תרגיל בית 1 - מבוא לבינה מלאכותית 236501

בני נזימוב 314862129
ליעד ארם 315695783

.1

$$k! \cdot (m+1)^k \cdot m$$

כאשר $k!$ זה מתן סדר לדירות, $(m+1)^k$ מספר האופציות לביקור\חוסר ביקור במעבדות בין כל שתי דירות ולפני הדירה הראשונה, ו-m זה מספר האופציות לסיים במעבדה.

.2

.3

K	M	#possiblePaths	Estimated calculation time
7	2	22.04×10^6	$18.47[s]$
7	3	24.77×10^7	$3.8[mins]$
8	3	79.27×10^8	$2.3[hours]$
8	4	63×10^9	$19.6[hours]$
9	3	28.54×10^{10}	$3.7[days]$
10	3	11.42×10^{12}	$5.3[months]$
11	3	50.27×10^{13}	$20.8[years]$
12	3	24.11×10^{15}	$1.1[thousand\ years]$
12	4	46.78×10^{16}	$22.1[thousand\ years]$
13	4	30.41×10^{18}	$1.5[million\ years]$

.4

ערך הקיצון המקסימלי הוא $k+m$ והוא יתקבל במצב בו לא ביקרנו באף אחת מהמעבדות ומספר המטושים באמבולנס קטן ממספר המטושים המקסימלי, וגם לא ביקרנו באף אחת מהדירות ומספר המטושים באמבולנס גדול מספיק כדי שנוכל לבקר בכל אחת מהן.
ערך הקיצון המינימלי הוא 0, והוא יתקבל במצב בו ביקרנו בכל הדירות, וביקרנו בכל המעבדות, וכעת אנו נמצאים במעבדה האחרונה שלה העברנו את כל המטושים שנשארו באמבולנס.

.5

לא ייתכנו מעגלים במרחב המצבים. מצב שבו ה$cur.loc$ מתאר דירה, אינו יכול להיות חלק ממעגל, כיוון שלא ניתן לבקר בדירה פעמיים. כלומר, מעגלים יכולים להתקבל רק ממצבים בהם המיקום $(cur.loc)$ מתאר מעבדות. נשים לב כי ביקור במעבדה מרוקן את קבוצת הבדיקות באמבולנס $(Taken)$ ולכן לא יתכן שנחזור למעבדה שבה מסרנו את הבדיקות. נשים לב כי בסיטואציה של רצף ביקורים במעבדות שבכולם ה$Taken$ הוא ריק, המצבים עדיין נבדלים במספר המטושים הזמינים באמבולנס, שכן אם ביקרנו במעבדה כאשר ה$Taken$ ריק מספר המטושים שלנו בהכרח יגדל ולכן מדובר על מצבים שונים. \\\\\\\\\\\\\\\\\\\\\\\\\\דרוש שינוי ניסוח?

.6

מרחב המצבים כפי שהוגדר הוא אינסופי, כיוון שאחד האיברים בחמישיה שמגדירה מצב הוא מספר מטושים באמבולנס ($Matoshim$), ומספר זה מוגדר להיות מספר טבעי כלשהו. בפועל, לא כל המצבים ישיגים, למשל מצבים בהם הערך $Matoshim$ גדול מסכום כל המטושים בכל המעבדות + מספר המטושים ההתחלתי, אינם ישיגים. זאת מכיוון שלקחנו את כל המטושים הזמינים במעבדות ואין לנו מאיפה לקבל עוד מטושים.

7.

כן. לדוגמא מצב $s \in S$ כך ש- $s.visitiedLabs = Labs$ (כלומר ביקרנו בכל המעבדות ולקחנו את כל המטושים הזמינים) אך $s.Matoshim < d_i.roomates$ לכל i עבורו לא ביקרנו בדירה ה-d_i.

8.

