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׳י.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 עבור סביבת כדורי הדרקון. כאשר זה מרחב המצבים, , זה מרחב האופרטורים, , זה המצב ההתחלתי ו הוא קבוצת מצבי המטרה. מה גודל מרחב המצבים S? הסבירו.

*גודל מרחב המצבים הוא 64 כי יש 64 משבצות על הלוח בהן הסוכן יכול להיות והלוח לא משתנה.*

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

קבוצת כל המצבים מהם אפשר לעלות מעלה, .

(עבור התאים בשורה הראשונה, הפעלת UP תשאיר את הסוכן במקומו)

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

קיימים מעגלים במרחב החיפוש. ניתן לנוע בין 2 משבצות F סמוכות שוב ושוב.

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

4, קיימים 4 צעדים אפשריים אותם ניתן לעשות מכל משבצת (שאינה חור).

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

אינסוף, הסוכן יתקע במעגל ולא יגיע למצב הסופי לעולם.

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

14 פעולות. הסוכן לא יחזור אחורה, אז בסה"כ ייקח 7 צעדים ימינה ו-7 צעדים למטה.

* 1. 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.

הטענה אינה נכונה, נראה דוגמה:

|  |  |  |
| --- | --- | --- |
| **G** | **F** | **S** |
| **H** | **H** | **L** |
| **G** | **L** | **L** |

מצב המטרה הקרוב ביותר הוא 3, ומחיר ההגעה אליו הוא 11.

מצב המטרה הרחוק ביותר הוא 8, ומחיר ההגעה אליו הוא 4.

קיבלנו שמצב המטרה הקרוב ביותר הוא לא בהכרח המצב בעל המסלול הקל ביותר.

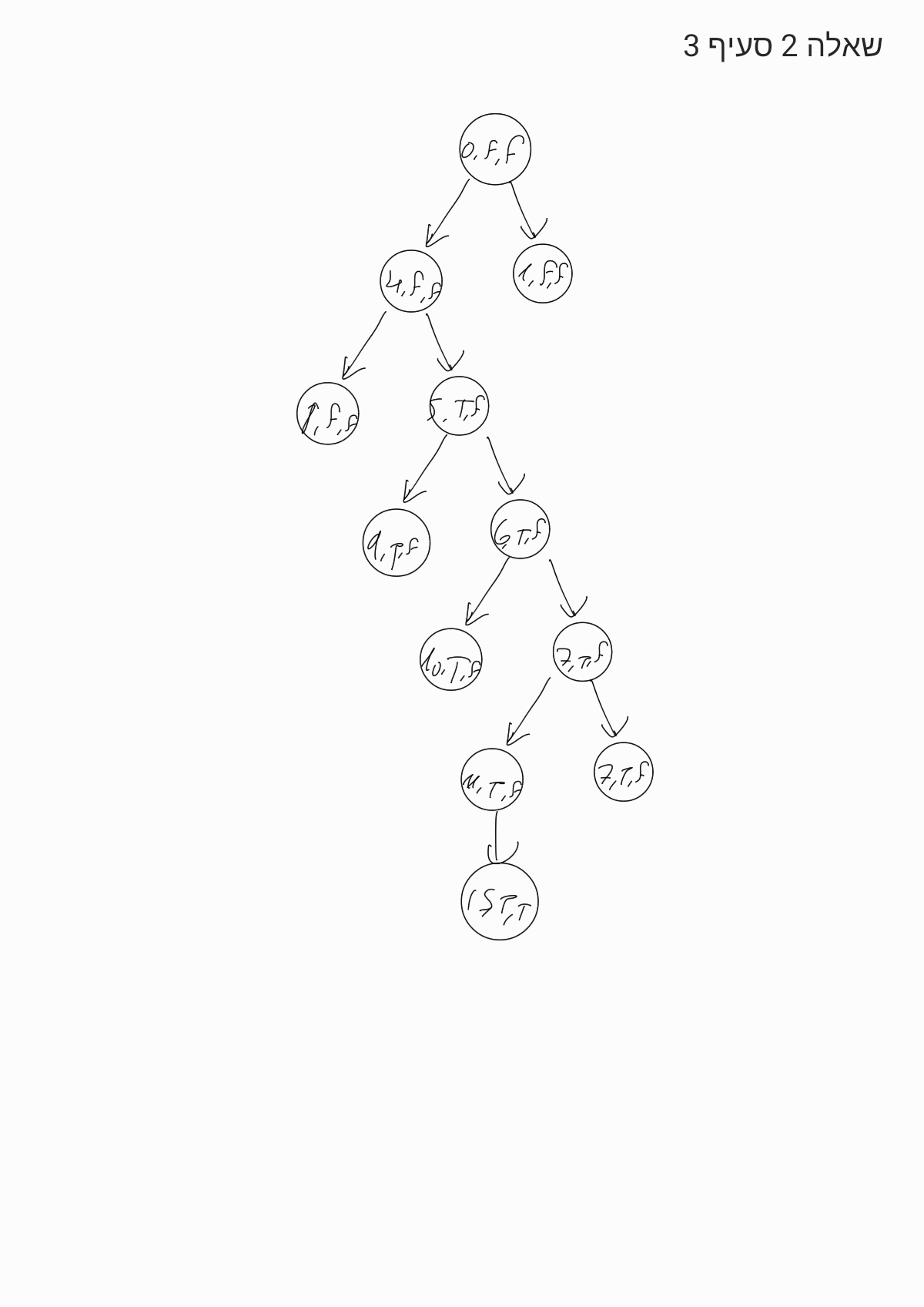
### שאלה 2 – Breadth First Search-G (7 נק׳):

