעה במקרה שבו הוא יוצר את צומת המטרה, ובלי להתחשב במשקל הקשת שלה לעומת מסלולים אחרים, הוא יחזיר את המסלול אליה, בלי לפתח את שאר הצמתים.



UCS השגוי יתן תשובה נכונה כאשר הקשת שאיתה הוא מגלה את צומת המטרה, היא חלק מהמסלול הקל ביותר לצומת זו. בדוגמא הספציפית הזו, מתקיים גם תנאי זה ובנוסף קיים גם מסלול יחיד לצומת המטרה, ולכן גם במקרה זה (שהוא מקרה פרטני למה שציינו), האלגוריתם השגוי ימצא מסלול נכון.



שאלה 7 - יוריסטיקות (8 נק׳):

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

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

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

עבור 4 הסעיפים הראשון, נציין כי במקרה של השאלה שלנו, המרחק הקצר ביותר בין נקודת מקור כלשהי לצומת המטרה, הוא מרחק אוקלידי, על בסיס הנוסחא $\sqrt{(a^2+b^2)}$. נשים לב כי אם הכביש בין כל 2 נקודות הוא בעל פיתולים (כלומר לא ישר) הוא בוודאי ארוך יותר מאשר האופציה של כביש ישר, כלומר מרחק אוקלידי. לכן מרחק אוקלידי הוא החסם ההדוק ביותר עבורנו.

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

עבור אפסילון ששווה $\sqrt{2}$ נקבל יוריסטיקה אפסילון קבילה.
נסמן $a = |G_x - P_x|$, $b = |G_y - P_y|$,
כלומר $h_{MD} = a + b$
נמצא אפסילון עבורו מתקיים: $a + b \leq \varepsilon * \sqrt{a^2+b^2}$
כלומר $\varepsilon \geq \frac{a+b}{\sqrt{a^2+b^2}}$.
נרצה להסתכל על החסם ההדוק ביותר, כלומר מתי היחס בין מרחק אוקלידי למנהטן הוא הגדול ביותר. מצב זה מתקיים עבור $a = b$,
כלומר נקבל כי:

$$\varepsilon \leq \frac{2a}{\sqrt{2a^2}} = \frac{2a}{\sqrt{2}a} = \sqrt{2}$$

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

עבור אפסילון ששווה 1 נקבל יוריסטיקה אפסילון קבילה.
נסמן $a = G_x - P_x$ $b = G_y - P_y$,
נמצא אפסילון עבורו מתקיים: $\min\{a,b\} \leq \varepsilon * \sqrt{|a|^2+|b|^2}$
עפי משפט פיתגורס, מתקיים $a^2+b^2=c^2$, בנוסף מתקיים $a,b<c$ ובפרט מתקיים גם $\min\{a,b\}<c$
כלומר, עבור $\varepsilon = 1$, נקבל יוריסטיקה אפסילון קבילה.

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

עבור אפסילון ששווה 1 נקבל יוריסטיקה אפסילון קבילה.
נסמן $a = |G_x - P_x|$ $b = |G_y - P_y|$, ידוע כי תמיד נורמת L3 קטנה מנורמת L2, כלומר מתקיים:

$$\sqrt[3]{a^3+b^3} < \sqrt{a^2+b^2} \underbrace{=}_{for\ \varepsilon=1} \varepsilon * \sqrt{a^2+b^2}$$

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 .

האפסילון ההדוק ביותר הוא סכום האפסילונים.
נסמן את h_* כיוריסטיקה האופטימלית, אז מתקיים:

$$h_3 = h_1 + h_2$$

$$h_3 \le \varepsilon_1 * h_* + \varepsilon_2 * h_* = (\varepsilon_1 + \varepsilon_2) * h_*$$

כלומר האפסילון ההדוק ביותר הינו סכום האפסילונים. נניח בשלילהכי קיים חסם הדוק יותר, שאינו קטן מהמינימלי מבין אפסילון1 ואפסילון 2, אזי נקבל סתירה שכן הם מוגדרים כחסמים ההדוקים ביותר עבור היוריסטיקות הנתונות. (בדומה לכך שבאלגו ראינו שבמסלול קל ביותר, כל חלק של המסלול הוא גם כן הקל ביותר בין אותם 2 צמתים)

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.

