236501 - מבוא לבינה מלאכותית

תרגיל בית 1

מרחבי חיפוש

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

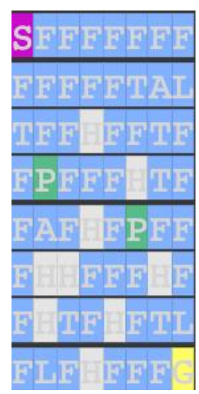
סיפור מסגרת

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 $(S, 0, I, G)$ עבור סביבת האגם הקפוא. כאשר S זה מרחב המצבים, O, זה מרחב האופרטורים, I, זה המצב ההתחלתי וG הוא קבוצת מצבי המטרה. מה גודל מרחב המצבים S? הסבירו.
תשובה:

$$S = \{s | s \in [63..0]\}$$
$$O = \{Down, Up, Left, Right\}$$
$$I = \{0\}$$
$$G = \{63\}$$

אם גודל הלוח בסביבת האגם הקפוא הוא M*N, כך גם גודל מרחב המצבים היות שכל מצב מתאר את מיקום הסוכן על הלוח, ללא תלות במהלכים או הדרך בה הגיע לנקודה זו. לכן במקרה שלנו כאשר הלוח הוא 8*8, גודל מרחב המצבים הוא 64.

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

תשובה: תמיד אפשר לנוע מעלה כל עוד המצב הנוכחי אינו חור. לכן Domain(UP) מחזיר את קבוצת כל המצבים שאינם חור. גם אם אנחנו בקצה העליון הפעולה פשוט תשאיר אותנו במקום ולכן היא חוקית.

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

תשובה: Succ מחזירה לנו את רשימת המצבים אליהם ניתן להגיע מהמצב הנוכחי, כלומר קבוצת המשבצות סביב המצב ההתחלתי. ממצב זה מכיוון שזו הפינה השמאלית העליונה ניתן להישאר במקום ולנוע ימינה או מטה. לכן קבוצת המצבים שתוחזר על המצב ההתחלתי היא 0, 1 ו-8 (כאשר הלוח הינו 8 על 8).

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

תשובה: כן כפי שתיארנו בסעיף הראשון המצבים תלוים רק במיקום הנוכחי של הסוכן ולכן אם נבחר ארבע משבצות צמודות שאינם חורים, למשל מהמצב ההתחלתי, נוכל ללכת למטה, ימינה, מעלה ושמאלה ולחזור במעגל לאותה הנקודה\מצב וליצור מעגל מכיוון שלא סיימנו ונוכחל לחזור על רצף הפעולות הזה שוב אינסוף פעמים.

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

תשובה: מקדם הסיעוף הינו 4 כי קיימת משבצת שממנה ניתן להגיע ל-4 משבצות אחרות, כאשר כל משבצת היא מצב ובהנחה שהוא אינו חור.

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

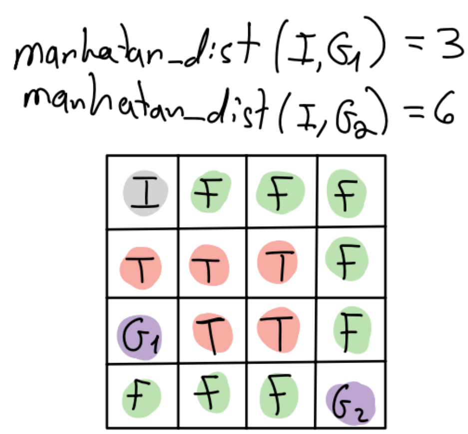
תשובה: אינסוף\לא יגיע כלל בדומה להסבר של סעיף 5.

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

תשובה: כתלות באורך של המסלול הקצר ביותר מנקודת ההתחלה לנקודת סיום כלשהיא, תוך המנעות מחורים. במקרה שלנו – במקרה הטוב ביותר נלך ישירות לפורטל שישגר אותנו לפורטל השני, ומשם ישירות לנקודת הסיום G, סה"כ 9 צעדים.

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.

תשובה: לא, ייתכן למשל שהמצב המטרה הקרוב ביותר לפי מרחק מנהטן נמצא "מאחורי" (תמונה להמחשה מצורפת) משבצות של חורים אשר גורמים לסוכן להיאלץ לבצע עיקוף כדי להגיע אליו דבר אשר מאריך את הדרך בפועל מול הדרך אל מצב מטרה אחר אשר אינו חסום על ידי חורים. מכיוון שבדוגמה השתמשנו רק במשבצות F, המסלול הקצר ביותר הוא גם הקל ביותר והוא מגיע אל מצב המטרה הרחוק יותר מבחינת מסלול מנהטן ממצב ההתחלה בניגוד להנחת סעיף זה.



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?