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?

שלא יהיו מעגלים בגרף. אם יש מעגלים ייתכנו צמתים שיהיו במרחק שונה מצומת המקור בגרף\עץ.

* 1. יבש (2 נק׳): עבור הלוח“4x4” שמופיע במחברת, ציירו את גרף המצבים.



* 1. יבש (2 נק׳): נתון לוח בגודל NxN. 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 .

ניצור גרף חדש G’ באופן הבא:  
לכל נוסיף צמתים (שבינם קשתות) באופן הבא , כל מסלול בגרף החדש יהיה בעל מספר קשתות שווה למחירו, ומכיוון ש-BFS מוצא את המסלול הקצר ביותר, הוא יחזיר את המסלול שמקביל למסלול הקצר ביותר בגרף המקורי (ממנו ננקה את הצמתים שהוספנו ונקבל את הצמתים (לפי הסדר) בהם עבר המסלול המקורי).

דוגמה:

BFS יחזיר את המסלול שהוא המסלול הקל ביותר בגרף המקורי.

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

מצב ההתחלה ומצב המטרה נמצאים בקצוות מנוגדים של הלוח, ומצב המטרה הוא המצב הכי רחוק בלוח ממצב ההתחלה. לכן כאשר נגיע למצב המטרה בהכרח יצרנו ופיתחנו את כל שאר המצבים - מצבים.

כשנגיע למצב המטרה האלגוריתם יצור אותו אך לא יפתח אותו (כי מצאנו את מה שחיפשנו), לכן בסה"כ יפותחו צמתים ויווצרו צמתים.

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

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

DFS-G מנהל רשימה של OPEN ו-CLOSE, מה שמונע ממנו להיכנס למעגל. זהו לוח סופי אז מובטח שימצא פתרון ולכן הוא שלם.

אולם DFS-G לא קביל, הוא מתקדם לעומק בלי להתחשב במשקל התאים אז לא מובטח שימצא את הפתרון האופטימלי.

|  |  |  |
| --- | --- | --- |
| F | F | F |
| H | F | H |
| H | H | H |

1. 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. האלגוריתם יכול להיכנס למעגל אינסופי. נסתכל למשל על חלק מלוח שנראה כך:

DFS-G יתקדם מהמשבצת האמצעית בשורה העליונה אל זו שתחתיה, ומשם יעלה בחזרה לשורה העליונה ויחזור על הפעולות האלה לנצח.