אכן קבילה. מעצם הגדרה של יוריסטיקה קבילה, היא תיתן לכל היותר את המרחק המקורי בין המקור למטרה, או מרחק קצר יותר מהמציאות. לכן, במקרה של הלוח שלנו, שבו ניתן לנוע עם הצירים בלבד, מרחק מנהטן יתן את היוריסטיקה האופטימלית, לכן לפי הגדרה היא קבילה.

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

עקבית. נשים לב כי עלות המעבר בין שני צמתים תלויה במחיר של צומת היעד הנוכחית. המחירים הנתונים בתרגיל הם בין 1 ל10. מרחק מנהטן יחשב כל צעד בין 2 משבצות צמודות כ-1, לכן ההפרש בין כל 2 יוריסטיקות של צמתים עוקבים, יהיה קטן או שווה לעלות המרחק בפועל, כלומר עקבית לפי הגדרה.

נגדיר יוריסטיקה חדשה :

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

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

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

לא קבילה. נסביר באמצעות דוגמא נגדית: נניח שקבוצת המטרות הינה גדולה מ-1, והלוח נראה כך:

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

המסלול האופטימלי לצומת מטרה מצומת S (שכמובן חייב לעבור דרך 2 כדורי הדרקון), יהיה הבא:
S, פניה ימינה לD, פניה למטה לD, פניה שמאלה לG וסיימנו.
סהכ, עלות המסלול תהיה סה"כ 3, שכן עלות מעבר לצומת D או לצומג G הינה 1. על כן מתקיים $h_*(s) = 3$.
לעומת זאת, אם נשתמש ביוריסטיקה h_{new}, היא תחזיר את אורך המסלול הארוך ביותר מs לכל צמתי המטרה, כלומר היא תחזיר את אורך המסלול לצומת המטרה G שנמצא במשבצת התחתונה הימנית. אורך מסלול זה במרחק מנהטן הינו 5, כלומר יתקיים $h_{new}(s) = 5$.
נקבל כי היוריסטיקה הנתונה מקיימת $h_{new}(s) > h_*(s)$, לכן לפי הגדרה אינה קבילה.

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

לא עקבית, בגלל שאינה קבילה (לפי הגדרה ולפי הנלמד בשיעור).

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?
האלג' שלם, מפני שהוא יפתח את כל המצבים האפשריים, מרחב המצבים סופי וקשיר. לכן אם קיים פתרון, הוא יחזיר אותו.
האלג' לא קביל, מפני שהבחירה בצעד הבא תעשה בהסתכלות על מצב נוכחי בלבד ולא על המשך המסלול של כל אופציה, לכן עלול למצוא פתרון שאינו אופטימלי, כנלמד על אלגוריתמים חמדניים.

2. יבש (2 נק׳): תנו יתרון וחיסרון של אלגוריתם Greedy Best first Search לעומת Beam Search.
יתרון: בbeam search, כיוון שאנו שומרים רק K צמתים בOPEN, בהסתכלות רחבה על הפתרון יתכן כי "נזרוק" פתרון אופטימלי, שכן אנו "זורקים" אופציות שערכן היוריסטי הוא הגבוה ביותר בכל פעם. לכן אם הפתרון האופטימלי בגרף, יעבור בצומת שהיוריסטיקה שלה גבוהה (כלומר צומת שנזרוק במהלך הריצה), הפתרון שbeam יחזיר לא יהיה האופטימלי האפשרי. לעומת זאת, best first לא יזרוק אף פתרון.
חסרון: סיבוכיות מקום/זכרון – ב beamאנו מגבילים את כמות הצמתים שנשמור במחסנית OPEN ואילו בbest first אין הגבלה, לכן ככל הנראה נשמור צמתים רבים בכל רגע בריצת האלגוריתם.

שאלה 9 – W-A* (2 נק׳):

השאלות בחלק זה מתבססות על הלוח "8x8" שמופיע במחברת.

