דו"ח - מטלה 1 אינטליגנציה מלאכותית

מגיש: בני כץ 204031553

הסבר על מימוש האלגוריתם:

הקדמה : עבור שני האלגוריתמים max\_depth=31

- מצב התחלתי: רשימה פתוחה עם קודקוד התחלתי בתוכה, רשימה סגור הריקה.

אם הקודקוד הנוכחי גם הקודקוד הסופי- מצאנו פתרון. עצור והכרז פתרון.

אחרת:

כל עוד הרשימה הריקה אינה ריקה:

1. נוציא את הקודקוד עם ה-f הכי נמוך- נקרא לו קודקוד נוכחי.
2. ניצור את כל הילדים האפשריים:

עבור כל ילד:

1. אם h=0 מצאנו פתרון אופטימלי, מרוקנים את הרשימה הריקה ומכריזים פתרון.
2. אם הילד כבר נמצא ברשימה הריקה או הסגורה המשך לילד הבא.
3. אם h>max\_depth סימן שהילד לא יוביל אותנו לפתרון ועלינו להכניס אותו לרשימה הסגורה.
4. אחרת הכנס את הילד לרשימה הפתוחה
5. הכנס את הקודקוד הנוכחי לרשימה הסגורה
6. חזור לשלב 1.

אם הרשימה ריקה ואין פתרון הכרז אין פתרון. אם יש פתרון הכרז יש פתרון.

– מצב התחלתי: קודקוד התחלתי הוא הקודקוד הנוכחי, גבול עליון הוא אינסוף.

1.אם הקודקוד הנוכחי הוא הקודקוד הסופי עצור הכרז פתרון.

אחרת:

כל עוד מצב נוכחי אינו Null:

1. אם הקודקוד הנוכחי הוא הקודקוד הסופי מצאנו פתרון:
2. אם הפתרון שמצאנו טוב מהפתרון הנוכחי עדכן את הפתרון הנוכחי ואת הגבול העליון

הרוג את הקודקוד הנוכחי, הכנס את הקודקוד לרשימת המתים, הגדר את האבא של הקודקוד כקודקוד נוכחי וחזור לסעיף 1.

1. אם אין לקודקוד ילדים:
2. אם הגבול התחתון של הקודקוד הנוכחי גרוע מהגבול הנוכחי או max\_depth הרוג את הקודקוד והגדר את האבא כקודקוד נוכחי. חזור לסעיף 1
3. אחרת, צור ילדים ותכריז על הילד הכי מבטיח כקודקוד נוכחי. חזור לסעיף 1.
4. אם יש לקודקוד ילדים בחר את הילד החי הכי מבטיח. אם אין ילדים חיים, הרוג את הקודקוד הנוכחי והגדר את האבא כקודקוד נוכחי.
5. אחרת: הגדר את הקודקוד הנוכחי מת, הגדר את האבא כקודקוד נוכחי וחזור לסעיף 1.

אם המצב הנוכחי Null ואין פתרון הכרז אין פתרון. אם יש פתרון הכרז יש פתרון.

יוריסטיקות: 1) משבצות לא מקום- מספר המשבצות שמיקומן שונה ממיקומן הסופי

2) מרחק מנהטן – סך מרחקי המשבצות של כל אחת ממרחקן הסופי(מרחק X+ מרחקY)

3) מרחק מנהטן+החלפות – מרחק מנהטן + כמו המשבצות השכנות המשוחלפות ביחס למיקומן הסופי.

קשיים: הקושי העיקרי מבחינתי היה להגביל את זמני הריצה. בהתחלה זמני הריצה היו ארוכים מאד ולא היה נראה לי הגיוני שלבעיה כזו יש כל כך הרבה זמן ריצה. הבנתי שניתן לתחום את עומק העצים ב31- אורך הפתרון הארוך ביותר.

לכן, אם לא נמצא פתרון ב31 הצעדים הראשונים כבר לא יהיה פתרון. זה עזר להרוג קודקודים עם f>31 מכיוון שהם לא יובילו לפתרון. זה חתך באופן משמעותי את כמות הצעדים המקסימלית ואת זמן הריצה .