   תשובה:

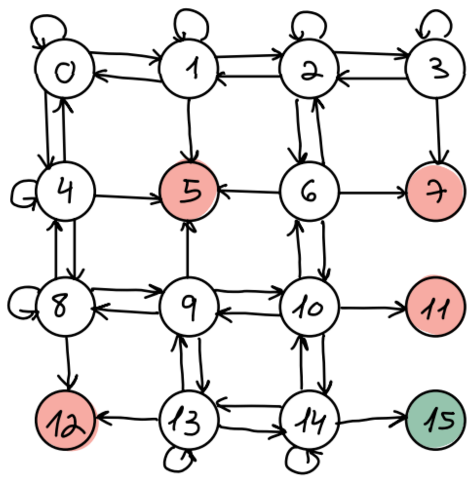
   א. **שהגרף יהיה חסר מעגלים**

   ב. להתחיל את החיפוש מאותה הצומת בגרף ובעץ

   ג. שסדר מעבר על הבנים יהיה זהה בשניהם

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

   תשובה:



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

תשובה: בהנחה שמחיר של כל קשת מספר שלם גדול מ-0 (ואם לא, ניתן לבצע נירמול לשלמים שישמור על היחס), נגדיר פונקציה T מ-G ל-'G שמפרקת קשתות בעלות משקל n>1 ל-n קשתות במשקל 1 כל אחת ומוסיפה n-1 צמתי ביניים שיוגדרו כמצבים חוקיים ולא סופיים, כך יצרנו שקילות בין משקל מסלול ב-G לאורך מסלול ב-'G.

כעת בהרצת G-BFS על גרף 'G, נקבל את המסלול הקצר ביותר בגרף 'G מצומת מקור לצומת היעד ולכן המסלול הקל ביותר בגרף G, מהגדרת הפונקציה T.

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 BFS-G? הסבירו?

תשובה: במהלך חיפוש G-BFS יפותחו 2 צמתים פחות משנוצרו שכן צומת היעד היא בפינה הימנית תחתונה ולכן יש לה שני שכנים, כלומר כאשר נגיע לאחד השכנים ונבצע OPEN לבנים שלו, נמצא את צומת היעד ונעצור. כלומר, פתחנו את כל הצמתים למעט צומת היעד ואחת מצמתי השכנים שלה, שלא ממנו הגענו לצומת היעד.

בגלל שאנחנו ב-BFS וצומת ההתחלה וצומת היעד נמצאות בפינות מנוגדות בגרף, נקבל שאיטרציות הפתיחה של ה-BFS ייפתחו באלכסון ולכן נגלה את צומת היעד רק בהגעה לשכבה האחרונה, כלומר לאחר היווצרות כל הצמתים, כלומר ייווצרו N^2 צמתים.

לכן בסה"כ יפותחו $N^2 - 2$ צמתים וייווצרו N^2 צמתים.

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?
תשובה: באופן כללי DFS אינו שלם מכיוון שבמקרה בו קיימים מעגלים הוא ייכנס ללולאה אינסופית. עם זאת, עבור בעיית האגם הקפוא והלוח הנתון האלגוריתם G-DFS אכן שלם מכיוון שאנו שומרים על הצמתים בהם ביקרנו ולא מפתחים אותם בשנית וכך מונעים היכנסות למעגל אינסופי. מסופיות הלוח אנחנו תמיד נגיע לצומת היעד במידה וקיים מסלול אליו. מבחינת קבילות הוא אינו קביל כי ייתכן שנקבל מסלול חוקי ליעד שאינו אופטימלי, כפי שראינו בהשוואה מול אלגוריתם BFS, בו קיבלנו מסלול טוב יותר.

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?
תשובה: באלגוריתם DFS על עץ, ניתן לבצע את אותה הפעולה שוב ושוב ללא תלות ב-OPEN ו-CLOSED (שלא קיימים). לכן אלגוריתם זה על הלוח לא יעבוד מכיוון שקיים מצב חוקי בו חוזרים לאותה המשבצת עבור משבצות הנמצאות בקצוות הלוח.
מכיוון שסדר הפעולות הוא קודם כל לרדת למטה, אלגוריתם זה על הלוח ירד למשבצת התחתונה השמאלית ויתקע בה כי ימשיך לבצע פעולה חוקית של ירידה למטה אשר מחזירה את הסוכן לאותה המשבצת.

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?
תשובה: אלגוריתם ה-DFS יבחר תמיד להיכנס לעומק לפני חזרה. לפיכך בלוח שלנו הוא יתקדם ב"שרוך" לכיוון הפתרון על ידי ירידה למטה עד הסוף ואז ימינה עד למשבצת הסופית (מסדר תיעדוף הצעדים). לכן, כאורך המסלול יפותחו 2N-2 צמתים. כל צומת במסלול גם ייווצר בתוספת של כל צומת הצמודה אליו, אליה ניתן להגיע. בתוספת צמתים אלו ובהורדת צמתים חופפות, נקבל כי ייווצרו 4N-5 צמתים סך הכל (כלומר המסלול + העמודה והשורה הצמודה למסלול בהתאמה).

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?
תשובה: באלגוריתם DFS backtracking נתון כי צמתים נוצרות בצורה עצלה, כלומר ניצור צומת רק רגע לפני שאנו מפתחים אותה. לכן לא נוצרות יותר צמתים משמפותחות. מנגד, כל צומת שמפותחת בהכרח נוצרת קודם לכן ולכן נקבל שערכים אלו שווים. מ-DFS ומהנתון שבלוח לא קיימים חורים או פורטלים, נקבל כי האלגוריתם ייכנס לעומק הכי עמוק שהוא יכול עד למציאת צומת היעד (ירידה עד הסוף למטה וימינה למשבצת הימנית התחתונה, בדומה לסעיף הקודם). מכאן שייווצרו ויפותחו 2N-2 צמתים סך הכל.