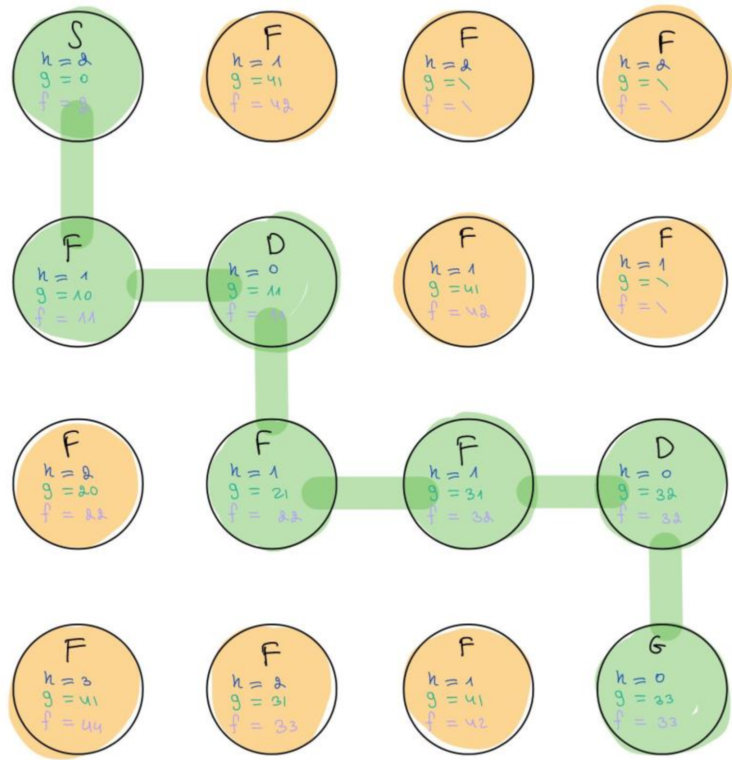
1. **רטוב**: ממשו את החלקים החסרים באלג׳ W-A* בקובץ ע״פ ההנחיות המופיעות שם. עליכם להשתמש ביוריסטיקה h_{MSAP}.

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.
   דוגמא נגדית: נשתמש ביוריסטיקת האפס, שהיא כנלמד קבילה. אז מתקיים:
   $$f_1 = g + w_1 * 0 = g = g + w_2 * 0 = f_2$$
   $$\Longrightarrow cost(p1) = cost(p2)$$
   (ולא קטן ממש)
   b. יוריסטיקה כללית (לא בהכרח קבילה) h. אם כן הסבירו. אם לא, ספקו דוגמה נגדית.
   דוגמא נגדית: אותה הדוגמא מהסעיף לעיל.

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: צריכת זכרון. בIDA צריכת הזכרון לינארי באורך המסלול. בA צריכת הזכרון פרופורציונית למספר הצמתים שנוצרו.
חסרון: בIDA יש פיתוח מצבים חוזרים (בלי לדעת שכבר ביקרנו בהם) ואילו בA לא נפתח מצב חוזר במצבים בהם אין 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 ?



שאלה 11 – A* epsilon (6 נק׳):

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

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

   יתרון – כתוצאה מהאפשרות לבחור צומת "כמעט הכי טובה", נוכל למצוא פתרון מהר יותר מאשר A רגיל.
   חסרון – הפתרון אינו אופטימלי בהכרח ועל כן האלגוריתם בוודאי אינו קביל.

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.
   יוריסטיקה: נבחר את הצומת שמרחק מנהטן שלה מצומת המטרה הוא הקטן ביותר. באופן זה, נוכל לבחור צומת "כמעט הכי טובה" שתקרב אותנו אל היעד. נציין כי בבחירת יוריסטיקה זו, "צמצמנו" את הבעיה כך שאנחנו לא מתחשבים באיסוף הכדורים או בחורים. לצורך העניין, יכולנו לבחור גם במרחק אוקלידי, אבל בדומה לדוגמא של פקמן מההרצאה, גם כאן הדמות (הסוכן) נעה רק בכיווני הצירים, ועל כן מרחק מנהטן יותר רלוונטי.

