**דו"ח מסכם – ניסוי 1**

**מעבדה בבינה מלאכותית**

מגישים: רועי זהר 209896174 רועי מזן 319042800

**מבנה הפתרון:**

צירפנו למסמך זה ארבעה מסמכים נוספים:

1. Final.py – הSOURCE CODE עבור הפתרון. התוכנה נכתבה בפייתון ולכן לא יכלנו לספק קובץ הרצה EXE. הרצת התוכנית מתבצע בעזרת שורת הפקודה python3 final.py.
2. Ans\_with\_cost\_1.txt הפלט של התוכנית עבור הפונקציה ההיורסטית הראשונה שהצענו + סטטיסטיקות.
3. Ans\_with\_cost\_2.txt הפלט של התוכנית עבור הפונקציה ההיורסטית השנייה שהצענו + סטטיסטיקות.
4. Ans.txt – הפתרונות של הפונקציה ההיוריסטית השנייה בפורמט שהתבקשנו.

**הצגת הפתרון:**

כפי שנראה בכיתה, מימשנו חיפוש מיודע בעץ בעזרת שתי פונקציות היורסיטיות שהצענו. הפונקציה הראשונה, מתארת את מספר המכוניות החוסמות את המכונית האדומה מהיציאה, למשל עבור הדוגמא הבאה:



הפונקציה הראשונה תחזיר 1, משום שרק המשאית הצהובה נמצאת בדרכה של האדומה. פונקציה זו היא אדמיסבילית, משום שאכן נידרש לבצע מהלכים כמספר המכוניות העומדות בדרכה של המכונית האדומה. פונקציה זו היא גם מונוטונית, משום שבכל מהלך אנחנו מקטינים את הפונקציה ההיוריסטית ב1 לכל היותר (מכונית זזה מדרכה של האדומה), וזה אכן המחיר של ההזזה.

באופן דומה, הפונקציה השנייה שהצענו היא למעשה סכום אורכי המכוניות מהפונקציה הראשונה. פונקציה זו הרגישה לנו כמאין הרחבה לפונקציה הראשונה, משום שהיא מתארת להערכתנו את גודל החסימה שיש על המכונית האדומה בצורה טובה יותר. עם זאת, פונקציה זו אינה אדמיסבילית, משום שלמשל במקרה לעיל, הפונקציה תחזיר את הערך 3, למרות שבמהלך אחד ניתן לסיים את השלב. מאותה דוגמא נובע גם שפונקציה זו אינה מונוטונית.

המימוש התבצע בעזרת תור ממויין הכבר קיים בפייתון, בשם PriorityQueue. מתוך הדוקומנטציה על מבנה הנתונים הנ"ל, ראינו כי הוא ממומש בעזרת heapq, דהיינו ערימה בינארית שעברה אופטימיזציה בפייתון.

למרות האופטימזציות של פייתון, ייתכן שערימת פיבונאצ'י הייתה עובדת בצורה טובה יותר כאן, משום שבניגוד לערימה בינארית, זמן ההכנסה לשיעורין בערימת פיבונאצ'י הוא קבוע לעומת לוגריתמי בערימה בינארית.

לאחר התבוננות בקבצי הסטטיסטיקות, ראינו כי את התרגילים הEASY התוכנה פותרת בזמנים שהם מתחת ל0.1 כמעט תמיד, שזהו זמן מהיר מאוד. דווקא את תרגיל 14 התוכנה לא הצליחה לפתור בזמן שהקצבנו לה, למרות שהוא לא מסומן כתרגיל קשה. ראינו שלוקח לה להריץ את התרגיל בתוך כ5-7 שניות, ולכן כנראה שהפונקציה ההיוריסטית לא נתנ הבתרגיל הזה אינדיקציה טובה לגבי מרחב החיפוש.

ראינו גם בתרגילים INTERMEDIATE, HARD שהזמן חיפוש עלה, עומקי העצים גדלו, וגם אורכי הפתרונות גדלו. בהשוואה לפתרונות שמצאנו בחיפושים לא מיועדים, אשר מניבים פתרון באורך אופטימלי, ראינו הבדלים באורכי הפתרונות.

סך הכל, שתי הפונקציות ההיורסטיות הצליחו את כל התרגילים במגבלת זמן של 35 שניות. בחרנו להציג ריצות של התוכנה בכ25 שניות לכל פונקציה, כך שהן נכשלות במבחן 14 שהיווה להם אתגר מסויים. את פירוט התוצאות עבור כל אחד מהאלגוריתמים ניתן לראות בתחתית הקבצים הרלוונטיים.