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.
   תשובה: שינוי לבעיית החיפוש לפתרון בעיה זו הוא שינוי הגרף G כך שכל זוג קשתות מאוחד לקשת אחת בעלת משקל ששווה לסכום המשקלים של שתיהן. כך למשל אם קיים מסלול מצומת I לצומת K באורך 2, ומסלול מהצומת K לצומת G באורך 2, לאחר השרשור יהיו שני מסלולים מהצומת I, אחד ל-K באורך 2 ואחד ל-G באורך 4. אמנם מ-K לא יהיה מסלול ל-G יותר, אך המסלול קיים ומובטח לנו ב-DFS-L שהוא עובר על כל המסלולים בעומק L. כך נקבל עבור מסלול באורך d ב-G, מסלול מקביל באורך d/2 ב-G'.

   לאחר ביצוע שינוי זה, נריץ L-DFS עם המגבלה של ריק על אורך המסלול. כפי שהסברנו כלל המסלולים יתקצרו בחצי אך נשמור על המשקל האמיתי של כל מסלול, כך נאפשר מציאת המסלול מהמקור ליעד אשר ידוע כי אורכו D ועל כן אורכו D\2 ב-G', מהגדרתו. משלמות DFS נמצא פתרון.

   b. יבש (1נק׳): האם השתנה מקדם הסיעוף? מה מקדם הסיעוף החדש b'? אם כן רשמו את התשובה כתלות ב b (מקדם הסיעוף בבעיה המקורית).
   תשובה: כן, מקדם הסיעוף החדש הוא בדיוק פי 2 ממקדם הסיעוף הישן וזה למעשה ה-trade off שביצענו על מנת לפתור את הבעיה עם חצי מהעומק הקודם. כלומר b' = 2*b, לפי הסבר שינוי הגרף G בסעיף הקודם

.c יבש (1נק׳): מה סיבוכיות הזמן והמקום החדשים בדם? 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?

תשובה: מהגדרת L-DFS והשינוי שביצענו, סיבוכיות הזמן החדשה הינה $O\left(b'^{\left(\frac{d}{2}\right)}\right) = O\left((2b)^{\frac{d}{2}}\right)$ וסיבוכיות המקום החדשה הינה $O\left(b' * \frac{d}{2}\right) = O(b * d)$. זאת בהשוואה ל-L-DFS רגיל עם עומק d אשר סיבוכיות הזמן שלו הינה $O(b^d)$ וסיבוכיות המקום שלו הינה $O(b * d)$.

תשובתנו הייתה משתנה אם היינו משתמשים ב-backtracking רק מבחינת סיבוכיות המקום (סיבוכיות הזמן אותו הדבר). סיבוכיות המקום החדשה הייתה $O(d)$.

.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 שבה DFS-L במרחב המקורי עדיף. בתשובתכם התייחסו למספר צמתים שפותחו. דוגמאות יכולות להיות כלליות ולא בהכרח מסביבת הfrozen lake .

תשובה: מהסעיף הקודם ראינו שסיבוכיות המקום לא משתנה ועל כן ניתן לייעל רק את סיבוכיות הזמן. לכן עדיף להשתמש במרחב החיפוש החדש כאשר $(2b)^{\frac{d}{2}} < b^d$ כי מסעיף קודם נקבל ש-L-DFS במרחב חיפוש החדש יבוצע מהר יותר. באופן דומה נעדיף את מרחב החיפוש הישן כאשר אי-השוויון מתהפך.

שאלה 5 – ReverseDFS (6 נק׳):

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

תשובה: כן , בהינתן וידוע D חסם עליון על מרחק למצב מטרה עבור DFS-L עם פרמטר D (כלומר איטרציה ראשונה של הלולאה הראשית) בהכרח DFS יצליח למצוא פתרון מכיוון שDFS עם פרמטר שהוא חסם עליון יעבור על כל מסלול עד אורך זה ובפרט על פתרון אפשרי לצומת יעד.

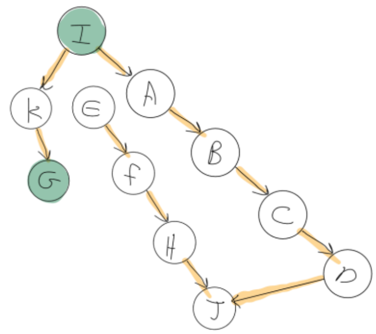
.b (1 נק׳) האם האלגוריתם אופטימלי? אם כן, הוכיחו. אם לא, ספקו דוגמה נגדית

תשובה: כן, נניח כי אורך הפתרון האופטימלי הוא a. ידוע כי a<D ולכן לכל איטרציה גדולה של הלולאה עד a , יתקבל עבור DFS-l לפחות מסלול ליעד באורך a. ועל כן הלולאה תמשיך ולא תיעצר. עבור האיטרציה בה L=a נמצא את הפתרון האופטימלי מכיוון שdfs עובר על כל המסלולים עד אורך a ובפרט המסלול ליעד. אם היה מסלול קצר יותר זהו לא היה מסלול אופטימלי והיינו מקבלים a קטן יותר. באיטרציה הבאה כבר לא נמצא עוד מסלול בDFS והאלגוריתם יעצר עם 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.