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

מנקודת ההתחלה נרד ישר למטה את כל גובה הלוח (העמודה השמאלית ביותר) וניצור את כל הצמתים האפשריים מכל צומת אותה ירדנו – ניצור את כל צמתי העמודה הצמודה לעמודה השמאלית ביותר (לכל צומת, התא מעליה ומתחתיה נוצר\ייווצר במהלך הירידה ואין תא לשמאלה, אז התא היחיד שמתווסף הוא התא מימינה), אז בסה"כ במהלך הירידה ניצור צמתים ונפתח צמתים.

כעת נתקדם ימינה לכל רוחב הלוח (השורה התחתונה ביותר) וכמו במהלך הירידה ניצור את כל הצמתים האפשריים מכל צומת ממנה מתקדמים ימינה – ניצור את כל צמתי השורה שמעל השורה התחתונה ביותר (לכל צומת, התא מימינה ומשאלה נוצר במהלך ההתקדמות ואין תא מתחתיה, אז התא היחיד שמתווסף הוא התא שמעליה), בסה"כ במהלך ההתקדמות ימינה ניצור צמתים ונפתח צמתים.

בסה"כ ייווצרו צמתים ויפותחו צמתים.

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

Backtracking DFS-G יוצר רק את הצמתים שהוא מתקדם אליהם אז במהלך הדרך לא יצור את את הצמתים שהוא לא מתקדם עליהם – יצור רק את העמודה השמאלית ביותר ואת השורה התחתונה ביותר (הצמתים שייווצרו אלו הצמתים שיפותחו), אז בסה"כ הוא ייצור צמתים ויפתח צמתים.

### שאלה 4 – ID-DFS (6 נק׳):

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

ID-DFS הוא אלגוריתם שלם. נוכיח באינדוקציה על עומק הפתרון:

בסיס:

עץ בו הפתרון בעומק 1, ID-DFS יבדוק אם הפתרון בעומק 0 – יקבל לא, ימשיך לבדוק האם הפתרון בעומק 1 – ויחזיר את הפתרון.

נניח נכונות עבור מקרה בו הפתרון בעומק n ונוכיח עבור n+1:

אם הייתה צומת מטרה בעומק קטן או שווה ל-n, היינו מוצאים אותו במהלך אחת האיטרציות בו האלגוריתם מגיע לעומק ≥ n (מהנחת האינדוקציה), כיוון שלא מצאנו, נמשיך באיטרציה הבאה לעומק n+1 ושם אכן נמצא את הפתרון (כי לא נתקדם מעבר ל-n+1 לפני שנעבור על כל הצמתים בעומק הזה)

* 1. (1 נק׳) נניח כי עלות כל פעולה היא 1, האם האלגוריתם קביל? אם כן, הוכיחו. אם לא, הסבירו.

ID-DFS קביל כאשר משקל הקשתות זהה, נוכיח:

כל קשת היא בעלת משקל זהה, אז עלות הפתרון פרופורציונלית לעומקו, והפתרון בעל המחיר הנמוך ביותר הוא גם הפתרון בעל המסלול הקצר ביותר. כיוון שבכל איטרציה מגדילים את עומק החיפוש ב-1 ואנחנו עוברים על כל הצמתים בעץ עד עומק האיטרציה הנוכחית, מובטח שכשנמצא פתרון זהו הפתרון הרדוד ביותר שקיים (ולכן גם האופטימלי בעל המשקל הנמוך ביותר)

* 1. (1 נק׳) הציעו דרך לעדכן את אלגוריתם על מנת לתקן את הבעיה מהסעיף הקודם.