   השוואה בין היוריסטיקה שלנו לעומת שימוש בg: נשים לב כי היוריסטיקה שלנו מסתמכת על מרחקים עתידיים מהיעד, כלומר מרחק שעוד לא ביצענו, לעומת השימוש בg, שמתאר את המרחק שכבר עברנו מצומת ההתחלה עד כה. הסתמכות על g לא בהכרח יוכל לקדם אותנו אל היעד, שכן עלול לגרום לנו ללכת בכיוונים מנוגדים לכיוון היעד (למשל אם היעד בצד ימין למטה, אך הצומת הקרובה ביותר היא הצומת שמעל הסוכן, אז הסוכן בהסתמכות על g יתקדם מעלה במקום לכיוון היעד).
   לעומת זאת, חשוב לשים לב כי במקרה שבו במסלול יש מכשולים רבים (חורים למשל), נאלץ ככל הנראה לבצע "סיבובים" רבים עד ההגעה למטרה, ובמקרה זה בחירה בg עלולה להיות עדיפה.
   בממוצע, מספר הפיתוחים של 2 השיטות יהיה דומה.

4. יבש (1נק׳): אם נגדיר שאפסילון שווה לאינסוף איך תהיה ההתנהגות של האלגוריתם עם סביבת כדורי הדרקון.
במקרה זה, רשימת FOCAL תכיל את כל רשימת OPEN של הצמתים. כלומר אין פה בחירה של הצמתים "כמעט הכי טובים", אלא עכשיו אנחנו בוחרים מבין כלל הצמתים. הבחירה תעשה לפי היוריסטיקה הנבחרת של h_{focal}.

שאלה 12 – Benchmarking (2 נק׳):

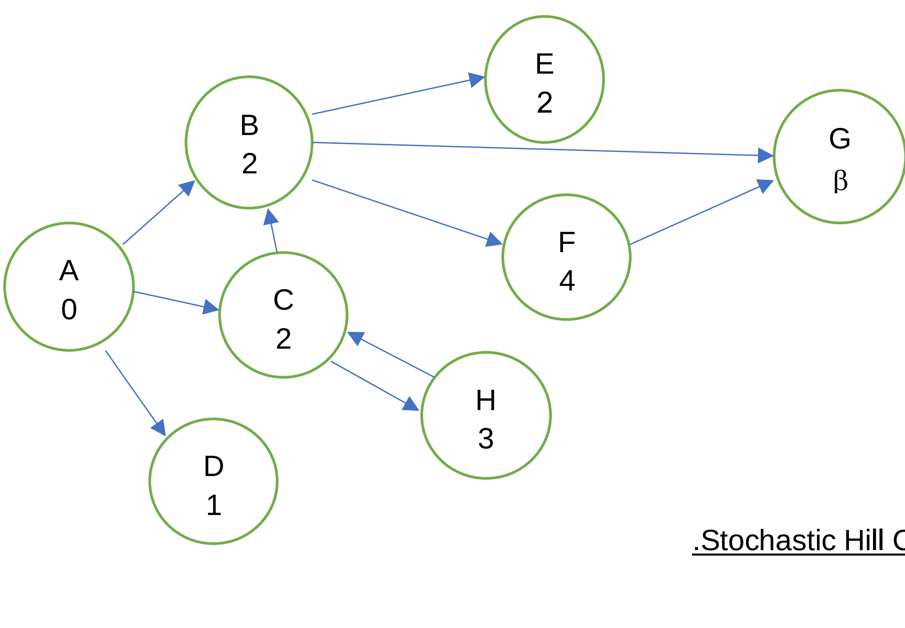
בשאלה זאת נשווה בין אלגוריתמי חיפוש שונים על בעיות שונות. הריצו את החלק הרלוונטי במחברת ותיראו שנוצר קובץ csv. (ניתן לפתוח עם Excel).

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.

התוצאות אכן תואמות את הציפיות שלנו.
למשל, נשים לב כי עבור *WA, ככל שהיוריסטיקה מיודעת יותר, כך אנו מקבלים תוצאה טובה יותר, אך עם פחות צמתים שפיתחנו, כלומר הביצועים שלנו (ברמת עלות מול תועלת) משתפרים כשהיוריסטיקה מיודעת יותר.
בנוסף, כמות הצמתים שאלגוריתם הBFS פיתח, גבוה משמעותית (פי 2) מכמות הצמתים שפיתח אלגוריתם ה*WA עבור כל ערכי המשקל הנתונים. גם העלות במקרה של BFS הינה גדולה יותר מהאלגוריתם האחר, לא מאוד משמעותי אך עדיין קיים הפרש לא מבוטל.
כמצופה, אלגוריתם *A יעיל יותר בכל המובנים מאשר BFS, שכן הינו משתמש ביוריסטיקה שמסייעת לפתרון, זאת כפי שציפינו וכמו שלמדנו.