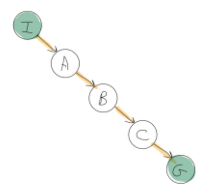
תשובה:

דוגמה עבור ID-DFS עדיף על ReverseDFS :



בדוגמה זו, ReverseDFS יקבל חסם עליון D, נניח כעומק המסלול מ-I ל-E ויבחר לרוץ קודם מ-I ל-A (נניח לפי תעדוף של סדר לקסיקוגרפי), נקבל שReverse DFS ירוץ עד ל-E לפני מעבר לצומת K ומציאת צומת היעד G, חיפוש בזבזני בהשוואה לשימוש ב-ID-DFS אשר במקרה זה היה מוצא את G לאחר שני איטרציות ובהגעה לעומק של הצומת B בלבד.

דוגמה עבור ReverseDFS עדיף על ID-DFS :



במקרה זה עבור I-DFS נצטרך לעבור מ0 עד אורך המסלול האופטימלי, כל פעם נגלה קשת נוספת אך נצטרך בכל איטרציה להתחיל את חיפוש המסלול מחדש. לעומת זאת, עבור ReverseDFS עם חסם D=4 כבר באיטרציה הראשונה של הלולאה הראשית נבצע DFS על הגרף ונגלה באופן ישיר את המסלול תוך מעבר על הצמתים במסלול מהמקור ליעד בלבד.

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

תשובה: במקום לעדכן את L להיות L-1 בכל איטרציה, נעדכן אותו להיות אורך המסלול המתקבל מה-DFS באיטרציה הנוכחית, כך נחסוך באיטרציות "בזבזניות" אשר D גדול מאורך המסלול מהמקור ליעד בהן ונעבור מיד לפתרון האופטימלי תוך שני איטרציות בלבד.

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.

תשובה: אלגוריתם UCS פועל באותו האופן כמו BFS עבור בעיות חיפוש בהן משקל כל הקשתות זהה, מכיוון ש-UCS תמיד מוצא את המסלול הקל ביותר וב-BFS ימצא אותו רק אם הוא גם הקצר ביותר – דבר אשר מתרחש תמיד ב-BFS רק כאשר כל הקשתות במשקל זהה.

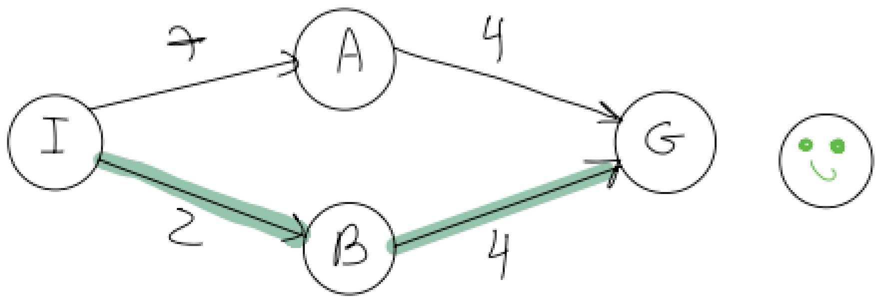
3. יבש (1 נק׳): האם בבעיית החיפוש שלנו, עבור לוח NxN, האלגוריתם הוא שלם? האם הוא קביל?

תשובה: בבעיית החיפוש שלנו, UCS אכן שלם מכיוון שהוא שומר בדומה לשאר הסוכנים, רשימה של OPEN ו-CLOSED אשר מבטיחה המנעות ממעגלים. כמו כן מכיוון ש-UCS בוחר תמיד בצומת שתביא אותו ליעד במחיר הנמוך ביותר, הוא נמנע מבורות

ובוחר בצמתים אשר יקדמו אותו אל היעד בעלות הנמוכה ביותר. לכן במידה וקיים מסלול מהמקור אל היעד, UCS ימצא אותו ולכן שלם. סיבה נוספת לשלמות היא שפונקצית המחיר עבור הבעיה שלנו חסומה מלמטה על ידי מספר שלם גדול מ-0 (1) שכן לא קיימת קשת בעלת עלות נמוכה מ-1, לכן מובטחת שלמות לפי הגדרת UCS בתרגולים. בנוסף, מאופן בחירת הצמתים המתוארת הוא גם קביל כי הוא תמיד בוחר רק את הצמתים אשר יביאו אותו ליעד בעלות הזולה ביותר.

