**​​מבוא לבינה מלאכותית - 236501**

**תרגיל בית 1**

**מרחבי חיפוש**

**A picture containing music

Description automatically generated**

**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, 1.12.2022, בשעה 23:59.

· את המטלה יש להגיש **בזוגות בלבד**.

· יש להגיש מטלות מוקלדות בלבד. פתרונות בכתב יד לא ייבדקו.

· ניתן לשלוח שאלות בנוגע לתרגיל [**בפיאצה**](https://piazza.com/class/laarvd34xa74hm) **בלבד**.

· המתרגלת האחראית על תרגיל: **טל חקלאי**.

· 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- 10% מהציון הסופי במקצוע **ולכן העתקות תטופלנה בחומרה!**

· ציון המטלה יורכב מהגורמים הבאים:

o **60% - המסמך היבש.** מעבר לתשובות הנכונות, אתם נבחנים גם על הצגת הנתונים והתוצאות בצורה קריאה ומסודרת במקומות בהם התבקשתם לכך. הניקוד המפורט בסעיפים של מסמך זה הינו מתוך הציון היבש בלבד.

o **40% - הקוד המוגש.** הקוד שלכם ייבדק באופן מקיף ע״י טסטים. הטסטים יבדקו את התוצאות שלכם לעומת התוצאות המתקבלות במימוש שלנו. אנו מצפים שתקבלו את אותם הערכים בדיוק. נבדוק בין היתר את המסלול המתקבל, את עלותו ואת מס׳ הפיתוחים. לכן עליכם להיצמד להוראות בתרגיל זה. הבדיקות יהיו כמובן מוגבלות בזמן ריצה. יינתן לכם זמן סביר ביותר להרצת כל טסט. אם תעקבו אחר ההוראות במסמך זה ובקוד אין סיבה שלא תעמדו בזמנים אלו. בנוסף, יש להקפיד על הגשת קוד מסודר בהתאם להנחיות. יש לכתוב הערות במקומות חשובים בקוד כדי שיהיה קריא וקל לבדיקה ידנית.

· 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.

## **חלק א' – מבוא**

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

במהלך התרגיל תתבקשו להריץ מספר ניסויים ולדווח על תוצאותיהם. אתם נדרשים לבצע ניתוח של התוצאות, כפי שיוסבר בהמשך.

### 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.



## **חלק ב' – מתחילים לתכנת (40 נקודות)**

### משימה – רטוב

לפני שמתחילים בבקשה צפו בסרטון [הזה](https://drive.google.com/file/d/1Io8WCH4QQn8AqhvXSR0I63c75lspJ98E/view?usp=share_link).

כעת נעבור לחלק הרטוב של התרגיל. אנו נעבוד בסביבה שבנינו לתרגיל זה על בסיס הסביבה Frozen-Lake שפותחה ע"י OpenAI.

את החלק הרטוב אתם צריכים לפתור במחברת 236501-hw1.ipynb. אנחנו ממליצים לעבוד ב-[Google Colab](https://colab.research.google.com/drive/1wTl8nXZ5xWKmJHSJoG-I5VjhD9j9UAdk?usp=sharing). כדי לעשות זאת עליכם להעלות את תוכן התיקייה של התרגיל לתוך תיקייה ב-Drive האישי שלכם. לאחר מכן פתחו את המחברת דרך Google Colab ופעלו לפי ההוראות.

מומלץ לעבור על הקוד במחברת במקביל לקובץ הנוכחי וככה שלב שלב לענות על השאלות השונות.

## 

## **חלק ג' – שאלות יבשות על הרטוב (48 נקודות)**

התחילו לענות על חלק זה רק לאחר שהבנתם את סביבת העבודה.

### 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. מה גודל מרחב המצבים S? הסבירו.

- זהו ה-observation space של OpenAI Gym, ובמקרה שלנו הוא מייצג את כל המקומות האפשריים על הלוח שבהם הסוכן יכול להימצא.  
 – הפעולות האפשריות של הסוכן בסביבה. בסביבת האגם הקפוא הפעולות האפשריות הן .  
 – המיקום ההתחלתי של הסוכן. בסביבת האגם הקפוא זו תמיד תהיה המשבצת השמאלית ביותר העליונה.  
 – קבוצת המצבים הסופיים, ובסביבת האגם הקפוא תמיד יש מצב סופי יחיד שהוא המשבצת הימנית ביותר התחתונה במפה.