אורך המסלול המינימלי הוא $k+1$ קשתות, יתקבל כאשר $initialNrMatoshimAmb \geq \sum_{i=1}^{k} d_i.roomates$, וגם $ambulanceTestCapacity \geq \sum_{i\in[k]} d_i.roomates$, לכן נעבור על כל הדירות (k קשתות) ובסוף נסיים במעבדה כלשהי. סך הכל $k+1$ קשתות.
אורך המסלול המקסימלי (שמסתיים במצב סופי) מתקבל כאשר נבקר תחילה בכל המעבדות וניקח את כל המטושים הזמינים (m קשתות), ולאחר מכן נבקר במעבדה כלשהי פעם נוספת לאחר ביקור בכל דירה ($2k$ קשתות) כדי לשים את הבדיקות שאספנו מהדירה במעבדה. סך הכל נקבל מסלול באורך $2k+m$.
סך הכל הטווח הוא בין $k+1$ ל-$2k+m$.

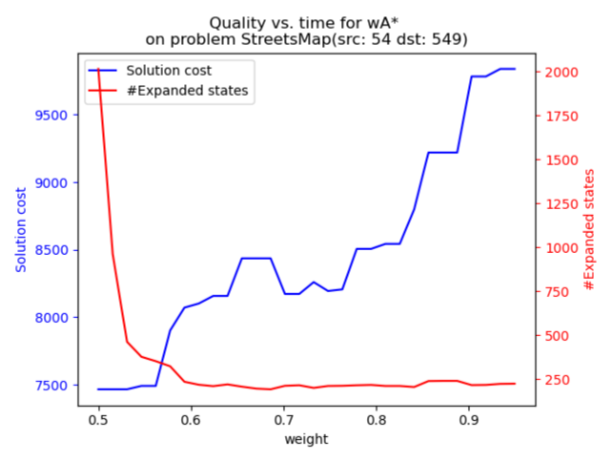
9.

$$
\begin{aligned}
Succ_{MDA}(s)=&\{(l_i,\emptyset,s.Taken\cup s.Transferred,s.Matoshim+l_i.matoshim,\{l_i\}\cup s.VisitedLabs) \mid i\in[m]\wedge s.Taken\neq\emptyset)\}\\
&\cup\{(l_i,\emptyset,s.Transferred,s.Matoshim+l_i.matoshim,\{l_i\}\cup s.VisitedLabs) \mid i\in[m]\wedge l_i\notin s.VisitedLabs\wedge s.Taken=\emptyset)\}\\
&\cup\{(d_i,\{d_i\}\cup s.Taken,s.Transferred,s.Matoshim-d_i.roomates,s.VisitedLabs) \mid\\
&\quad i\in[k]\wedge d_i\notin s.Taken\cup S.Transferred\wedge d_i.roomates\leq s.matoshim\wedge d_i.roomates\leq ambulanceTestCapcitiy-\textstyle\sum_{d\in s.Taken} d.roomates)\}
\end{aligned}
$$

14.

$$\#dev\_saved\_percentage = \frac{\#dev\_blind - \#dev\_heuristic}{\#dev\_blind} = \frac{17354 - 2015}{17354} \cdot 100\% = 88.3888\%$$

16.



הגרף האדום מתאר את מספר הפיתוחים, וככל שיש יותר פיתוחים זמן הריצה גבוה יותר. הגרף הכחול מתאר את איכות הפתרון. ניתן לראות כי ככל שנותנים לפונקציה היוריסטית משקל גבוה יותר כך זמן הריצה למציאת פתרון קטן, אך הפתרון פחות איכותי (יותר יקר). האזור הכדאי על פי הגרף הוא (בערך) $0.53 \le w \le 0.57$ מכיוון שבאזור יש ירידה חדה בזמן הריצה אך איכות הפתרון לא מתרחקת בהרבה מהפתרון האופטימלי.
נסתכל על $w_1 \approx 0.67$ ו-$w_2 \approx 0.7$. מתקיים כי $w_2 > w_1$ אך איכות הפתרון שמתקבלת עבור w_1 נמוכה יותר מאיכות הפתרון עבור w_2, כפי שנאמר בדגש.
כמו כן, נסתכל על $w_3 \approx 0.8$ ו-$w_4 \approx 0.87$, מתקיים כי $w_4 > w_3$ אך מספר הפיתוחים שמתקבל עבור w_3 נמוך יותר ממספר הפיתוחים עבור w_4, כפי שנאמר בדגש.