**//**

1. הניחו כי יש לנו ידע מקדים על חסם עליון למרחק למצב מטרה, נסמנן D. בת (Beth) הציעה את אלגוריתם החיפוש 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

1. בשאלות הבאות הניחו כי יש מספיק זמן לסיום האיטרציה הראשונה.
   1. (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.

נגדיר את שובר השוויון בין הצמתים לפי יחס הסדר הרגיל.

נתבונן במקרים בהם צומת המטרה שונה:

**ID-DFS עדיף:**

עבור , ReverseDFS יעמיק קודם כל במסלול ורק אח"כ יתקדם , בעוד ID-DFS יבדוק שאין פתרון בעומק 0 ואז ימשיך לעומק 1 ויבדוק את 1,4 וימצא את הפתרון.

**ReverseDFS עדיף:**

ReverseDFS יעמיק קודם במסלול וישר ימצא את הפתרון ב-3, בעוד ID-DFS ירד ל-1 ול-4 ורק אז ימשיך לרדת עד 3.

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

נוסיף 2 משתנים:

= חסם תחתון – אין פתרון בשכבה רדודה ממנו.

= חסם עליון – קיים פתרון בשכבה רדודה ממנו.

נאתחל את האלגוריתם כמו באלגוריתם המקורי, עם עומק L.

בכל איטרציה נעדכן את עומק החיפוש כך:

* + אם מצאנו פתרון באיטרציה האחרונה:
  + אם לא מצאנו פתרון באיטרציה האחרונה:

בכל איטרציה אנחנו תוחמים את גבולות הפתרון (ו"מתקדמים" אליו מהר יותר), האלגוריתם שלם (וקביל במקרה בו משקל כל הקשתות שוות), ויגלה את עומק הפתרון הרדוד ביותר תוך איטרציות לכל היותר (בעוד RevereseDFS ידרוש איטרציות לכל היותר)

### 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.

בעיות בהן משקל כל הקשתות זהות. UCS מעדכן את g של כל צומת לפי סכום משקל הקשתות שעל המסלול. עבור מחיר שונה – סדר החיפוש ישתנה בין גרף לגרף, עבור גרף בו משקל כל הקשתות שווה – נקבל שהצומת הרדודה ביותר היא גם הזולה ביותר ולכן גם זו שקודמת לצאת מ-OPEN – זהה ל-BFS.

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

UCS שלם. ראינו בתרגול שאם פונקציית המשקל חסומה מלמטה האלגוריתם שלם, פה משקל כל קשת הוא מינימום 1 אז האלגוריתם אכן שלם.

בנוסף, משקל כל צומת אותה מפתחים עם UCS הוא תמיד הנמוך ביותר בין שאר הצמתים ב-OPEN – UCS גם קביל.

1. 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.

יהי

נראה מקרה בו שאדי צודק ומקרה בו הוא טועה, כתלות במשקל הקשתות:

* מקרה בו יתקבל פתרון נכון:

שני האלגוריתמים יתקדמו קודם לצומת 1 כי אליו הכי זול להגיע מ-0, UCS המקורי יכניס את 3 ל-OPEN ו(אחרי שהכניס גם את 2 ל-OPEN)כשיפתח אותו (כי הוא הזול ביותר ב-OPEN) יגלה שהוא מטרה, והאלגוריתם השגוי יגלה שהוא מטרה מיד כשיכניס אותו ל-OPEN. שני האלגוריתמים יגיעו לאותה התוצאה.

* מקרה בו יתקבל פתרון שגוי:

UCS יפתח את 1 ו-2, ואז יבדוק מה יהיה המסלול הזול ביותר ל-3 (יגלה שעובר דרך 1)

האלגוריתם השגוי יפתח את 2 (כי יותר זול) ומיד יגלה ש-3 הוא מצב מטרה ויסיים (למרות שזה לא המסלול האופטימלי).