שאלה 13 – Local Search (5 נק׳):

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



נשתמש באלגוריתם <u>Stochastic Hill Climbing</u>.

כמו כן ידוע כי $\boldsymbol{\beta > 3}$.

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

   מ-A ניתן לעבור:
   1. לB : מתקיים $\Delta U = 2 - 0 = 2$.
   2. ל-C: מתקיים $\Delta U = 2 - 0 = 2$.
   3. לD : מתקיים $\Delta U = 1 - 0 = 1$.

על כן מתקיים: $\Sigma\Delta U = 5$.

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

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

   לכל היותר 3 צעדים. בהנתן בטא גדולה מ-4, במידה והסוכן הגיע לצומת F, הוא יבחר לעבור לצומת G.

   מצומת A קיימות 3 אפשרויות:
   1. מעבר לצומת D, ואז סיימנו. כלומר צעד 1.
   2. מעבר לצומת C, לא יבחר לעבור לצומת B (כי אין שיפור), לכן יעבור לצומת H, ואז סיימנו (לא יחזור לC שכן אין שיפור) – סהכ 2 צעדים.
   3. מעבר לצומת B, לא יעבור לצומת E שכן אין שיפור. מכאן יוכל:
      a. לעבור לצומת F, ואז יעבור לG אם בטא גדולה ממש מ4 (3 צעדים). אחרת, ישאר בF (2 צעדים).
      b. יעבור ישירות לG (בטא גדולה מ-3 ולכן יהיה שיפור מB לG), ואז 2 צעדים.

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

   לא, כפי שהסברנו בסעיף למעלה, כאשר נגיע לצומת C נעבור לצומת H ו"נתקע" בה, האלגוריתם יסתיים עם הערך U=3. נבחין כי קיימת האפשרות לעבור לB, ומשם לF ואז פוטנציאלית לG, כלומר לכל הפחות ערך U יהיה שווה ל-4, ועל כן U=3 אינו מקסימום גלובלי.

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

נחלק למקרים לפי ערכי בטא אפשריים:

1. אם בטא בין 3 ל-4: אזי המקסימום הגלובלי נמצא בF והוא שווה ל-4.
   a. עלולים להתקע בD - $p(d|a) = \frac{1}{5}$
   b. עלולים להתקע בH – $p(c|a) * p(h|c) = \frac{2}{5} * 1 = \frac{2}{5}$
   c. עלולים להתקע בG – $p(b|a) * p(g|b) = \frac{2}{5} * \frac{\beta-2}{\beta}$

   כלומר סהכ ההסתברות להגיע לכשלון באופציה זו היא: $\frac{1}{5} + \frac{2}{5} + \frac{2(\beta-2)}{5\beta}$
2. אם בטא גדולה/שווה 4: המקסימום הגלובלי הוא בG (או בF במידה ובטא היא 4), כלומר נתכנס לפתרון שאינו אופטימלי במקרים שבהם נתקע בH או בD, כלומר הסתברות של $\frac{1}{5} + \frac{2}{5} = \frac{3}{5}$

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

$$p(b|a) = \frac{2}{5}, p(f|b) = \frac{2}{\beta}, p(g|b) = \frac{\beta - 2}{\beta}$$

אם הסוכן ילך בדיוק שלושה צעדים, זה אומר שהוא יעבור דרך F בדרך לG. לשם כך בתור התחלה נדרוש $\beta > 4$.

כעת נבדוק מה צריך להיות ערך בטא כדי שההסתברות תהיה גדולה מחמישית:

$$\frac{2}{5} * \left(\frac{2}{\beta}\right) > \frac{1}{5}$$

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

$$4 > \beta$$

כלומר לא קיים ערך של בטא עבורו התנאי הנתון בשאלה מתקיים.

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.