.19

החסרון של הגישה מבחינת יעילות הפתרון היה מתבטא בחוסר שימוש ב-$cache$. במימוש הנוכחי בבעיית המדא כאשר אנו פותרים את בעיית המפה כ"בעיית ביניים" יתכן כי הפתרון כבר נמצא ב-$cache$ ונוכל להביאו ולחסוך את עלות החישוב. אם היינו במרחב משולב שהוצע בשאלה, לא היינו יכולים להשתמש ב-$cache$ מכיוון שכל פעם היינו פותרים בעיה אחרת.

.20

.i

```
@dataclass(frozen=True)
```

.ii

השורה הזאת אינה מספיקה, שכן אם אחד מהשדות הוא מבנה נתונים (למשל set או $list$) אז נוכל לשנות את הערכים בתוך המבנה.
כדי למנוע זאת השדות שהם מבני נתונים הוגדרו כ-$frozenset$, שזהו $immutable\ set$, כלומר set שלא ניתן לשנות את איבריו.

```
current_site: Union[Junction, Laboratory, ApartmentWithSymptomsReport]
tests_on_ambulance: FrozenSet[ApartmentWithSymptomsReport]
tests_transferred_to_lab: FrozenSet[ApartmentWithSymptomsReport]
nr_matoshim_on_ambulance: int
visited_labs: FrozenSet[Laboratory]
```

.iii

כן, צומת יכול לעבור מ-$close$ ל-$open$ אם מצאנו מסלול יותר זול אליו מהמסלולים שמצאנו עד כה.

$OPEN \leftarrow OPEN \cup \{\text{old\_node}\}$; Move old node from CLOSED to OPEN

.iv

א למימוש שגוי של המתודה expand_state_with_costs היא : State_to_expand.tests_on_ambulance = state_to_expand.tests_on_ambuland
| {apartment}
כלומר עדכון של שדה ספציפי ב-MDAState במקום יצירת MDAState חדש לגמרי בעזרת c'tor . מימוש זה בעייתי מכיוון שאנו עלולים להיתקל במצב מסוים בשנית גם לאחר שפיתחנו אותו כבר בעבר, כפי שצוין בסעיף לעיל. במקרה זה, אם ננסה לעדכן את MDAState מבלי ליצור אחד חדש מה שיקרה בפועל הוא שנדרוס את השדות של המצב בו כבר נתקלנו מכיוון שבפייתון אנחנו מחזיקים מצביע לאובייקט ולא מעתיקים אותו. כלומר, המצב שכרגע נמצא ב-close יתעדכן, וכאשר נחפש מצב מתאים לעדכון ב-close לא נמצא אותו.

.23

נוכיח כי לכל צומת n מתקיים $0 \le h(n) \le h^*(n)$.
מכיוון שמדובר במרחק, ברור כי $h(n) \ge 0$ לכל צומת n.
יהי מצב n. אם נשארה דירה אחת d לבקר בה, המרחק האווירי שלה מעצמה הוא 0 ולכן לפי הגדרת h נקבל: $h(n) =$

$0 \leq h^*(n)$.
אחרת, יהיו $d_1, d_2, ...d_k$ דירות אשר נשאר לאמבולנס לבקר בהן בשלב כלשהו בתכנית. נניח כי המרחק האווירי המקסימלי בין שתי דירות מכל זוגות הדירות הוא $\delta_{max}(d_i, d_j)$.
נשים לב כי לכל מסלול ממצב n למצב מטרה המוביל למחיר אופטימלי $h^*(n)$, האמבולנס יעבור במסלול שלו ב-d_i וב-d_j (נניח בה"כ שיעבור קודם ב-d_i) בדרך כלשהי, נסמנה $dist(d_i, d_j)$. לכן:

$$0 \leq h(n) \overset{(1)}{=} \delta_{max}(d_i, d_j) \overset{(2)}{\leq} dist(d_i, d_j) \overset{(3)}{\leq} h^*(n)$$