### שאלה 7 - יוריסטיקות (8 נק׳):

יהי מרחב חיפוש (G,I,O,S) , נסתכל על בעיית הניווט לכדור דרקון יחיד. . המטרה היא למצוא מסלול זול ביותר מהמוצא 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 כך שלכל מצב מתקיים .

נזכיר כי הינה פונקציית המחיר המסלול האופטימאלי מ-s לצומת היעד .

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

1. יבש (1 נק׳): מרחק מנהטן :

מרחק מנהטן היא יוריסטיקה -קבילה *כאשר החסם ההדוק ביותר.*

*היוריסטיקה חסומה מלמטה ע"י המרחק האוקלידי שיש ביניהם כביש.*

*נסמן ונקבל:*

*ולכן:*

1. יבש (1 נק׳):

היוריסטיקה היא -קבילה כאשר החסם ההדוק ביותר.

*נסמן ואז*

*כל אד מהביטויים קטן או שווה ל- ומכיוון ש- הערך המינימלי נקבל שהוא גם החסם ההדוק ביותר.*

1. יבש (1 נק׳): :

היוריסטיקה היא -קבילה כאשר החסם ההדוק ביותר.

*נסמן ונקבל:*

*כמו בסעיף 1, היוריסטיקה קבילה עם חסם הדוק .*

1. יבש (1 נק׳): נתונות יוריסטיקות שהן קבילות בהתאמה וכי הם האפסילונים ההדוקים ביותר.

הראו כי היא – קבילה , מצאו את ההדוק ביותר והוכיחו .

החסם ההדוק ביותר הוא , נוכיח:

נתון:

נגדיר ואז:

נניח בשלילה, הוא לא החסם ההדוק ביותר, אז קיים חסם הדוק ביותר, ונקבל:

בסתירה לכך ש- הוא החסם ההדוק ביותר של .

נגדיר יוריסטיקה חדשה :

* D היא קבוצת כדורי הדרקון, .

הערה: בנוסחת המרחק מתייחסים למיקום של צומת.

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.

קבילה על כל לוח. הצעדים היחידים האפשריים על הלוח הם לצדדים/למעלה/למטה ו- מחזירה את מרחק מנהטן מהיעד הקרוב ביותר. אז אם נעשה צעד לכיוון היעד, עדיין תחזיר את המרחק אל אותו היעד (לפני הצעד הוא היה הכי קרוב והצעד היה לכיוונו), ראינו בתרגול שאם ניתן לעשות רק צעדי שריג אז יוריסטיקה שמחשבת מרחק מנהטן תהיה עקבית ולכן גם קבילה.

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

עקבית, כמו שראינו בסעיף הקודם.

נגדיר יוריסטיקה חדשה :

* D היא קבוצת כדורי הדרקון, .

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

לא קבילה.

דוגמה: לוח עם 3 מצבים – [S, D, G], כאשר נהיה ב-G היוריסטיקה תחזיר 1, ואחד התנאים לכך שיוריסטיקה תהיה קבילה הם שעבור מצב מטרה יוחזר 0.

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

לא עקבית.

דוגמה: [S, D, F, F, F, G], , וכאשר נצעד בכיוון G נקבל ולא מתקיים:

### 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?

מרחב המצבים סופי אז GBFS שלם. GBFS לא קביל (יתכנו יוריסטיקות עבורן הוא קביל)

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

**יתרון:** Beam Search "זורק" חלק מהצמתים ב-OPEN אז יתכן מצב שלא יחזיר פתרון אופטימלי, בעוד GBFS מחפש במרב "רחב" יותר

**חסרון:** צריכת זיכרון גבוהה יותר**.**

### 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.

הטענה אינה נכונה.

דוגמה: היא יוריסטיקה קבילה כי

אבל .

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

הטענה אינה נכונה. ראינו בסעיף הקודם שגם עבור יוריסטיקה קבילה הטענה אינה נכונה.