*גודל מרחב המצבים תלוי בגודל המפה והוא כמספר המשבצות שיש במפה. בשאלה הוגדר שהמפה היא ולכן*

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

הפונקציה תחזיר את כל המצבים חוץ מאלה שמתאימים למשבצות בשורה העליונה ביותר במפה.

* 1. יבש (1 נק׳): מה תחזיר לנו הפונקציה Succ על המצב ההתחלתי 0?
  2. יבש (1 נק׳): האם קיימים מעגלים במרחב החיפוש שלנו?

כן, קיימים מעגלים, כמו למשל לבצע פעולה וישר אחריה (בלי הגבלת הכלליות – נניח שניתן לבצע את רצף הפעולות האלה בלי ליפול אל בור).

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

מכל מצב ניתן לבצע עד פעולות שונות, ולכן מקדם הסיעוף הוא .

* 1. יבש (1 נק׳): במקרה הגרוע ביותר, כמה פעולות ידרשו לסוכן random (כפי שממומש במחברת) להגיע למצב הסופי?

במקרה הגרוע ביותר הוא יתקע בלולאה אינסופית ולעולם לא יגיע אל מצב סופי.

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

במקרה הטוב ביותר הסוכן יגריל פעולות שתואמות לפעולות של מסלול אופטימלי קצר ביותר (בלי להתחשב בעלויות המעברים) ולכן יגיע אל המצב הסופי בתוך 9 פעולות. (בהנחה שפעולת מעבר על צומת של פורטל מעבירה אוטומטית אל הפורטל המתאים ללא ספירת פעולה נוספת)

* 1. יבש (1 נק׳): עבור לוח כללי, כאשר המצב ההתחלתי והסופי נמצאים בקצוות הלוח (בדומה ללוח "8X8"), האם סוכן שמחפש את המסלול הקצר ביותר יחליט תמיד לעבור דרך ה-portal?

A picture containing text, hitting, clock, scoreboard

Description automatically generatedלא, יכול להיות ששני ה-portalים ממוקמים על המפה באופן כזה שלא מקצר את הדרך האופטימלית אל היעד ולכן הסוכן לא בהכרח יבחר לעבור דרך הפורטל. לדוגמה:

### שאלה 2 – BFS-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?

התנאי הוא שגרף החיפוש יהיה עץ (או, מורכב מעצים), כלומר גרף חסר מעגלים.

* 1. Diagram

     Description automatically generatedיבש (2 נק׳): עבור הלוח“4x4” שמופיע במחברת, ציירו את גרף המצבים.
  2. 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 .

הפונקציה תעבוד על גרף המצבים ותהפוך אותו לגרף מצבים חדש באופן הבא:

לכל קשת בגרף שעלותה היא , נחליף אותה במסלול הבא:

*(נניח בה"כ שהעלויות על הקשתות. אם העלות היא על הצומת, אפשר בקלות להמיר אותה בעלות על קשת כך שכל קשת שנכנסת אל הצומת מקבלת את העלות של הצומת.)*

*באופן זה החלפנו את העלויות של הקשתות במספר צמתים שיאריך את המסלול שהחליף את הקשתות במספר צמתים שמתאים לעלות שהיתה על הקשת כך שבסך הכל קיבלנו גרף שבו מספר הצמתים גדל בהתאם למשקלים שהיו על הקשתות בעבר, וכאשר נריץ עליו את אלגוריתם שמחפש את המסלול הקצר ביותר נמצא בהכרח מסלול קצר ביותר אופטימלי בגרף שמשרה על הגרף מסלול אופטימלי לפי עלות מינימלית.*

* 1. יבש (2 נק׳): נתון לוח בגודל NxN, ללא חורים, ללא Portals, המכיל N-2 משבצות רגילות (F,T,A,L) ושתי משבצות של מצב סופי ומצב התחלתי. כמה צמתים יפותחו וייווצרו במהלך חיפוש BFS-G? הסבירו?