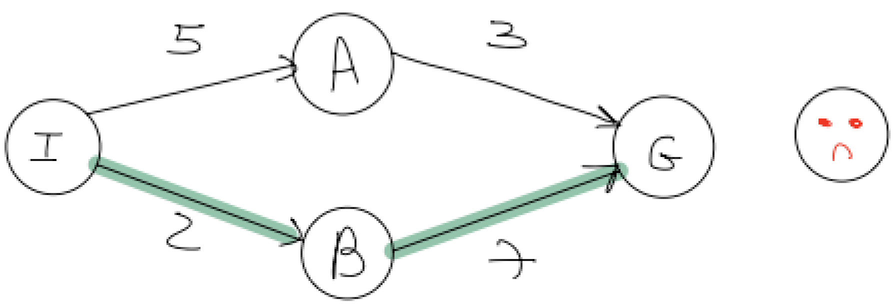
4. יבש (2 נק׳): דן טעה במימוש של אלגוריתם UCS ובטעות בדק בעת יצירת הצומת האם היא צומת מטרה במקום בפיתוח שלה. הביאו דוגמה לגרף חיפוש שעבורו דן יחזיר בכל זאת את המסלול הקל ביותר ודוגמה לגרף חיפוש שעבורו דן לא יחזיר את המסלול הקל ביותר. עבור כל דוגמה הסבירו מה המסלול והעלות ש-UCS השגוי החזיר, ומה המסלול 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.
תשובה: כאן נשים לב שבדיקה לפני הזמן בסך הכל מונעת מאיתנו השוואה של מסלולים טובים יותר באורך זהה, לכן הוא יימצא את המסלול הקל ביותר במקרים בהם הוא בחר בו ראשון (מהגדרת סדר הפעולות) ולא יבחר בו כאשר יבחר במסלול אחר ראשון.

דוגמה שבה המימוש של דן עדיין מוצא את המסלול הקל ביותר:



בצעד הראשון האלגוריתם יבחר מעבר לצומת B מכיוון שהקשת אליה זולה יותר מהקשת אל צומת A. לאחר מכן האלגוריתם יפתח את B, יגלה את G שהוא צומת יעד ויסיים עם המסלול הקל ביותר (מחזיר מסלול מ-I ל-G דרך B בעל משקל 6, שהוא הקל ביותר).

דוגמה שבה המימוש של דן לא מוצא את המסלול הקל ביותר:



בצעד הראשון האלגוריתם שוב יבחר בצומת B מכיוון שהקשת אליה זולה יותר מהקשת אל צומת A. לאחר מכן הוא יפתח את B, יגלה שהוא צומת יעד ויסיים מבלי להשוות אל הקשת מ-I ל-A העליונה שמובילה למסלול פתרון הקל ביותר! (מחזיר מסלול מ-I ל-G דרך B בעל משקל 9, כאשר הקל ביותר הוא מ-I ל-G דרך A בעל משקל 8).

שאלה 7 - יוריסטיקות (8 נק׳):

1. יבש (1 נק׳): בהיתן שתי יוריסטיקות קבילות h_1, h_2. האם $h = \min\{h_1, h_2\}$ קבילה? אם כן, הוכיחו. אם לא, הפריכו.
תשובה: כן. מכיוון שלפי הגדרת הקבילות מתקיים $0 \le h2 \le h*$ וגם $0 \le h1 \le h*$ לכן גם בהכרח מתקיים <=
$0 \le \min\{h2, h1\} \le h*$ ולכן גם זו יוריסטיקה קבילה על פי הגדרה.

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.
תשובה: בדומה לסעיף הקודם גם יוריסטיקה זו קבילה מכיוון שמתקיים ש- 0< max(h1, h2) < h*.

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.
תשובה: עבור יוריסטיקה h1, יהי צומת S1 ויהי צומת S1' שכנה אליה, הפרש היוריסטיקות h1(s1)-h1(s1')<=cost(s1, s1') מהגדרת יוריסטיקה עקבית. באופן דומה עבור יוריסטיקה h2, הפרש היוריסטיקות h2(s1)-h2(s1')<=cost(s1, s1') מהגדרת יוריסטיקה עקבית. נקח את היוריסטיקה מינימלית מביניהן עבור כל איבר, נניח בה"כ שעבור s1 המינימלי הינו h2. אם עבור s1' המינימלי הוא h2, היוריסטיקה תתקיים וסיימנו. באופן דומה במקרה של שוויון בין h1 ל-h2 קיבלנו יוריסטיקה זהה ל-h2 ולכן סיימנו גם כן. אחרת בהכרח נקבל שזה h1 ולכן נקבל יוריסטיקה שמקיימת:

$$h2\ (s1) - h1\ (s1') \le\ h1\ (s1) - h1\ (s1') \le cost(s1, s1')$$

*מהמינימליות של h2(s1).

** נציין לשלמות שהיוריסטיקה של מצבים סופיים נשמרת אפס ולכן היוריסטיקה המתקבלת אכן עקבית.

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.
תשובה: באופן דומה לסעיף הקודם, נניח בה"כ כי h1 הוא המקסימלי על s1. אם הוא מקסימלי גם על s1', סיימנו מהנתון כי h1 הינה יוריסטיקה עקבית. אחרת h2 הוא מקסימלי עבור s1' לכן מתקבל:
.

$$h1(s1) - h2\ (s1') \le h1(s1) - h1(s1') \le cost(s1, s1')$$

*ממקסימליות h2(s1)

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.
תשובה: קבילה תמיד. מכיוון שיוריסטיקת מנהטן מקיימת את תנאי הקבילות ($0 \le h_{sap}(s) \le h*$), אז המינימום בינה לבין כל פונקציה יוריסטית אחרת תמיד יהיה קטן מ-h*. מכיוון ש-cost(p)>=0, תנאי הקבילות מתקיים בצורה מלאה.

6. יבש (1 נק׳): האם היוריסטיקה h_{SAP} עקבית על כל לוח? אם כן הסבר, אם לא הבא דוגמה נגדית.
תשובה: הפרש היוריסטיקות בין שני צמתים שכנים הוא תמיד לכל היותר 1 היות שלוקחים את המינימלי בין עלות פורטל (100) ומרחק מנהטן מצומת היעד. מרחק משתנה באחד כאשר זזים משבצת אחת כידוע, ומכיוון שלוקחים את המינימלי, כאשר יהיו שני צמתים שכנות אשר היוריסטיקה תבחר בעלות הפורטל עבור אחד ובמרחק מנהטן עבור השני, הרי שנהייה בדיוק בתפר עבור מרחק מנהטן = 100 ולכן ההפרש יהיה לכל היותר 1 גם במקרה זה. מכיוון שעלות המינימלית של צומת הוא 1, תנאי העקביות מתקיים עבור כל 2 צמתים שכנות.

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.
תשובה: כן בדומה לסעיף 5 יוריסטיקת מנהטן קבילה על כל לוח והמינימום בינה לבין יוריסטיקה אחרת תמיד קביל גם כן.

8. יבש (1 נק׳): האם היוריסטיקה h_{MSAP} עקבית על כל לוח? אם כן הסבר, אם לא הבא דוגמה נגדית.
תשובה: מקרה זה זהה לחלוטין לסעיף 6 עם תוספת יחידה שמאפשרת בחירת מצב יעד שונה כאשר צומת היעד הקרובה ביותר משתנה, דבר אשר קורה רק כאשר ההפרש בין המרחקים הוא לכל היותר אחד, לכן גם כאן היוריסטיקה עקבית.

באופן כללי נבחין כי כאשר מדובר ביוריסטיקה המתבססת על מרחקים ההפרש בין שני צמתים שכנים הוא לכל היותר 1.

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?
תשובה: האלגוריתם אכן שלם מהגדרת Greedy BFS אם מרחב החיפוש הוא סופי, האלגוריתם שלם. באופן דומה לסעיפים קודמים מתיחזוק OPEN ו-CLOSED אנו מוודאים הימנעות ממעגלים וכך הופכים את מספר המצבים שלנו לסופי (כי לא נחזור לאותו מצב שוב). האלגוריתם איננו קביל כי הוא אינו מוצא את המסלול האופטימלי ביותר מכיוון שהוא מסתמך על היוריסטיקה שאינה מושלמת. במקרה שלנו מדובר ביוריסטיקת HSAP אשר לוקחת את המינימום בין מרחק מנהטן של הצומת ועלות קשת לפורטל. במקרה שלנו מכיון שעלות קשת כזו היא 100 והלוח הוא 8 על 8, מרחק מנהטן המקסימלי האפשרי הוא 16 ולכן היוריסטיקה של כל צומת תהייה מרחק מנהטן מהמטרה – אך זו לא יוריסטיקה מושלמת כי היא יכולה לגרום לנו לתעדף את הצמתים הקרובים יותר אפילו אם פניה הפוכה לכיון המטרה יכולה להביא אותנו דרך צמתים זולות יותר, מה שיוביל למסלול קל יותר.

3. יבש (2 נק׳): תנו יתרון וחיסרון של אלגוריתם Greedy Best first Search לעומת Beam Search.
תשובה: יתרון של Greedy BFS על Beam Search הוא באופטימליות הפתרון. Beam Search מסתמך על היוריסטיקה אפילו יותר היות שהוא מתחזק רק K צמתים ב-OPEN וזורק את אלו עם הערך היוריסטי הגדול ביותר (מעל K צמתים) ולכן מאבד מסלולים העוברים בצמתים אלו גם אם הם חלק ממסלול אופטימלי יותר בהסתכלות הרחבה.

מנגד, חסרון של Greedy BFS על Beam Search הוא ביעילות, הן מבחינת זמן והן מבחינת זיכרון היות ש-Beam Search עובר על פחות צמתים ומחזיק פחות מהן בזיכרון, מההסבר לעיל (מחזיק עד K צמתים).

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.
   תשובה: הערה: מסתמכים על הערת מתרגל בפיאצה שהשאלה מתיחסת למקרה כללי ולא ספציפית ללוח שבמחברת (https://piazza.com/class/lgxyn1wkaxz1hs/post/84)

   עבור גרף בו קיים מסלול יחיד מצומת המקור אל היעד (לדוגמה שרוך או לוח בו כל שאר הצמתים הם חור חוץ ממסלול יחיד), נקבל בשני המקרים את אותו המסלול בעל משקל זהה ולכן טענה זו שגויה.

   אם יש מסלול יחיד הסוכן שלנו יתקדם באותה הדרך ללא תלות במשקל של היוריסטיקה.

   b. יוריסטיקה כללית (לא בהכרח קבילה) h. אם כן הסבירו. אם לא, ספקו דוגמה נגדית.
   תשובה: בדומה לסעיף הקודם נקבל סתירה לטענה עם מסלול יחיד בעל משקל זהה עבור שני המקרים.

   שוב נדגיש כי אין כאן תלות בקבילות היוריסטיקה מכיוון שמדובר במסלול יחיד אל המטרה.

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 ל-A*. באילו מקרים הייתם מעדיפים להשתמש בכל אחד מהם?

תשובה: יתרון של IDA* על A* הינו בצריכת הזיכרון שכן ב-IDA* צריכת הזיכרון הינה לינארית באורך המסלול בניגוד ל-A* בו צריכת הזיכרון פרופורציונית למספר הצמתים שיצרנו ולכן יכולה להיות גדולה יותר.

חיסרון של IDA* על A* הוא שהוא עשוי לפתח מצבים חוזרים מבלי לדעת שביקר בהם בעבר (בעץ) בעוד A* נמנע מפיתוחים חוזרים כאשר אין שיפור.

שאלה 11 – A* epsilon (6 נק׳):

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

תשובה: יתרון של אפסילון הינו במהירות החיפוש, אשר טובה יותר בעקבות הזנחות שהוא מבצע (מאפשר בחירה לא אופטימלית בתקווה למציאת פתרון מהר יותר).
בהמשך ישיר ליתרון, החיסרון שלו הוא ביעילות מכיוון שבניגוד ל-A* ייתכן מקרה בו לא מחזיק את הפתרון האופטימלי ביותר (אינו קביל תמיד ומאפשר בחירה לא אופטימלית בתקווה למציאת פתרון מהר יותר).

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.

תשובה: נציע להשתמש ביוריסטיקה המבוססת על מרחק מנהטן של צומת מצומת היעד הקרובה ביותר. היא תחשב עבור כל צומת את מרחק המנהטן מצומת היעד הקרובה אליה. יוריסטיקה זו תוודא שהסוכן שלנו יבחר תמיד צעדים שמקדמים אותו אל המטרה, כלומר תבחר את המסלול הקצר ביותר בהינתן לוח ללא חורים או פורטלים.

נבחר את היוריסטיקה שמתארת את המרחק מכל צומת אל היעד הקרוב ביותר אליה מבחינת מרחק מנהטן.

במידה ונבחר את הצומת הבא לפיתוח מתוך FOCAL לפי הערך g(v) נקבל צמתים שמסתמכים על התוצאות שכבר ראינו (המשקל של המסלול המוביל אליהן), לעומת היוריסטיקה שמסתמכת על העתיד (המרחק מהצומת אל היעד) ומתעדפת צמתים שמבטיחים יותר יוריסטית. לכן אם המגרש שלנו מאפשר תנועה פתוחה מהצומת הנוכחי ליעד, הסתמכות על היוריסטיקה תאפשר הגעה מהירה יותר עם פחות פיתוחים. מנגד עבור מסלול עם מכשולים רבים (חורים ופורטלים), ניתן להניח שהסתמכות על G (העבר) תועיל יותר או במידה שווה ליוריסטיקה.

לכן עבור מקרה ממוצע נקבל תוצאה דומה מבחינת מספר הפיתוחים, המסלול ועלותו עבור היוריסטיקה ועבור הערך G.

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.

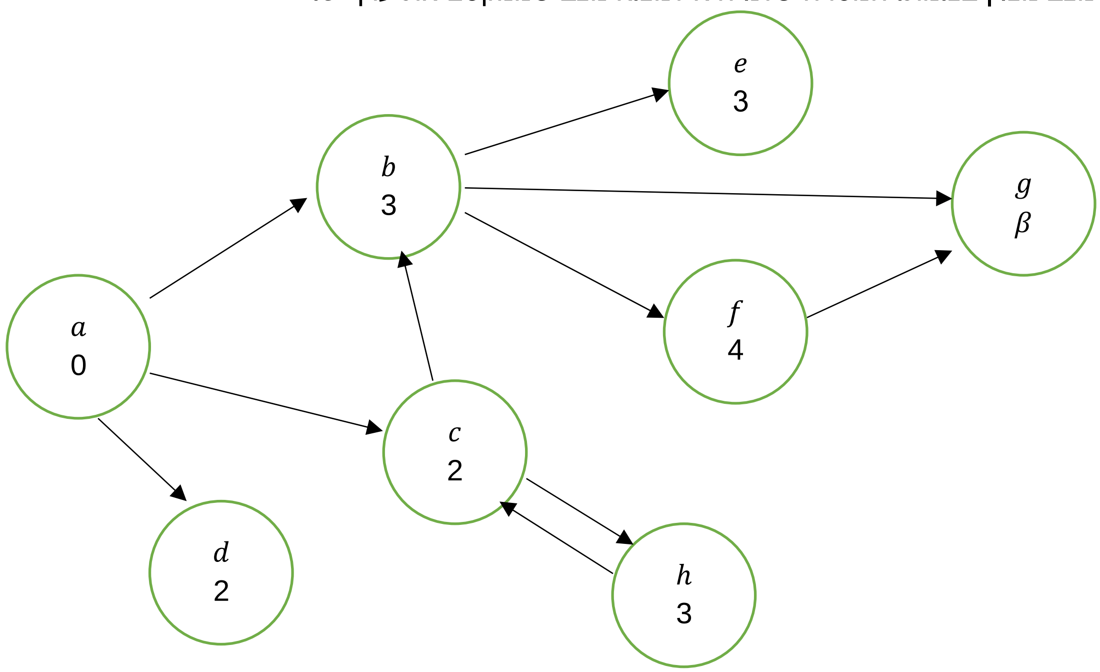
תשובה: התוצאות שקיבלנו אכן תואמות לציפיות שלנו. נבחין כי המסלולים שקיבלנו מ-UCS הם אכן המסלולים הקלים ביותר ביחס לכל שאר האלגוריתמים. בנוסף עבור WA* כאשר W=0.7 נבחין כי מקבלים תוצאה זהה ל-UCS, כלומר הסוכן מוצא את המסלול הקל ביותר אך עושה זאת בזמן קצר יותר עם פחות צמתים ב-EXPANDED. עבור BFS ו-DFS נבחין כי BFS יעיל יותר מבחינת עלות המשקל המוחזר אך מפתח כמעט את כל המגרש ולכן צריכת הזיכרון שלו גדולה יותר מאשר DFS.

שוב התוצאות תואמות את כל הציפיות שלנו עבור הסוכנים השונים וכמוסבר עבור WA*, נבחין כי ככל שהיוריסטיקה מיודעת יותר, אנו מתקרבים לתוצאה טובה יותר עם פחות צמתים ב-EXPANDED, עבור אלגוריתמים שמתבססים על היוריסטיקות.

בנוסף, אלגוריתם GREEDY פועל באופן מהיר יחסית (בעל מספר נמוך של צמתים פתוחים), אבל מוצא את המשקל בעל העלות הגדולה ביותר ולכן הכי פחות אופטימלי.

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 Stochastic Hill Climbing.

כמו כן ידוע כי $\boldsymbol{\beta > 4}$.

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

   תשובה:

   p(d|a) = 2/7

   p(c|a) = 2/7

   p(b|a) = 3/7

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

   תשובה: במקרה המקסימלי האלגוריתם יבצע 4 צעדים לפני עצירה. זאת מכיוון שהאלגוריתם עוצר בעת הגעה לצומת היעד או כאשר הוא מגיע למקסימום לוקאלי – כלומר צומת אשר כל השכנים שלו בעל ערך שאינו משפר. בהתבוננות על הגרף הנתון, נבחין כי המסלול שמקיים את תנאים אלו הארוך ביותר הינו a->c->b->f->g

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

   תשובה: כן, הוא יתכנס למקסימום הגלובלי, זאת מכיוון שאינו יכול לעבור ממצב B למצב E כי לא קיימת אפשרות ל- sideway steps (מעבר לצומת בעל ערך שווה לשלך). לכן הוא מחויב לעבור מ-B ל-G ולסיים או ל-F ואז ל-G ולסיים.

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

תשובה: המצבים אשר הגעה אליהם גורמים לכך שהאלגוריתם יתכנס לפתרון לא אופטימלי הם המקסימומים הלוקאליים שקיימת הסתברות כלשהי להגיע אליהם (בניגוד ל-E כמוסבר בסעיף הקודם). לכן המצבים האפשריים הם רק D ו-H. נחשב את ההסתברות להגעה אליהם:

p(d|a) = 2/7
p(h|c) = 1/2
p(c|a) = 2/7
p(h|a) = 2/7 * 1/2 = 1/7

סה"כ נקבל הסתברות של 3/7 = 2/7 + 1/7 להגעה לפתרון לא אופטימלי.

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

תשובה: נבחין כי באופן כללי קיימים שני מסלולים בעלי בדיוק 3 צעדים מצומת המקור אל המקסימום הגלובלי (הצומת G).

מסלול 1: a->b->f->g
מסלול 2: a->c->b->g

נחשב את סכום הסתברויות המסלולים הנ"ל כתלות בערך בטא ונבדוק מתי סכום זה שווה לחמישית כנדרש:

$a \to b \to f \to g:$

$$P_1 = P(b|a)\cdot P(f|b)\cdot P(g|f) = \frac{3}{7}\cdot\frac{1}{1+(\beta-3)}\cdot 1$$

$a \to c \to b \to g:$

$$P_2 = P(c|a)\cdot P(b|c)\cdot P(g|b) = \frac{2}{7}\cdot\frac{1}{2}\cdot\frac{\beta-3}{1+(\beta-3)}$$

נקבל:

$$P_1 + P_2 = \frac{1}{5}$$

$$\frac{3}{7}\cdot\frac{1}{\beta-2} + \frac{1}{7}\cdot\frac{\beta-3}{\beta-2} = \frac{1}{5}$$

$$\Longrightarrow \boxed{\beta = 7}$$

לכן קיבלנו כי הדרישה מתקיימת עבור בטא השווה ל-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.