### 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 של IDA\*: סיבוכיות מקום נמוכה יותר, אלגוריתם עדיף בבעיות עם מרחב מצבים גדול שדורשות המון זיכרון.

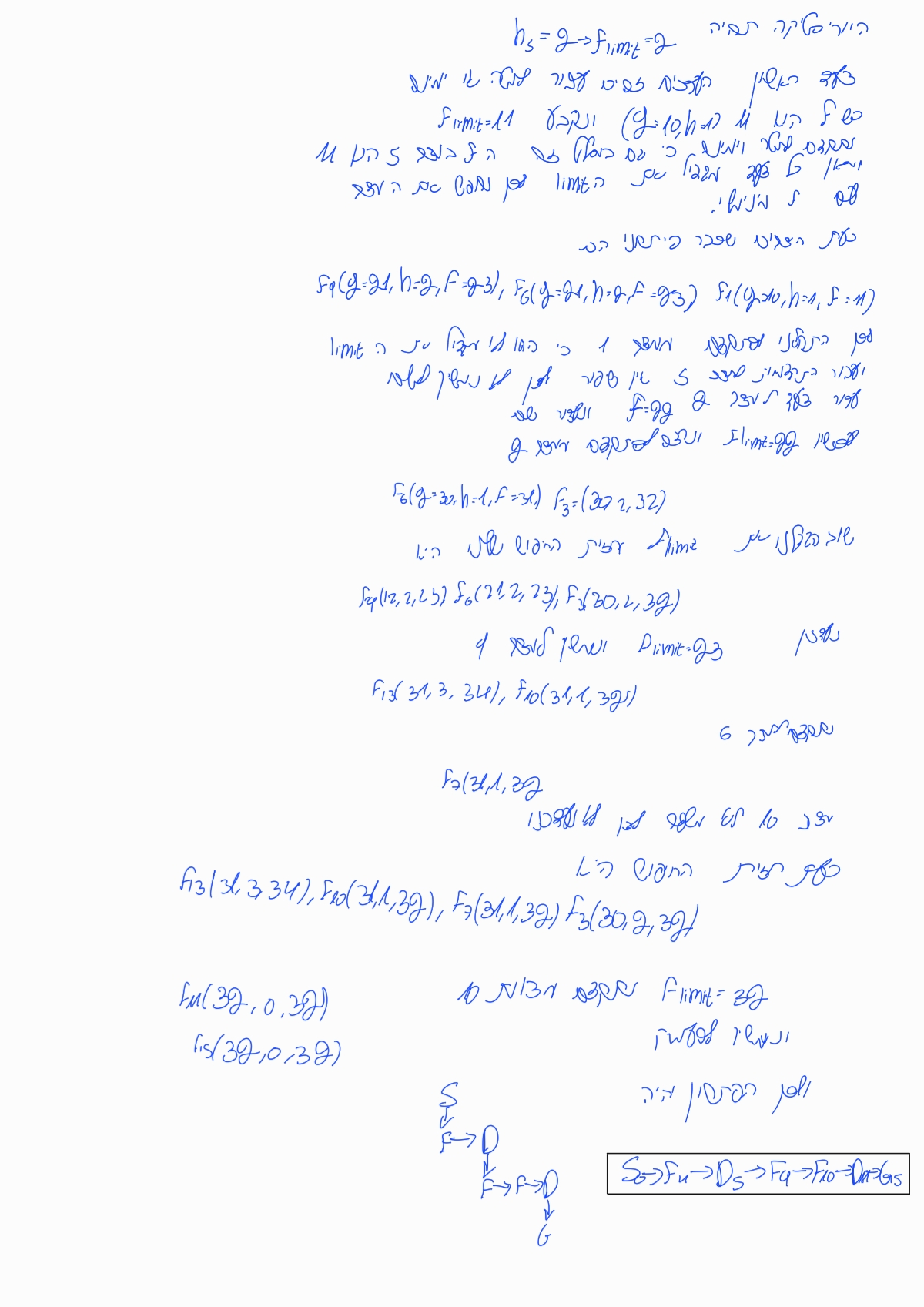
יתרון של A\*: מפתח מצב שוב רק אם זה תורם לאיכות הפתרון, עדיף בבעיות עם מרחב מצבים קטן והזיכרון לא מהווה פקטור.