יווצרו צמתים ומתוכם יפותחו צמתים.  
בחיפוש BFS-G עוברים על תאי הלוח בשכבות, החל מהמשבצת ההתחלתית שהיא בקיצון השמאלי למעלה כך שהצמתים שנוצרים הם באלכסונים יורדים עד שלבסוף מגיעים למשבצת האחרונה בקיצון הימני למטה שהיא משבצת המטרה. לכן, כדי להגיע אליה חייבים ליצור את כל הצמתים שמתאימים לכל משבצות הלוח, ומאחר שלא יוצרים משבצת פעמיים – יוצרים צמתים.  
בנוסף, ברגע שנפתח את הצומת הראשונה בשכבה האחרונה, בהכרח ניצור רק את צומת היעד ולפי הגדרת האלגוריתם בשלב זה האלגוריתם יסתיים ונחזיר את הפתרון. לא נפתח לא את צומת המטרה ולא את הצומת השנייה שבשכבה הפנימית ביותר (שהרי בשכבה הפנימית ביותר יש רק 2 צמתים באלכסון) ולכן יפותחו בסך הכל צמתים.

### שאלה 3 – DFS-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 אכן שלם שכן אנחנו שומרים רשימת CLOSE של מצבים שהסוכן כבר ביקר בהם, ולכן אין את הסכנה הנפוצה של אלגוריתם DFS של היתקעות במעגל.  
אמנם, האלגוריתם אינו קביל שכן אין הבטחה לאופטימליות של הפתרון. אפילו רואים אמפירית שקיבלנו פתרון קצר יותר מסוכן ה-BFS-G מאשר סוכן ה-DFS-G.

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?

האלגוריתם עלול להיתקע בלולאה אינסופית שכן כאן הוא אינו שומר קבוצת CLOSE של מצבים שהסוכן כבר ביקר בו. למעשה, בגלל שסדר פיתוח הצמתים הוא למטה, ימינה, למעלה, ואז שמאלה – אלא אם קיים פתרון טריויאלי של ללכת עד הקיר התחתון ואז ללכת בקו ישר עד הקיר הימני ששם ימצא את צומת היעד, לא יחזור פתרון שכן הסוכן יתקע בלולאה אינסופית.

1. יבש (2 נק׳): נתון לוח בגודל NxN, ללא חורים, ללא Portals, המכיל N-2 משבצות רגילות (F,T,A,L) ושתי משבצות של מצב סופי ומצב התחלתי. כמה צמתים יפותחו וייווצרו במהלך חיפוש DFS-G? הסבירו?

נשים לב שהמסלול לפתרון במקרה כזה תמיד יהיה ללכת עד הסוף למטה ואחר כך ללכת עד הסוף ימינה.  
במהלך ריצת הסוכן הוא תמיד יצור את כל הצמתים שפותחו מהצמתים שהוא מבקר בהם בדרך, ואלה כל הצמתים של השורה האחרונה ושתי העמודות הראשונות. בנוסף, הוא יפתח את כל הצמתים במסלול הפתרון פרט לצומת האחרון שכן צומת היעד לא מפותח, ולכן יפתח את הצמתים על שני צלעות הריבוע פרט לצומת היעד. בסה"כ נקבל:

מספר הצמתים שיווצרו:   
מספר הצמתים שיפותחו:

1. יבש (2 נק׳): נתון לוח בגודל NxN, ללא חורים, ללא Portals, המכיל N-2 משבצות רגילות (F,T,A,L) ושתי משבצות של מצב סופי ומצב התחלתי. כמה צמתים יפותחו וייווצרו במהלך חיפוש Backtracking-G? הסבירו?

באלגוריתם זה לא כל הצמתים נוצרים מיידית כשמפתחים צומת, אלא מבצעים פיתוח עצל שבו יוצרים את הצומת הבא בלולאה רק כאשר עומדים לטפל בו. לכן, אלגוריתם זה יתפקד בדיוק אותו הדבר כמו אלגוריתם DFS-G, פרט לכך שהוא לא יצור את כל הצמתים בדרכו, ולכן העמודה השנייה לא תווצר כלל. בסה"כ נקבל:  
מספר הצמתים שיווצרו:

מספר הצמתים שיפותחו:

### שאלה 4 – ID-DFS-G (5 נק׳):

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

1. **רטוב**: ממשו את החלקים החסרים של אלג׳ ID-DFS-G בקובץ ע״פ ההנחיות המופיעות שם.
2. יבש (1 נק'): האם האלגוריתם ID-DFS-G שלם?

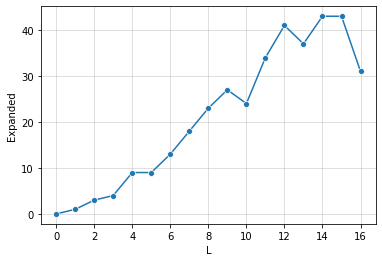
כן, אם קיים פתרון בעומק סופי הוא יימצא.

1. יבש (2 נק'): נניח כי עלות כל פעולה היא 1, האלגוריתם ID-DFS-G לא קביל. הסבירו מדוע. בנוסף, הציעו דרך לעדכן את האלגוריתם על מנת שיהיה קביל במקרה הזה.

אם נתבונן במסלול אופטימלי כלשהו, יתכן שיש צומת ביניים x שנמצא בתוך מעגל בגרף המצבים, אבל בגלל סדר בחירת הצמתים לפיתוח של הסוכן, הוא תמיד יגיע אל x במספר צעדים גדול מהמינימלי האפשרי, ובגלל שכבר ביקר בו בעבר, או שהוא עדיין נמצא ברשימת ה-OPEN שלו, הוא לא יבקר בו שוב במסלול הקצר יותר (מהחלק ה"מהיר" של המעגל) וכתוצאה מכך הפתרון האופטימלי הזה לעולם לא יחזור מריצת הסוכן על מרחב החיפוש. אם תופעה זו חוזרת בכל מסלול אופטימלי, המסלול שיחזור מריצת הסוכן לא יהיה אופטימלי ולכן האלגוריתם אינו קביל.

דוגמה נגדית:  
במפה הבאה, קיים הפתרון האופטימלי הבא מאורך 8:  
אך המסלול שהאלגוריתם מוצא לפי סדר פיתוח הצמתים שהוגדר אצלינו הוא המסלול הבא מאורך 12:  
וכמובן שזהו אינו פתרון אופטימלי.

דרך אפשרית לעדכן את האלגוריתם כדי שיהיה קביל הוא להכניס את הצמתים והמצבים לתוך רשימות וקבוצות ה-OPEN, CLOSE ביחד עם מספר הצעדים שהסוכן עבר מצומת ההתחלה עד למצב/צומת זה, ואם הסוכן מפתח צומת שאחד מילדיו הוא מצב שכבר מופיע ב-OPEN/CLOSE אך עם מרחק גדול יותר מנקודת ההתחלה מאשר המרחק הנוכחי, הסוכן יעדכן את הצומת/המצב ברשימה המתאימה ויאפשר מעבר דרכו שוב, ככה שהמסלול החדש אל הצומת שהתגלה שנית יהיה קצר יותר מהמסלול הקודם שהתגלה.

1. יבש (2 נק׳): הציגו גרף המראה את השפעת L (לפחות 5 ערכים שונים) על מספר הצמתים שמפותחים בכל העמקה. הסבירו בקצרה את הגרף.

ככל שהעומק גדל כך מספר הצמתים שיש לפתח גדל, ולכן מגמת הגרף הכללית היא עליה.

בחיפוש על עץ היה מתקבל גרף שעולה באופן מעריכי, אך מאותה הסיבה שתיארנו בסעיף הקודם חזרה על מצבים שכבר בוקרו נמנעת ולכן גם כל הבנים של מצבים אלה לא יפותחו, ולכן המון צמתים לא מפותחים בניגוד לחיפוש על עץ וכתוצאה מכך בחלק משינויי ערך ה-L ניתן לראות שלפעמים הערך Expanded לא משתנה ואף יורד.  
אמנם, ניתן לראות כי המגמה הכללית היא כן של עליה, ובעומק שבו נמצא פתרון יש ירידה משמעותית יחסית לערכים בסביבתו שכן נמצא פתרון ואין צורך להמשיך לפתח עוד צמתים עבור ריצת ה-DFS-G באותו העומק.

### שאלה 5 - 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.

האלגוריתמים יפעלו באותו האופן עבור בעיות חיפוש שבהן כל העלויות שוות (במעברים).