– נאמר שהגענו למצב סופי והפתרון שלנו הוא . נחזור לשלב ה- במהלך חוקי. כך נבצע במשך צעדים. אם הגענו למצב ההתחלתי הרי שיש לנו פתרון תקין.

– אם קיים פתרון הוא קיים ב-31 השלבים הראשונים. גם אם היוריסטיקה שלנו היא נוראית כזאת הנותנת ערך שווה לכל ילד של קודקוד ונצטרך לסרוק את כל העץ ומאחר ויש לכל היותר 9! מצבים. מאחר וכמות המצבים היא סופית, אם יש פתרון הוא ימצא לכל היותר בקודקוד ה9!

– האלגוריתם מסתיים כאשר הרשימה הפתוחה מתרוקנת או מוצאים פתרון. אם מצאנו פתרון נרוקן את הרשימה וסיימנו את האלגוריתם. אם לא נמצא פתרון, לא ניתן להכניס יותר מ-9! קודקודים לרשימה הפתוחה. לכן כאשר כל הרשימה הפתוחה תתרוקן האלגוריתם יסיים ויוכרז שאין פתרון.

– נאמר ש הוא הפתרון האופטימלי אם:

*נניח בשלילה שקיים שהוא פתרון עדיף על .*

נניח שאנו בשלב ה- ובשלב הבא נמצא הפתרון. בחזית נמצאים קודקוד - השייך ל- וקודקוד – השייך ל-.

*מכיוון שנבחר להרחיב את ולא הרי ש- ..*

*מכיוון ש במצב סופי הרי ש- לכן . מכיוון ש- אדמיסבילית .*

*עבור כל קודקוד המרחיב את וכך*

*לכן, .*

– נאמר שהגענו למצב סופי והפתרון שלנו הוא . נחזור לשלב ה- במהלך חוקי. כך נבצע במשך צעדים. אם הגענו למצב ההתחלתי הרי שיש לנו פתרון תקין.

– אם קיים פתרון הוא קיים . גם אם היוריסטיקה שלנו היא נוראית כזאת הנותנת ערך שווה לכל ילד של קודקוד ונצטרך לסרוק את כל העץ ומאחר ויש לכל היותר 9! מצבים. מאחר וכמות המצבים היא סופית, אם יש פתרון הוא ימצא לכל היותר בקודקוד ה9!.

– האלגוריתם מסתיים כאשר המצב הנוכחי הופך ל-Null.האלגוריתם מוגבל לעומק 31. לכן יש לו כמות מוגבלת של קודקודים לסרוק. הוא לא יכול לפתוח קודקוד פעמיים כי הוא כבר יהיה ב-Dead\_list לכן לא יהיו מעגלים. כמות הקודקודים סופית וכאשר המצב הנוכחי יהפוך ל-Null – עם פתרון או בלי.

– נאמר מצאנו פתרון האופטימלי בשני תרחישים:

*מצאנו את -פתרון כלשהו. כעת מצאנו את .*

1. *אם עדיף על או טוב כמו - נשאיר את פתרון אופטימלי .*
2. *אם נחות מ- נגדיר את כפתרון אופטימלי חדש.*

*בסוף הריצה נחזיר את הפתרון הטוב ביותר שמצאנו.*

*לצורך השוואה בין השיטות השונות בניתי פונקציה המגרילה 50 מטריצות פתירות ומריצה אותן על האלגוריתמים.*

*בסוף ריצה של אלגוריתם הוא מחזיר מידע המכיל גם כמות איטרציות – כמות הפעמים שקודקוד יצר ילדים.*

*סידרתי מהמצב ההתחלתי הקל לקשה לפי כמות איטרציות ממוצעת בין האלגוריתמים השונים וכך יצרתי אינדיקציה לפתרון. מצורף קובץ results.csv עם התוצאות של האלגוריתמים.*

*אציין כי B&B מאד גרוע כאשר ההיוריסטיקה מאד גרועה והיוריסטיקה טובה משפרת אותו פלאים.*

*בנוסף, כפי שגיא שני ציין- נרצה יוריסטיקה כמה שיותר הדוקה מלמטה לכן* מרחק מנהטן+החלפות היא ההיוריסטיקה העדיפה ביותר.

*אציין שמצורפים גם גרפים להשוואה בין יוריסטיקות, והשוואה בין יוריסטיקות ואלגוריתמים.*