כאשר מעבר (1) נובע מהגדרת הפונקציה היוריסטית $h(n)$, מעבר (2) נובע מאי-שוויון המשולש, ומעבר (3) נובע מכך ש-$dist(d_i, d_j)$ הוא חלק מהמסלול האופטימלי. לכן היוריסטיקה קבילה.

.26

היוריסטיקה אינה קבילה, נפריך בעזרת דוגמא נגדית:
נסתכל על המרחב הבא במצב s: יש 4 דירות והאמבולנס נמצא בנקודה (0,0).
דירה A בנקודה (0,0) שבא האמבולנס נמצא כרגע, דירה B בנקודה (2,0), דירה C בנקודה (4,0) ודירה D בנקודה (0,3).
נראה שמתקיים $h(s) > h(s^*)$.

$$h(s) \overset{(1)}{=} cost^{dist}_{MDA}(A \to B \to C \to D) = 9 > 8.6 = cost^{dist}_{MDA}(A \to D \to B \to C) \overset{(2)}{=} h^*(s)$$

מעבר (1) נובע מהגדרת היוריסטיקה שבונה את המסלול כך שהיא תמיד בוחרת את הדירה הבאה במסלול בתור הדירה הקרובה ביותר באותו רגע.
מעבר (2) נובע מכך שזה המסלול שעובר בכל ארבעת הדירות (כאשר מתחילים ב-A) במחיר האופטימלי.
סרטוט להמחשה:

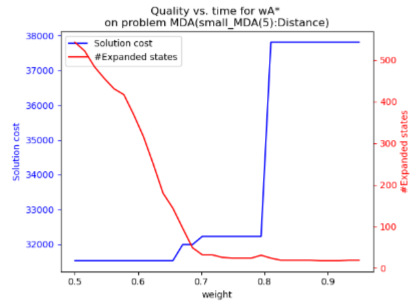
.29

נוכיח כי לכל צומת n מתקיים $0 \leq h(n) \leq h^*(n)$.
מכיוון שמדובר במרחק, ברור כי $h(n) \geq 0$ לכל צומת n.
יהי מצב n. אם נשארה דירה אחת d לבקר בה, העץ הפורש מכיל רק אותה, ולכן ערכו 0. נקבל לפי הגדרת h: $h(n) = 0 \leq h^*(n)$.
אחרת, יהיו $d_1, d_2, ...d_k$ דירות אשר נשאר לאמבולנס לבקר בהן בשלב כלשהו בתכנית.
נסמן ב-G את הגרף המלא שצמתיו הן הדירות שנשאר לאמבולנס לעבור בהן, ומשקלי הקשתות הן המרחק האווירי בין הצמתים שהן מחברות.
נשים לב כי כל מסלול ממצב n למצב מטרה המוביל למחיר אופטימלי $h^*(n)$ עובר בכל הדירות שנותרו, ובפרט פעם אחת בכל דירה. נניח כי הפתרון הוא $P: d_1 \to ... \to d_k$.
נסמן ב-T את הגרף המושרה המתקבל על ידי המסלול P בגרף G.
מעצם בנייתו, T הינו שרוך ולכן בפרט עץ פורש לגרף G. נסמן ב-T^* עץ פורש מינימום לגרף G. נקבל:

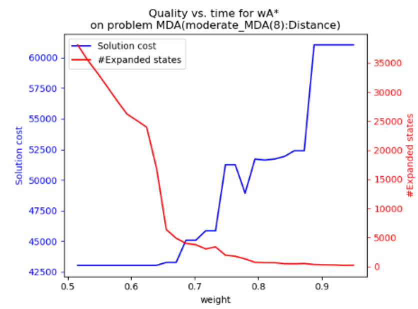
$$0 \leq h(n) \overset{(1)}{=} w(T^*) \overset{(2)}{\leq} w(T) \overset{(3)}{\leq} w(P) = h^*(n)$$