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

פונקציית המחיר חסומה מלמטה על ידי ולכן כפי שראינו בתרגול, האלגוריתם שלם. בנוסף, ראינו גם שבהינתן שהאלגוריתם UCS שלם, הוא גם קביל.

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.

Diagram, shape

Description automatically generatedדוגמה לגרף שבו דן **לא** יחזיר את המסלול הקל ביותר:  
המסלול שיחזור מריצת האלגוריתם יהיה , אך משקלו הוא בניגוד למסלול שמשקלו שהאלגוריתם הנכון היה מחזיר (היה מתבצע שלב עדכון למשקל כשמפתחים את ).

(המשך בעמוד הבא)

Diagram, shape

Description automatically generatedדוגמה לגרף שבו דן **כן** יחזיר את המסלול הקל ביותר:

המסלול שיחזור מריצת האלגוריתם יהיה ומשקלו יהיה וזהו גם המסלול שהאלגוריתם הנכון היה מחזיר, כי לא היה מתבצע שלב עדכון בעת פיתוח צומת .

### שאלה 6 – יוריסטיקה (2 נק׳):

נגדיר יוריסטיקה חדשה:

### 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.

כן, היוריסטיקה קבילה על כל לוח. הוכחה:

נחלק למקרים.  
: אנחנו יודעים שבלוח שבו ניתן ללכת רק בצעדי סריג, העלות של כל מסלול מכל מצב אל היעד גדולה או שווה למרחק מנהטן בין היעד למצב. כאן, העלות היא לא רק המרחק שהסוכן צועד אלא סכום המשקלים של המצבים שהוא עובר אליהם, ומאחר שאלה גדולים או שווים ל-1 (כי המסלול לא יכול לכלול חור שהעלות להיכנס אליו היא 0) הרי שסכום המשקלים הזה גדול או שווה תמיד למרחק המנהטן.  
נשים לב שעלות מעבר בפורטל במקרה זה גדולה יותר ממרחק המנהטן ולכן גם כאן הטענה לעיל מתקיימת:

*: כפי שנימקנו לעיל, מתקיים:  
ולכן לכל מצב היוריסטיקה מקיימת את הדרישות כדי להיות יוריסטיקה קבילה.*

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

A screenshot of a game

Description automatically generated with low confidenceלא, היוריסטיקה לא עקבית על כל לוח. דוגמה נגדית:

עבור שמסומנים בתמונה מתקיים כי לפי הגדרת עלות מעבר אל צומת מסוג . אמנם:

ולכן היוריסטיקה אינה עקבית.

כן, היוריסטיקה עקבית על כל לוח.  
יהי לוח, ויהי מצב כלשהו . נראה כי לכל מתקיים כי ובכך נוכיח שהיוריסטיקה עקבית. בהוכחה נתעלם ממקרים שבהם המשבצת היא חור תוך הנחה שהעלות של מעבר ממנה לכל מצב אחר היא אינסוף.

נשים לב שכל עוד גם וגם אינן משבצות Portal, השינוי שמתבצע הוא צעד אחד בכיוון או בכיוון ולכן בסה"כ מוסיפים לכל היותר למרחק מנהטן מהמצב אל מצב היעד.  
אם אחד מ- או כן משבצות פורטל, יש לחלק לשני מקרים.  
אם פורטל ו- משבצת מסוג אחר, הרי שגם כאן הטענה מתקיימת והמרחק גדל בסה"כ ב-1.  
אחרת, אם משבצת רגילה ו- היא פורטל, הרי שהעלות היא עכשיו , אבל ההפרש בין לבין הוא בהכרח קטן מ- שכן כל אחד מהם חסום מלמעלה על ידי .

*במקרה האחרון, אם גם וגם משבצות פורטל, הרי שהעלות היא , וכפי שציינו קודם, כל אחד מבין וגם חסום מלמעלה על ידי ולכן גם הפרש זה קטן מ-.*

*בסך הכל הראינו שבכל מקרה אפשרי מתקיים כי ולכן היוריסטיקה קבילה.*

### שאלה 7 – 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?

המרחב סופי (לא מאפשרים מעגלים כי מנהלים רשימת CLOSE) וקשיר, ולכן כפי שראינו בתרגול האלגוריתם הוא שלם.

