

בינה מלאכותית – מטלה 1 דו"ח

1.1) סדר החיפוש הוא מה שהייתי מצפה שכן בתחילה הוא מחפש במצבים הקרובים אליו ביותר ולאחר שהוא חוקר אותם הוא עובר לחפש במצבים רחוקים יותר, ובנוסף הוא מפתח רק מצבים שנמצאים במסלולים שנראים לו בתחילה הגיוניים שיובילו למצב המטרה. בפועל הפאקמן לא עובר בכל המצבים שנחקרו אלא רק באלה שייתנו לנו את המסלול הקצר ביותר אל נקודת הסיום.

1.2) זהו אינו פתרון בעלות הנמוכה ביותר. אלגוריתם DFS עובד בצורה כזאת שאנו בכל שלב יורדים לעומק העץ ולכן מה שעלול לקרות זה שנבחר בטעות לרדת בעץ באזור שלא נמצא בו הפתרון ולכן זה עלול ליצור לנו פתרון שאינו אופטימלי לבעיה שלנו.

4.1) בהרצה של openMaze עם האלגוריתמים השונים מתקבל:
DFS – עלות מסלול של 158, מספר הצמתים שפותחו הוא 404.

BFS – עלות מסלול של 54, מספר הצמתים שפותחו הוא 682.

UCS – עלות מסלול של 54, מספר הצמתים שפותחו הוא 682.

A* - עלות מסלול של 54, מספר הצמתים שפותחו הוא 535.

5.1) בחרתי לייצג את מצב ההתחלה של הפאקמן כטאפל המכיל באיבר הראשון את המיקום ההתחלתי של האפקמן ובאיבר השני נמצאות הפינות של הלוח.

לאחר מכן, בחרתי להגדיר את המצבים האפשריים עבור כל מצב כטאפל, כך שהאיבר הראשון מכיל את הנקודה הבאה אליה ניתן להגיע לפי הכללים והאיבר השני הוא טאפל של כל הפינות הנותרות ביחס למצב ממנו הגעתי.

בנוסף הגדרתי, שאם מדובר בפינה אז היא יורדת מרשימת הפינות הנותרות. לבסוף, הגדרתי מצב מטרה ככזה שמספר הפינות הקיימות במצב הנוכחי הוא 0.

6.1) הפונקציה היוריסטית שבחרתי מחשבת עבור כל קודקוד בעזרת מרחק מנהטן את המרחק מהפינה הקרובה ביותר, ואז אני כביכול הולך אליה מוריד אותה מרשימת הפינות הנותרות ואז אני מחשב בעזרת מרחק מנהטן את המרחק אל הפינה הקרובה ביותר וכן הלאה עבור כל הפינות, סוכם את המרחקים האלה ולבסוף מחזיר את ערך הסכום.

פונקציה זו מקיימת את הדרישות שכן המטרה שלי היא לבקר בכל הפינות, לכן המרחק אל הפינה הקרובה ביותר לפי מרחקי מנהטן יהיה קטן/שווה למרחק האמיתי אל הפינה הזאת וכן ערך זה הוא חיובי בהכרח, ואז הפינה הבאה הקרובה ביותר היא על בקו ישר מהפינה הנוכחית וכיוון שבמשחק האמיתי ישנם קירות שיכולים לגרום לי בפועל ללכת יותר מרחק שלי בעזרת מרחק מנהטן יהיה גם כאן קטן/שווה למרחק האמיתי, וכן

הסכום של המרחקים האלה בהכרח יהיה חיובי וקטן/שווה לסכום המרחקים האמיתיים שפאקמן יצטרך ללכת בפועל.

7.1) הפונקציה היוריסטית שבחרתי מחפשת את נקודת האוכל הרחוקה ביותר מהמצב ההתחלתי בעזרת הפונקציה `mazeDistance` שנותנת לי את המרחק האמיתי בהתחשב בקירות שיש בדרך, ומרחק זה הוא הערך שהפונקציה שלי מחזירה. מרחק זה תמיד יהיה קטן/שווה מהמרחק האמיתי מכיוון שאני צריך לעבור דרך שאר נקודות האוכל, ויש שתי אפשרויות:

1. שאר הנקודות נמצאות במסלול שהתקבל אל הנקודה האחרונה – במצב זה אורך המסלול האמיתי יהיה שווה לזה שיתקבל מהפונקציה.
2. שאר הנקודות לא נמצאות במסלול שהתקבל אל הנקודה האחרונה – במצב הזה מסלול ההליכה בפועל רק יגדל.