כאשר מעבר (1) נובע מהגדרת הפונקציה היוריסטית $h(n)$, מעבר (2) נובע ממינימליות T^*, ומעבר (3) נובע מכך שלכל $d_i \to d_{i+1}$ במסלול P, מתקיים כי $\delta(d_i, d_{i+1}) \leq dist(d_i, d_{i+1})$ כאשר δ הוא המרחק האווירי בין הצמתים ו-$dist$ הוא המרחק במסלול. לכן היוריסטיקה קבילה.

.30



האזור הכדאי על פי הגרף הוא (בערך) $0.65 \le w \le 0.8$ מכיוון שבאזור יש ירידה חדה בזמן הריצה אך איכות הפתרון לא מתרחקת בהרבה מהפתרון האופטימלי.



האזור הכדאי על פי הגרף הוא (בערך) $0.65 \le w \le 0.725$ מכיוון שבאזור יש ירידה חדה בזמן הריצה אך איכות הפתרון לא מתרחקת בהרבה מהפתרון האופטימלי.

.31

MDAMSTAirDistHeuristic	MDASumAirDistHeuristic	MDAMaxAirDistHeuristic
לא	לא	לא
לא	לא	לא

.32

```
MDACost(dist=  43034.794m, money=     95.847NIS, tests-travel= 176505.013m)
distance
MDACost(dist=  54951.037m, money=     77.201NIS, tests-travel= 172922.318m)
monetary
```

.34

נשים לב כי עבור צומת n, $h^*(n)$ היא הסכום המינימלי של כל הבדיקות מוכפלות במרחק שהן עברו עד שהגיעו למעבדה.
נוכיח כי לכל צומת n מתקיים $0 \le h(n) \le h^*(n)$.
מכיוון שמדובר במרחק, ברור כי $h(n) \ge 0$ לכל צומת n.
יהי מצב n. ויהי מסלול P למצב מטרה כך שמחירו אופטימלי, כלומר $cost_{MDA}^{test\ travel}(P) = h^*(n)$.
לכל דירה d_i שנשאר לבקר בה נסמן ב-$d_i.travelled$ את המרחק שעברו הבדיקות מ-d_i מהרגע שנאספו עד למסירתן למעבדה כלשהי, וב-L_i את המעבדה הקרובה ביותר ל-d_i. מתקיים:

$$h(n) = \sum_{i=1}^{n} \delta(d_i, L_i) \cdot d_i.roomates \overset{(1)}{\le} \sum_{i=1}^{n} d_i.travelled \cdot d_i.roomates \overset{(2)}{=} cost_{MDA}^{test\ travel} = h^*(n)$$

כאשר מעבר (1) נובע מכך שהמרחק המינימלי שבדיקה יכולה לעבור הוא בדיוק המרחק מהדירה בה היא נלקחה למעבדה הקרובה ביותר לדירה זו, ומעבר (2) הוא לפי הגדרה. לכן היוריסטיקה קבילה.

.35

```
MDACost(dist=  43034.794m, money=     95.847NIS, tests-travel= 176505.013m)
distance
MDACost(dist=  54951.037m, money=     77.201NIS, tests-travel= 172922.318m)
monetary
MDACost(dist=  93355.782m, money=    127.001NIS, tests-travel= 131265.153m)
tests travel
```

.36

נוכיח:
נניח בשלילה ש-A_1 לא החזיר פתרון. כלומר, הפעלת האופרטור לא הייתה חוקית עבור אף מצב.
מכך נובע שלכל מסלול P התקיים ש-$cost^{dist}_{MDA}(P) > (1+\varepsilon)\cdot C^*_{dist}$ אבל נתון כי היה קיים פתרון במרחב המקורי S_{MDA} ולכן בוודאות קיים מסלול P שעבורו $cost^{dist}_{MDA}(P) = C^*_{dist} \le (1+\varepsilon)\cdot C^*_{dist}$ ולכן נקבל סתירה לקבילות של A^*, לכן A_1 תמיד מחזיר פתרון.

.37

נוכיח:
יהי P הפתרון האופטימלי על פי הקריטריון המשולב, אז בפרט מתקיים כי $cost^{dist}_{MDA}(P) \le (1+\varepsilon)\cdot C^*_{dist}$, ולכן בשלב השלישי באלגוריתם A_1 הפעלת האופרטור היא חוקית והמצב יפותח. מכיוון ש-P אופטימלי, מתקיים עבורו כי $cost^{test\ travel}_{MDA}(P) \le cost^{test\ travel}_{MDA}(P')$ לכל $P \neq P'$, ומכיוון ש-UCS הוא אלגוריתם קביל, מובטח כי הוא יחזיר את הפתרון האופטימלי עבור מחיר זה, כלומר את הפתרון P.

.38

	Distance cost:	Tests travel cost:
$cost^{dist}_{MDA}$	43034.794m	176505.031m
$cost^{tests\ travel}_{MDA}$	93355.782m	131265.153m
$cost^{merged}_{MDA}$	66696.615m	134889.839m

ניתן לראות על פי הטבלה שקיבלנו מסלול יותר יקר מבחינת פונקציית המחיר $cost^{dist}_{MDA}$ (אך עדיין בתחום הנדרש לפי ε כפי שנראה למטה) וכמו כן מסלול טיפה יותר יקר ביחס לפונקציית המחיר $cost^{test\ travel}_{MDA}$ אך קרוב למדי לאופטימלי.
כלומר אכן התקיים האיזון בין שני המדדים.

$$\frac{DistCost(ReturnedSolution)}{C^*_{dist}} - 1 = \frac{66696.615}{43034.794} - 1 = 0.5498 < 0.6 = \varepsilon$$

ניתן לראות כי אכן נשמר ערך ה-ε הנקוב.

.39

הטענה אינה נכונה. נפריך בעזרת דוגמא נגדית.
נסתכל על הסימולציה הבאה: האמבולנס נמצא בנקודת ההתחלה $S(0,0)$ ללא מטושים כלל. דירה A בנקודה $(1,0)$ עם שני דיירים. מעבדה B בנקודה $(2,0)$ עם מטוש אחד במלאי. ומעבדה C בנקודה $(3,0)$ עם מטוש אחד במלאי.
תחילה נשים לב כי הפתרון האופטימלי שיוחזר עם פונקציית העלות $cost^{dist}_{MDA}$ הוא $S \to B \to C \to A$ שמחירו $C^*_S = 5$. כעת נגדיר $\varepsilon = 0.1$, כלומר $max\_distance\_cost = 5.5$, ונראה כי האלגוריתם מחזיר שאין פתרון:

צעד	$open$	$close$	הצומת הבא לפיתוח	הסבר
1	$s_1 = \{S, \emptyset, \emptyset, 0, \emptyset\}$	$\emptyset$	s_1	מפתחים את S לפי האלגוריתם
2	$s_2 = \{C, \emptyset, \emptyset, 1, \{C\}\}, g = 0, h = 2, f = 2$ $s_3 = \{B, \emptyset, \emptyset, 1, \{B\}\}, g = 0, h = 2, f = 2$	s_1	s_2	מכיוון שערכי f של שני הצמתים זהים, ניתן לבחור איזה צומת האלגוריתם יפתח.
3	$s_4 = \{B, \emptyset, \emptyset, 2, \{B, C\}\}, g = 0, h = 2, f = 2$ $s_3 = \{B, \emptyset, \emptyset, 1, \{B\}\}, g = 0, h = 2, f = 2$	s_1 s_2	s_4	כשבאנו לפתח את צומת C עדיין אין ברשותנו מספיק מטושים כדי ללכת לדירה A, לכן הצומת הבא היחיד שניתן להגיע אליו זה צומת B. נשים לב כי זה לא אותו מצב כמו s_3 שהוא מצב שמתאר מסלול שבו מתחילים בצומת B. כמו כן, ערכי f עדיין זהים ולכן נוכל לבחור להמשיך מ-s_4.
4	$s_5 = \{A, \{A\}, \emptyset, 0, \{B, C\}\}, g = 0, h = 2, f = 2$ $s_3 = \{B, \emptyset, \emptyset, 1, \{B\}\}, g = 0, h = 2, f = 2$	s_1 s_2 s_4	s_2	הגענו לדירה ולקחנו את הבדיקות. ערכי f בין המצבים עדיין זהים ולכן נמשיך לפתח את צומת A.
5	$s_3 = \{B, \emptyset, \emptyset, 1, \{B\}\}, g = 0, h = 2, f = 2$	s_1 s_2 s_4 s_5	s_3	$(*)$ הסבר מתחת לטבלה.
6	$s_8 = \{C, \emptyset, \emptyset, 2, \{B, C\}\}, g = 0, h = 2, f = 2$	s_1 s_2 s_4 s_5 s_3	s_8	מפתחים את s_3 לפי האלגוריתם
7	$s_9 = \{A, \{A\}, \emptyset, 0, \{B, C\}\}, g = 0, h = 2, f = 2$	s_1 s_2 s_4 s_5 s_3 s_8	s_9	הצומת הבא שהאלגוריתם יבוא לפתח הוא s_9. אך נשים לב כי s_9 זהה ל-s_5 עם אותו ערך f. ולכן האלגוריתם ייראה שהמצב הזה נמצא ב-$close$ ולא יפתח אותו פעם נוספת. האלגוריתם סיים לרוץ על כל המצבים ולא החזיר פתרון

$(*)$ הסבר צעד 5: לאחר הפיתוח של s_5 בשלב זה נוצרים שני המצבים הבאים:

$$s_6 = \{B, \emptyset, \{A\}, 0, \{B, C\}\}$$
$$s_7 = \{C, \emptyset, \{A\}, 0, \{B, C\}\}$$

נשים לב כי המסלול שעברנו עד כה הוא: $P = C \to B \to A$, ולכן מתקיים כי:

$$cost_{MDA}^{dist}(s_6) = cost_{MDA}^{dist}(C \to B \to A \to B) = 6 > 5.5$$
$$cost_{MDA}^{dist}(s_7) = cost_{MDA}^{dist}(C \to B \to A \to C) = 7 > 5.5$$

כתוצאה מכך, בגלל ההגבלה שנתנו אף אחד מהמצבים s_6, s_7 לא ייכנס ל-$open$. האלגוריתם יסיים את הפיתוח של s_5 ויעביר אותו ל-$close$. הצומת הבא לפיתוח יהיה s_3 שהוא היחיד שכרגע ב-$open$.

ניתן לראות כי למרות **שקיים הפתרון** C_S^* האלגוריתם **החזיר שאין פתרון.**

.40

.41

נשים לב כי בשני השלבים הראשונים האלגוריתמים A_1 ו-A_2 זהים. היתרון הצפוי של A_2 על פני A_1 נובע מכך שב-A_2 בשלב השלישי אנו מריצים $Astar$ על אותו מרחב כמו בשלב הראשון, ואילו ב-A_1 אנו מריצים $Astar$ על "מרחב המסלולים" שגודלו הוא מסדר גודל $2^{|S|}$ (כגודל $P(S)$) כאשר S הוא המרחב מהשלב הראשון, לכן זמן הריצה של A_1 יהיה גדול משל A_2.

.44

```
A* (h=MDA-MST-AirDist, w=0.500)   time:   0.42   #dev: 543     |space|: 877
A*eps (h=MDA-MST-AirDist, w=0.500)   time:   1.17   #dev: 492     |space|: 821
```

כפי שניתן לראות מתוצאות ההרצה, אכן חסכנו במספר הפיתוחים, פיתחנו כ-10% פחות צמתים. הגמישות של $A^*\varepsilon$ תעזור לנו בכך שהיא תיתן אופציה לצמתים עם ערך g גבוה יותר (לכל היותר ב-ε) אך עם ערך h נמוך יותר להיכנס ל-open. כלומר, בחיפוש שלנו נבדוק גם צמתים אשר הפונקציה היוריסטית תעריך אותם כיותר קרובים לפתרון, למרות שערך ה-g שלהם אינו אופטימלי.

חלק_י':

א'.

המדד הביצועי שאנו משפרים הוא **זכרון.** הסיבה לכך היא שהאלגוריתם IDA^* הוא אלגוריתם איטרטיבי המחפש לעומק, ולכן בעל דרישות זכרון נמוכות.

ב'.

i. זמן ריצה.

ii. עבור פונקציות יוריסטיות מסוימות יכול להיות שנרוויח קצת מאוד צמתים בעומק כל איטרציה, ולכן הריצה תהיה מאוד ארוכה.

iii. מדד זה נפגע פחות ביחס מהאופן שבו הוא נפגע ב-ID-DFS לעומת BFS מכיוון שעבור IDA^* קיימות פונקציות יוריסטיות קבילות שעבורן הבעיה שתיארנו בסעיף הקודם לא תיגרם ונקבל זמן ריצה טוב יותר. לעומת זאת ריצה של ID-DFS תמיד מכילה באיטרציה האחרונה שלה ריצה של BFS ולכן זמנה יהיה לכל הפחות כמו זמן הריצה של BFS ולרוב אף גבוה יותר.

ג'.

i. במקרה הגרוע ביותר, כל איטרציה נגדיל את $nextFlimit$ ב-$\frac{1}{k}$ כאשר ערכו ההתחלתי הוא $Q_k(h(I))$ כפי שהוגדר. ברגע ש-f יגיע לערך $Cost(A(S))$ האלגוריתם ימצא פתרון ויעצור. לכן מספר האיטרציות לכל היותר יהיה

$$\#max\_iterations = \left\lceil \frac{Cost(A(S)) - Q_k(h(I))}{\frac{1}{k}} \right\rceil = \lceil k \cdot (Cost(A(S)) - Q_k(h(I))) \rceil$$

ii. החסם ההדוק על $\varepsilon(A_1,S)$ הוא $\frac{1}{k}$, כלומר מתקיים $\varepsilon(A_1,S) < \frac{1}{k}$.
נוכיח: מתקיים כי $Q_k(origNextFLimit) \leq C^*_S$ לכל שלב באלגוריתם עד מציאת פתרון, מכיוון שמובטח לנו שקיים צומת שיביא אותנו לפתרון האופטימלי שערכו C^*_S, ולכן ערך ה-f שלו יהיה קטן שווה מ-C^*_S. נסתכל על האיטרציה האחרונה לפני שאנו מוצאים את הפתרון בה נקראת הפונקציה $nextFLimit$. כאשר $C^*_S - prevFLimit \leq \frac{1}{k}$ נקבל ש-$Q_k(origNextFLimit) \leq C^*_S \leq prevFLimit + \frac{1}{k}$ ואז:

$$nextFLimit = max\{prevFLimit + \frac{1}{k}, Q_k(origNextFLimit)\} = prevFLimit + \frac{1}{k}$$

כעת, באיטרציה האחרונה האלגוריתם יכול למצוא כל פתרון בטווח $[C^*_S, prevFLimit + \frac{1}{k}]$ נשים לב כי כאשר $C^*_S - prevFLimit \longrightarrow 0$ נקבל את הטווח $[C^*_S, C^*_S + \frac{1}{k})$ ולכן הסטייה המקסימלית מהפתרון חסומה ע"י $\frac{1}{k}$.