1. יבש (1 נק׳): ספק המחשה שלב אחר שלב של אלגוריתם IDA\* על הלוח (4x4) שמופיע במחברת, המראה כיצד החיפוש מתקדם באמצעות העמקה איטרטיבית?

הלוח המופיע במחברת: [[S,F,F,F],[F,D,F,F],[F,F,F,D],[F,F,F,G]]

נאתחל

היוריסטיקה תהיה hs = 2 -> flimit = 2



### שאלה 11 – A\* epsilon (6 נק׳):

1. **רטוב**: ממשו את החלקים החסרים באלג׳ W-A\* בקובץ ע״פ ההנחיות המופיעות שם. עליכם להשתמש ביוריסטיקה \
2. יבש (2 נק׳): תנו יתרון וחיסרון של A\*-epsilon לעומת A\*.

יתרון של A\* epsilon: מוצא את הפתרון (בגבול השגיאה) מהר יותר.

יתרון של A\*: מוצא פתרון אופטימלי.

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

נשתמש ביוריסטיקה שמחשבת את המרחק מצומת המטרה.

המסלול שנקבל עם היוריסטיקה הזו זהה למסלול שמתקבל מהשימוש ב- אבל נפתח פחות צמתים במהלך החיפוש כי האלגוריתם "יתחשב" יותר ביוריסטיקה מאשר במשקל המסלול (יותר מיודע).

1. יבש (1נק׳): אם נגדיר שאפסילון שווה לאינסוף איך תהיה ההתנהגות של האלגוריתם עם סביבת כדורי הדרקון.

האלגוריתם יכניס גם את הבורות ל-FOCAL ומכיוון שערך g שלהם גבוה הם לא יבחרו במהלך הריצה.

### 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.

BFS החזיר את המספרים הגרועים ביותר (משקל המסלול ומספר פיתוחים), מסתדר עם זה שהוא אלגוריתם לא מיודע ויורה לכל הכיוונים.

ראינו בתרגול שעבור אלגוריתם WA\* יש קשר הפוך בין איכות הפתרון למהירות בה מצאנו אותו ורואים שזה אכן מה שקרה. ככל שהגדלנו את W פיתחנו פחות צמתים (פתרון מהיר יותר אבל פחות אופטימלי). (היוצא מן הכלל הוא הלוח 20\*20 בו יש המון בורות אז האלגוריתם תיקן את המסלול הרבה פעמים כדי לעקוף אותם)

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

מקרים של 3 צעדים:

מקרים של 2 צעדים:

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

לא. האלגוריתם לא מתקדם בלי לשפר ממש את המצב הנוכחי אז מ-C הוא יתקדם רק ל-H שהוא מקסימום לוקאלי ושם יתקע בלי להגיע למקסימום גלובלי.

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

יש שני מקרים:

1. אם הצעד הראשון הוא ל-C (כמו בסעיף הקודם), ההסתברות היא 2/5.
2. אם הצעד הראשון הוא ל-D גם נתקע במקסימום לוקאלי, ההסתברות היא 1/5.

לכן בסה"כ נקבל שההסתברות להתכנס לפתרון לא אופטימלי היא 3/5.

(אם הצעד הראשון יהיה ל-B האלגוריתם ימשיך ל-F (ואז או שהוא עצמו פתרון אופטימלי או שנמשיך ממנו -G) או G ובכל מקרה נגיע למקסימום גלובלי)

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

פתרון אופטימלי תוך 3 צעדים הוא המסלול .

ההסתברות למסלול הזה: