**מבוא לבינה מלאכותית - 236501**

**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**](https://piazza.com/technion.ac.il/spring2023/236501) **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:

o **65% - המסמך היבש.**

o **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 תינתנּה 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 ״, 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 עבור סביבת האגם הקפוא. כאשר זה מרחב המצבים, , זה מרחב האופרטורים, , זה המצב ההתחלתי ו הוא קבוצת מצבי המטרה. מה גודל מרחב המצבים S? הסבירו.
  3. יבש (1 נק׳): מה תחזיר לנו הפונקציה Domain על אופרטור 2 (UP)?
  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}

* 1. 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.

* 1. 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.

* 1. 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.

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

נבחון 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.

* 1. יבש (1 נק׳): עבור לוח כללי בסביבת ה frozen lakeהמסלול 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:

|  |  |  |  |
| --- | --- | --- | --- |
| FROZEN | FROZEN | PORTAL | START |
| HOLE | HOLE | HOLE | HOLE |
| FROZEN | FROZEN | GOAL\_1 | FROZEN |
| HOLE | HOLE | HOLE | HOLE |
| GOAL\_2 | PORTAL | FROZEN | FROZEN |

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 ןלכן המרחק אליו סופי והמסלול הקל ביותר במקרה הזה הינו למצב המטרה "הרחוק יותר".

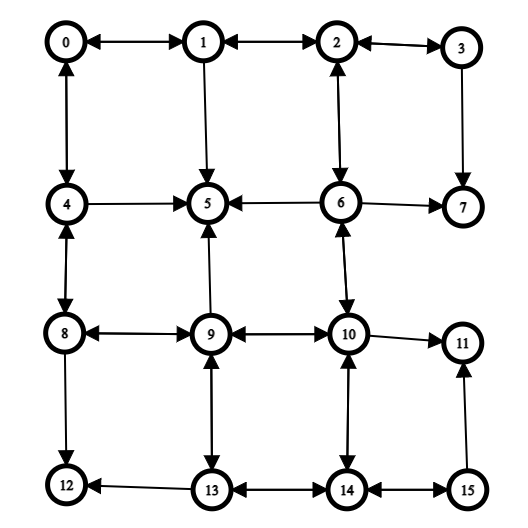
### שאלה 2 – Breadth First Search-G (7 נק׳):

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 לא יהיה בו מעגלים כך שבעצם לא נחזור על צמתים שכבר היו ב CLOSE,OPEN ואז האלגוריתם על עץ יפעל כמו האלגוריתם על גרף.

* 1. 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.

* 1. 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 המקבלת את גרף המצבים ויוצרת גרף חדש ובעזרתה למצוא את המסלול האופטימלי בגרף .

רעיון : לכל קשת בגרף G נכניס קשת "שמקבילה לה" בגרף 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. לכל :

אם weight(e = (u,v) ) = 1:

*אחרת: (משקלה גדול מ 1)*

*נסמן: weight( e = (u,v) ) = c*

*נסמן המסלול :*

*עכשיו בסיום האלגוריתם הנ"ל מתקבל לנו גרף G’ כך שאם הריץ עליו BFS-G מתקבל המסלול האופטימלי כי עכשיו אורך המסלול פרופורציונלי לגודלו גם כן.*

* 1. יבש (2 נק׳): נתון לוח בגודל NxN, ללא חורים, ללא Portals, המכיל 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?

לפי אופן פעולת BFS הוא יפתח את כל הצמתים הקרובים יותר לפני אלו שרחוקים יותר. ולכן לפני שנפתח צומת היעד, נפתח את כל שאר הצמתים בגרף ונקבל שמפתחים צמתים כי לצומת היעד יש שני שכנים ועוצרים כאשר מגלים אותו דרך אחד השכנים. אך יווצרו מצבים בגרף כי שני השכנים מווצרים אותם.

### 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 (לא נלך במעגלים כי שומרים CLOSE,OPEN) ולכן אם קיים פתרון אז נמצא אחד כי נמשיך להתקדם עד שבסוף נגיע ליעד(כי לא נלך במעגלים). לעומת זאת, האלגוריתם אינו קביל כי לא בהכרח מוצאים את הפתרון האופטימלי. יתכן שבריצת DFS נמצא מסלול ארוך/כבד יותר לפני המסלול האופטימלי ובמקרה כזה נחזיר אותו למרות שהוא לא פתרון אופטימלי.

1. 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 יתכן שעכשיו נלך במגלים על מצבים שביקרנו בהן בעבר כי עכשיו לא שומרים OPEN,CLOSE ולכן עלולים לחזור על מצב שביקרנו בו בעבר וכך המסלול עלול להיות אינסופי. 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.

1. יבש (2 נק׳): נתון לוח בגודל NxN, ללא חורים, ללא Portals, המכיל משבצות רגילות (F,T,A,L) מצב התחלתי בפינה השמאלית עליונה ומצב מטרה בפינה הימנית תחתונה. כמה צמתים יפותחו וייווצרו 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 – 2צמתי 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 : 2(N-1) + 1 + 2(N-2) = 4N-3 מצבים.

1. יבש (2 נק׳): נתון לוח בגודל NxN, ללא חורים, ללא Portals, המכיל משבצות רגילות (F,T,A,L) מצב התחלתי בפינה השמאלית עליונה ומצב מטרה בפינה הימנית תחתונה. כמה צמתים יפותחו וייווצרו במהלך חיפוש backtracking DFS-G? הסבירו?

עכשיו מווצרים רק את המצב הבא בפונקצית 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 אך ריק מגביל את החיפוש של ג׳רי לעומק .
   1. יבש (2 נק׳): הציעו שינוי **לבעיית החיפוש** 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 אנחנו בפועל הגענו לעומק של d כי כל פעולה שביצענו בדרך מקבילה לשתי פעולות במרחב האוריגינלי. (S,,, כאשר מתקיים ש:

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 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.*

* 1. יבש (1נק׳): האם השתנה מקדם הסיעוף? מה מקדם הסיעוף החדש ? אם כן רשמו את התשובה כתלות ב (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:

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

כרגע סיבוכיות הזמן הינה: וסיבוכיות המקום הינה : *. כי מריצים אלגורים עם חסם על העומק ששווה ל וזה מה שקובע את הסיבוכיות באלגוריתם זה. אם היינו משתמשים בבקטרקינג עם DFS-L אז היינו מייצרים צמתים רק לפני השימוש בהן ולכן סיבוכות המקום הייתה משתנה ל וסיבוכיות הזמן הייתה נשארת כמו שהיא.*

* 1. 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 ה frozen lake.

לחזור עם דוגמה!

### 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.

* 1. (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 נקטכ לפני שהוא מוצא פתרון לבעיה)

* 1. (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.

* 1. (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.

ReverseDFS ו IDDFS פועלים באופן דומה כאשר ID מתחיל מ L = 0 ומגדיל אותו עד שימצא פתרון ולכן ברגע שהוא מוצא פתרון הוא יודע שהגיע לפתרון האופטימלי. לעומת זאת, Reverse מתחיל L = D ומקטין את L עד שהוא לא יכול למצוא פתרון ואז הוא יודע שהפתרון האחרון שמצא הוא האופטימלי. ולכן עדיף להשתמש ב Reverse כאשר אנחנו יודעים שיש לנו חסם הדוק שקרוב לעומק הפתרון האמיתי, ועדיף להשתמש ב ID כאשר החסם שלנו אינו הדוק.

* 1. (2 נק׳) הציעו כיצד ניצן לייעל את האלגוריתם. רמז: האם אתם יכולים לחשוב על צעד עדכון עדיף לL?

ניתן ליעל את האלגוריתם על ידי שמירת ערך prev = D בהתחלה ששומר את הערך הקודם של L שחישבנו עבורו פתרון לבעייה (שונה מכשלון) ונחלק את L ב 2 כך שאם מקבלים failure יודעים שחרגנו מתחת לגבול ולכן נריץ ו נחשב את L החדש על ידי הממוצע בין L הנוכחי ו prev וכנל עד שנמצא פתרון כך שכל פעם שמוצאים פתרון לבעיה מעדכנים את prev להיות L (בתנאי ש L הוא קטן ממנו prev =).

### 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.

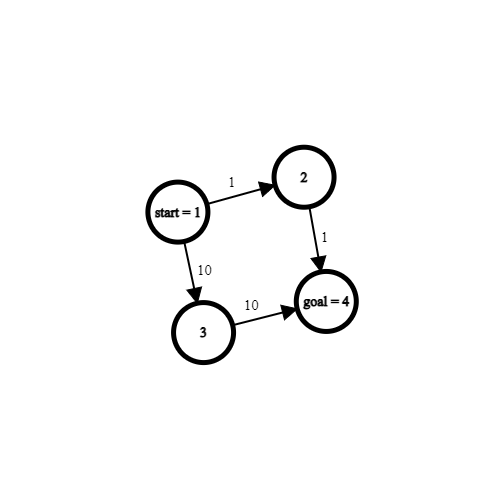
עבור בעיות בהן משקל הקשתות הינו אחיד בכל רמה מרמות הגרף. UCS תמיד יבחר בקשת הזולה ביותר ברמה וכיוון שכולם באותו משקל אז נעבור על כולן כמו BFS.

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

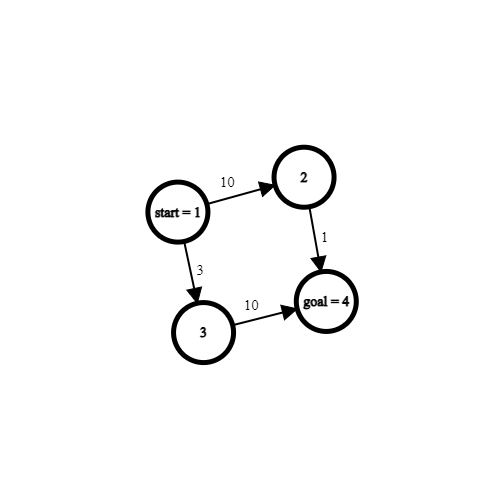
האלגוריתם הינו שלם כיוון שכל המשקלים על הקשתות בגרף שלנו הינם חיוביים ממש וחסומימם מלמטה על ידי 1 נקבל שלפי מה שראינו בכיתה האלגוריתם הוא שלם וגם הגרף שלנו הוא סופי. והוא קביל גם כן כי מחזיר המסלול הזול ביותר כאשר מגלה צומת היעד.

1. 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׳): בהיתן שתי יוריסטיקות קבילות . האם קבילה? אם כן, הוכיחו. אם לא, הפריכו.

יוריסטיקה h קבילה אם:

כיוון ש: קבילות מתקיים: ובפרט מתקיים ש :

לכן נתקבל ש: קבילה.

1. יבש (1 נק׳): בהיתן שתי יוריסטיקות קבילות . האם קבילה? אם כן, הוכיחו. אם לא, הפריכו.

יוריסטיקה h קבילה אם:

כיוון ש: קבילות מתקיים: ונניח בה"כ שמתקיים : עבור s כלשהו, והמקרה השני שבו מקבלים בו :

לכן נקבל ש: קבילה בכל מקרה.

1. יבש (1 נק׳): בהיתן שתי יוריסטיקות עקביות . האם עקבית? אם כן, הוכיחו. אם לא, הפריכו.

יוריסטיקה h עקבית אם:

ניקח למשל את היורסטיקות :

בלוח 3x3 כך שאפשר ללכת ימינה, למעלה, למטה, שמאלה ובאלכסון. כאשר העלות של כל אחד מהמעברים הינו

|  |  |  |
| --- | --- | --- |
| 2 | 1 | Start = 0 |
| 5 | 4 | 3 |
| Goal = 8 | 7 | 6 |

כך שמתקבל : ולכן מתקבל ש: וגם כלומר : לכן קיבלנו ש: אינה עקבית.

1. יבש (1 נק׳): בהיתן שתי יוריסטיקות עקביות . האם עקבית? אם כן, הוכיחו. אם לא, הפריכו.

לא, למשל עבור אותה דוגמה מסעיף קודם מתקיים ש: וגם כלומר:

לכן קיבלנו ש: אינה עקבית.

נגדיר יוריסטיקה חדשה עבור בעיות עם מצב מטרה יחיד :

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

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

היורסטיקה אכן קבילה כי על לוח במשחק שלנו מכיוון שניתן ללכת ימינה, שמאלה, למטה, למעלה.(אין מעברי אלכסון ועלות מינימלית למעבר בין שני מצבים היא 1) אז כדי להגיע לצומת היעד צריך לעבור לפחות מספר צעדים כמו מרחק מנהטן (אולי יש מכשולים בדרך שבגללם נאריך המסלול אך לפחות נעבור משבצות כמו מרחק מנהטן) ולכן הוא גדול / שווה לאפס לפי הגדרת מרחק מנהטן. בנוסף לכן מתקייפ ש Cost(p) = 100 ולכן כאשר מרחק מנהטן קטן מ 100 מקבלים ש : ובמקרה ש: מקבלים ש:

וזה כיוון שהסברנו שצריך לפחות לעבור משבצות כמו מרחק מנהטן ו Cost(P) קטן יותר ממנו.

לסיכום, קיבלנו יוריסטיקה קבילה.

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

נחלק הבעיה לתתי מקרים:

1. נקבל

נניח בשלילה שהיוריסטיקה אינה עקבית, נקבל שקיימים כך ש:

נחלק למקרים לפי :

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. ומתקבל ש:*

*וזו כמובן סתירה להנחת השלילה ולכן היורסטיקה עקבית.*

1. *: 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. ומתקבל ש:*

*וזו כמובן סתירה להנחת השלילה ולכן היורסטיקה עקבית.*

1. *: 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. ומתקבל ש:*

*וזו כמובן סתירה להנחת השלילה ולכן היורסטיקה עקבית.*

1. *: 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. ומתקבל ש:*

*וזו כמובן סתירה להנחת השלילה ולכן היורסטיקה עקבית.*

1. *: מתקיים ש מצב v הוא משבצת מסוג P ו u היא הקצה השני של ה PORTAL ולכן:*

*אם לאחד מהם יש נקבל ש:*

*ולכן כיוון שהיוריסטקה שלנו היא אי-שלילית ההפרש ביניהם הינו קטן / שווה ל 100 כלומר קיבלנו: וזו כמובן סתירה להנחת השלילה ולכן היורסטיקה עקבית.*

*ואם לשניהם יש אז מתקבל (מארתמיטקה של שני מספרים):*

*וזו כמובן סתירה להנחת השלילה ולכן היורסטיקה עקבית.*

*ואם לשניהם יש אז מתקבל:*

*לכן, וזו כמובן סתירה להנחת השלילה ולכן היורסטיקה עקבית.*

1. *זה מקרה שלא ייתכן כיוון שזה אומר שאנחנו במצב הבא:*

*וזו כמובן סתירה להנחת השלילה ולכן היורסטיקה עקבית.*

נכליל את היוריסטיקה לבעיות עם מספר מצבי מטרה על ידי:

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

1. יבש (1 נק׳): האם היוריסטיקה קבילה על כל לוח? אם כן הסבר, אם לא הבא קודמה נגדית.
2. יבש (1 נק׳): האם היוריסטיקה עקבית על כל לוח? אם כן הסבר, אם לא הבא דוגמה נגדית.

### 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 .
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 לא תמיד מוצאים את הפתרון האופטימלי.

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

יתרון של Greedy לעומת Beam: Greedy עלול לקבל פתרון יותר טובמזה של Beam כי הוא לא זורק צמתים "מיותרים" ומשאיר את כולם גם אם הערך היורסטי שלהם גבוה.

חסרון של Greedy לעומת Beam: 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.

### 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 .
2. (יבש 2 נק׳) בהינתן , נסמן את המסלולים המחוזרים על ידי W-A\* תחת הפורמולציה ב עבור בהתאמה. אזי עבור:
   1. יוריסטיקה קבילה . אם כן הסבירו. אם לא, ספקו דוגמה נגדית.

לא נכון, למשל ניקח את היוריסטיקה אשר תקיים שהיא קבילה כי בעצם מתקיים כי כל המעברים משקליהם חיוביים ממש וחסומים מלמטה על ידי . אך מתקיים ש: לכן האלגוריתם במקרה הזה מתנהג כמו UCS וכתוצאה מכך שני המסלולים הם באותו משקל מינימלי שאלגוריתם UCS מוצא בסתירה לטענה לכן הטענה שגויה.

* 1. יוריסטיקה כללית (לא בהכרח קבילה) . אם כן הסבירו. אם לא, ספקו דוגמה נגדית.

ראינו שהטענה שגויה עבור יוריסטיקה קבילה לכן בפרט היא שגויה עבור יוריסטיקה כללית כי יוריסטיקות כלליות מכילות בתוכן את הקבילות והלא קבילות כלומר 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 .
2. יבש (2 נק׳): ספקו יתרון וחסרון של IDA\* ביחס לA\*. באילו מקרים הייתם מעדיפים להשתמש בכל אחד מהם?

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

חסרון של IDA: 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\*.

יתרון של A\*-epsilon: הוא לפעמים ימצא פתרון ותר מהר כיוון שהוא מאפשר לבחור צומת שאינו מוערך כאופטימלי.

חסרון של A\*-epsilon: הוא עלול להחזיר פתרון תת-אופטימלי בפקטור של מהפתרון של A\*..

1. 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-, מבחינת מספר פיתוחים, מסלול שנבחר ועלות המסלול שנבחר.

לחזור!

### 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.

### שאלה 13 – Local Search (5 נק׳):

בהינתן מרחב המצבים הבא, כאשר הינו המצב ההתחלתי, הינה פונקציית ערך והערך עבור כל מצב מצוין בצומת. המטרה שלנו היא למצוא מצב שממקסם את ערך .

נשתמש באלגוריתם Stochastic Hill Climbing.

כמו כן ידוע כי .

1. יבש (1 נק׳): מה ההסתברויות למעבר מהצב ההתחלתי לכל אחד מהמצבים . רשמו את .

מתקיים ש: לכן:

1. 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.

1. יבש (1 נק׳): בהיתן שבצעד הראשון האלגוריתם עבר למצב . 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 . 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 .

וגם מתקיים:

וכך קיבלנו שמתכנסים אל הפתרון האופטימלי.

1. 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 .

1. נתקעים בצומת h

האופציות לזה הן, להתקדם מ a ל c ואז ל h. שזה מתרחש בסיכוי :

1. נתקעים בצומת e.

*שהאופציות לזה הן, להתקדם מ a ל b ל e.*

שזה מתרחש בסיכוי :

ולכן בסך הכל, הסיכוי להתכנס לפתרון לא אופטימלי הוא : .

1. יבש (1 נק׳): עבור אילו ערכים של ההסתברות להגיע מהמצב ההתחלתי למקסימום הגלובלי תוך בדיוק 3 צעדים גדול מ ?

נגיע לצומת g שהוא המקסימום הגלובלי בתוך 3 צעדים בדיוק רק אם הולכים מ a ל b ל f ל g. שזה מתרחש בסיכוי נדרוש : לכן: ומתקבל : וידוע לנו מנתוני השאלה שמתקיים : לכן בסך הכל קיבלנו:

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.