בהתבססות על הלוח "8x8" שמופיע במחברת, ניתן לראות שהאלגוריתם אינו קביל שכן בהרצת אלגוריתם UCS קיבלנו מסלול עם עלות נמוכה יותר מאשר המסלול שהתקבל באלגוריתם Greedy Best First Search.

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

**חיסרון**: Greedy Best First Search לא בהכרח מחזיר מסלול אופטימלי בעוד ש-UCS כן קביל.

**יתרון**: ל-Greedy Best First Search יש "חוש ריח" לכיוון הפתרון האופטימלי ולכן לא מחפש בצורה עיוורת ולא מיודעת בכל כיוון, בעוד ש-UCS לא מיועד בכיוון פתרון היעד.

### שאלה 8 – W-A\* (7 נק׳):

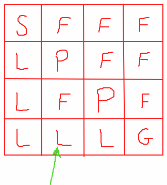
השאלות בחלק זה מתבססות על הלוח “8x8” שמופיע במחברת.

1. **רטוב**: ממשו את החלקים החסרים באלג׳ W-A\* בקובץ ע״פ ההנחיות המופיעות שם. עליכם להשתמש ביוריסטיקה .
2. יבש (4 נק׳): לפניכם מספר יוריסטיקות, הוכיחו או הפריכו בעזרת דוגמא נגדית את קבילותן של היוריסטיקות:

היוריסטיקה קבילה, שכן עבור זוג צמתים במסלול אל היעד, לכל שאינו מתאר מצב של חור, ומאחר שמסלול אל צומת יעד לא יכול לכלול מצב חור באמצע בהכרח יתקיים .

מאחר שהשאלה היא נקודתית על הלוח "8x8" שמופיע במחברת, היוריסטיקה הזו אכן קבילה שכן מרחק מנהטן מכל צומת בלוח זה אל צומת היעד הוא לכל היותר (7 צעדים מטה, ועוד 7 צעדים ימינה).  
שימוש בפורטל עולה 100 וכבר גדול יותר ממרחק המנהטן המקסימלי, ועלות מעבר אל כל צומת היא לפחות 1, ולכן סכום העלויות במסלול אל צומת היעד לא יכול להיות גדול ממרחק המנהטן אליו.

כאשר הם משבצות שיגור () ו- הוא צומת מטרה.

*מתקיים:  
ולכן היוריסטיקה קבילה.*

*היוריסטיקה לא קבילה. דוגמה נגדית:*

*הערך היוריסטי שהיוריסטיקה תחשב עבור המצב שמוצבע על ידי החיץ הירוק הוא ( למרחק המנהטן, ועוד למרחק שבין  
שני הפורטלים), ואילו העלות של המסלול עם העלות הטובה ביותר היא (המסלול האופטימלי הוא שני צעדים ימינה)*

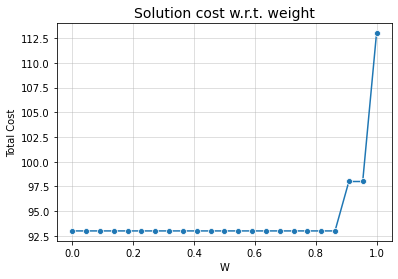
*ולכן התנאי לקבילות אינו מתקיים.*

1. יבש (נק׳ 1): איזו יוריסטיקה (מבין הקבילות) היא המיודעת ביותר (כולל היוריסטיקה )?

עלינו להשוות בין היוריסטיקות שנקרא להן בהתאם לסעיף שבו הן הופיעו - , וגם היוריסטיקה הותיקה מכיוון שרק הן קבילות.  
ראשית נשים לב שיוריסטיקה a היא החסם התחתון על כל שאר היוריסטיקות כי היא היוריסטיקה הטריוויאלית ביותר שמשמשת כאינדיקטור האם זהו מצב יעד או לא.  
עבור לוח 8x8, היוריסטיקות ו- שתיהן מתלכדות שכן לכל בלוח 8x8.  
לבסוף, ראינו כבר קודם שעבור היוריסטיקה c מתקיים כי לכל (עבור לוח ) ולכן היא פחות מיודעת מאשר ונוכל בסה"כ לומר שהיוריסטיקות וגם הן המיודעות ביותר.

1. יבש (נק׳ 2): הריצו את W-A\* עם ערכי W שונים והציגו שני גרפים:
   1. גרף 1: מספר הפיתוחים כתלות ב- W.
   2. גרף 2: עלות הפתרון שנמצא כתלות ב-W.

Chart, line chart

Description automatically generatedניתן לצייר את הגרף ביד או במחשב. הסבירו כל גרף בנפרד וגם את הקשר ביניהם.

ניתן לראות שככל ש- קרוב יותר ל- כך מספר הצמתים שהוא מפתח קטן יותר (במקרה הממוצע) אך עלות הפתרון שהאלגוריתם מוצא גבוה יותר.  
תופעה זו מוסברת מהנוסחה של :  
הצמתים יוצאים מה- לפי ערך ה- המינימלי, אך מאחר שהוא תלוי בערך סדר הוצאת הצמתים משתנה לפיו.

עבור הנוסחה היא פשוט ולכן במצב זה האלגוריתם מוציא צמתים מה- באותו הסדר כמו של , וכפי שידוע לנו זהו אלגוריתם חמדן שלא מתחשב בדרך שהוא עבר אלא רק "בחוש הריח" שלו, היוריסטיקה , בהינתן שהיוריסטיקה טובה הוא בדרך כלל יצליח למצוא מסלול אל צומת היעד תוך פיתוח פחות צמתים אך במחיר שאינו בהכרח אופטימלי שכן הוא אינו קביל.

מצד שני, עבור הנוסחה היא ולכן במקרה זה האלגוריתם מוציא צמתים מה- באותו הסדר כמו של , שידוע שהוא קביל ולכן מחזיר מסלול עם עלות מינימלית, אך במחיר של קושי חישובי גבוה – ועל כן רואים שהוא מפתח הרבה יותר צמתים.

ככל שמשנים את על הסקאלה שבין ל- מקבלים אלגוריתם שמוציא צמתים מה- שלו גם כן באסטרטגיה שנעה בין זו של לבין , כאשר עבור נקבל בדיוק את .

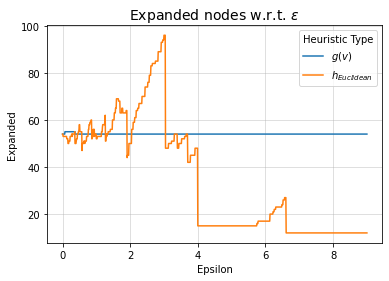
### שאלה 9 – A\*-epsilon (6 נק׳):

1. **רטוב**: ממשו את החלקים החסרים באלג׳ A\*-epsilon בקובץ ע״פ ההנחיות המופיעות שם. עליכם להשתמש ביוריסטיקה כדי ליצור את FOCAL וב- כדי לבחור את הצומת הבא לפיתוח מתוך FOCAL.
2. יבש (2 נק׳) תנו יתרון וחיסרון של A\*-epsilon לעומת A\*.

יתרון: מאפשר בחירה לא אופטימלית בתקווה למצוא פתרון מהר יותר.

חסרון: זמן חישוב יותר גדול כתוצאה מקושי החישוב של ה-Focal ובחירת המינימום שלו על ידי שלבים נוספים שאין להם צורך באלגוריתם ה-.

1. יבש/**רטוב** (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. **שימו לב- בקוד שאתם מגישים על הסוכן להשתמש ב-g(v) ולא ביוריסטיקה שבחרתם בסעיף זה.**
2. Chart

   Description automatically generatedהיוריסטיקה שאנחנו מציעים היא (מרחק אל מצב היעד).  
   להלן גרפים להשוואה מבחינת מספר פיתוחים, עלות המסלול שנבחר ואורכו:

**A screenshot of a computer

Description automatically generated with medium confidence**

## **חלק ד׳ – שאלות בסגנון מבחן (**12 **נקודות)**

1. וריאציות של A\* (6 נק׳)

1. נגדיר להיות צומת האב של צומת בגרף. כמו כן נניח ש- היא יוריסטיקה קבילה שאינה יוריסטיקת האפס וכן קיים במרחב מצב מטרה ישיג מהמצב ההתחלתי.

עבור כל אחד מהאלגוריתמים הסבר האם הוא שלם והאם הוא קביל:

1. A\* כפי שנלמד בהרצאה.
2. A\* שמתעלם מערך ה-h.
3. A\* שמתעלם מערך ה-g.
4. A\* כך ש-
5. A\* כך ש-
6. כעת נתון . נסמן להיות המסלולים המוחזרים מריצת w-A\* עם המשקלים בהתאמה. תזכורת לחישוב . הוכח/ הפרך:.
7. הוכח/הפרך: לכל בעיית חיפוש, המסלול שיחזור על ידי אלגוריתם UCS יכול להשתנות אם נוסיף ערך קבוע לכל עלות קשת במהלך החיפוש.
8. הוכח/הפרך: לכל בעיית חיפוש, המסלול שיחזור על ידי אלגוריתם UCS יכול להשתנות אם נכפיל ערך קבוע כל עלות קשת במהלך החיפוש.

2. ריק ומורטי (6 נק׳)

ריק ומורטי הלכו לאיבוד באגם הקפוא שלנו, שגודלו NxN. הם מעוניינים להיפגש, לא משנה באיזו משבצת, העיקר שעלות המסלול עד למפגש שלהם תהיה הזולה ביותר. בכל תור, כל אחד מהם מבצע את אחד מהצעדים הבאים {ימינה, שמאלה, למטה, למעלה, צעד במקום}. הניחו שכעת בלוח יש שני מצבים התחלתיים .

1. הגדירו את .
2. הגדירו את ה- לאחד האופרטורים לבחירתכם.
3. הגדירו את פונקציית ה- למצב ההתחלתי.
4. הציעו יוריסטיקה קבילה (שאינה יוריסטיקה ה-0).

## **חלק ה' – הגשת המטלה**

מעבר למימוש ולדו״ח, ציונכם מורכב גם מהגשה תקינה של המטלה לפי הכללים הבאים:

1. יש לכתוב קוד ברור:
   1. קטעי קוד מסובכים או לא קריאים יש לתעד.
   2. לתת שמות בעלי משמעות למשתנים.
2. הדוח:
   1. יש לכתוב בדוח את תעודת הזהות של שני המגישים.
   2. הדו״ח צריך להיות מוקלד במחשב ולא בכתב יש. הדוח צריך להיות בפורמט pdf.
   3. יש לשמור על סדר וקריאות בתוך הדו״ח.
   4. אלא אם נכתב אחרת יש לנמק את התשובות.
3. הגשה:
   1. יש להעלות לאתר קובץ zip בשם : AI1\_123456789\_123456789.zip (עם תעודות הזהות שלכם במקום המספרים).
   2. בתוך ה-zip צריך להיות:
      1. הדו״ח הסופי בפורמט pdf בשם AI1\_123456789\_123456789.pdf.
      2. קובץ ה-notebook בשם: AI1\_123456789\_123456789.ipynb

שימו לב: הקוד שלכם ייבדק ע״י מערכת בדיקות אוטומטיות תחת מגבלות זמני ריצה. במידה וחלק מהבדיקות יכשלו (או לא יעצרו תוך זמן סביר), הניקוד עבורן יורד באופן אוטומטי. לא תינתן הזדמנות להגשות חוזרות. אנא דאגו לעקוב באדיקות אחר הוראות ההגשה. שימו לב כי במהלך חלק מהבדיקות ייתכן שחלק מהקבצים שלכם יוחלפו במימושים שלנו. אם עקבתם אחר כל הדגשים שפורטו במסמך זה -עניין זה לא אמור להוות בעיה.

לא תתאפשרנה הגשות חוזרות, גם לא בגלל טעות טכנית קטנה ככל שתהיה. אחריותכם לוודא טרם ההגשה שהתרגיל רץ בסביבה שהגדרנו ושהקוד עומד בכל הדרישות שפירטנו.

אנא עברו בשנית על ההערות שפורסמו בתחילת מסמך זה. וודאו שאתם עומדים בהם.

שימו לב: **העתקות תטופלנה בחומרה.** אנא הימנעו מאי-נעימויות.

מקווים שתיהנו מהתרגיל!

