שי בן-ישי ,308052463

אביגיל כהן רמון, 318691078

## חלק א': מבוא והנחיות

1. נתון, מחשב יחיד יכול לבחון 230 סידורים בשנייה.

|  |  |  |  |
| --- | --- | --- | --- |
| Calculation time |  |  |  |
| < 1 sec | 21.7911 | 3628800 | 10 |
| 5.799[sec] | 32.5359 | 6227020800 | 13 |
| 20.29[mins] | 40.2501 | 1.3077\*1012 | 15 |
| 5.41[hours] | 44.2501 | 2.0923\*1012 | 16 |
| 3.83[days] | 48.3376 | 3.5569\*1014 | 17 |
| 71.84[years] | 61.0774 | 2.4329\*1018 | 20 |
| 1508.82[years] | 65.4697 | 5.1091\*1019 | 21 |
| 18.32[million years] | 79.0377 | 6.2045\*1023 | 24 |

## חלק ב': הגדרת מרחב החיפוש במפה

## חלק ג': הגדרת מרחב החיפוש של בעיית ההובלות

1. **ערכי קיצון אפשריים של דרגת היציאה במרחב החיפוש:**דרגה מקסימאלית: k+m  
   דרגה מינימלית: 0
2. **האם ייתכנו מעגלים:**לא ייתכן מעגל.  
   מעגל ייתכן רק במצב בו המצב ההתחלתי הוא עם מיקום הוא של מעבדה, כיון שלא ניתן לבקר בדירה יותר מפעם אחת.  
   נסתכל על מקרים בו מבוצע ביקור חוזר במעבדה מסוימת:  
   מטרת הביקור היא לקיחת מטושים, זהו ביקור ראשון במעבדה בהכרח ע"פ הגדרת האופרטורים. אם כך, בביקור חוזר הערך של VisitedLabs ישתנה ולכן לא ייתכן מעגל.  
   כלומר, המצב שיוצר את המעגל הוא מצב של ביקור חוזר במעבדה לשם העברת בדיקות. מאחר והעברנו בדיקות, קבוצת ה Transferred משתנה ולכן מדובר על מצבים שונים.
3. **מספר המצבים במרחב, האם כולם ישיגים:**לא. דוגמא למצב לא ישיג: מצב בו מספר המטושים גדול מסך כל המטושים בכל המעבדות + מספר המטושים במצב ההתחלתי:
4. **האם ייתכנו בורות ישיגים מהמצב ההתחלתי שאינם מצבי מטרה במרחב המצבים:**כן. מצב בו לא סיימנו לבקר בכל הדירות, אך אין לנו דרך להשיג עוד מטושים (כלומר ביקרנו כבר בכל המעבדות).  
   מצב זה יקרה אם מספר המטושים הכולל הקיים בכל המעבדות פלוס מספר המטושים ההתחלתי קטן ממספר הבדיקות שעלינו לבצע.
5. **טווח האורכים האפשריים של מסלולים במרחב ממצב התחלתי אל מצב סופי   
   (אורך מסלול = מספר קשתות):**אורך מסלול מינימלי הוא k+1: מספר המטושים ההתחלתי מספיק לנו ע"מ לבצע את כל הבדיקות, ישנו מספיק מקום על האמבולנס כדי לאחסן אותם, ואז נותר לבקר במעבדה אחת ע"מ להעביר את הבדיקות.

אורך מסלול מקסימלי הוא 2k+m: עוברים בכל המעבדות ע"מ לקחת את המטושים, ואז על כל ביקור בדירה מבצעים ביקור במעבדה

**הגדרה פורמלית וישירה של פונקציית העוקב המתאימה לבעיה זו (ללא שימוש בקבוצת האופרטורים):**

## חלק ד': מתחילים לתכנת

1. פלט ריצה של MapProblem אחרי השינוי:

Running all experiments

Solve the map problem.

StreetsMap(src: 54 dst: 549) UniformCost time: 0.68 #dev: 17354 |space|: 17514 total\_g\_cost: 7465.52560 |path|: 136 path: [ 54 ==> 55 ==> 56 ==> 57 ==> 58 ==> 59 ==> 60 ==> 28893 ==> 14580 ==> 14590 ==> 14591 ==> 14592 ==> 14593 ==> 81892 ==> 25814 ==> 81 ==> 26236 ==> 26234 ==> 1188 ==> 33068 ==> 33069 ==> 33070 ==> 15474 ==> 33071 ==> 5020 ==> 21699 ==> 33072 ==> 33073 ==> 33074 ==> 16203 ==> 9847 ==> 9848 ==> 9849 ==> 9850 ==> 9851 ==> 335 ==> 9852 ==> 82906 ==> 82907 ==> 82908 ==> 82909 ==> 95454 ==> 96539 ==> 72369 ==> 94627 ==> 38553 ==> 72367 ==> 29007 ==> 94632 ==> 96540 ==> 9269 ==> 82890 ==> 29049 ==> 29026 ==> 82682 ==> 71897 ==> 83380 ==> 96541 ==> 82904 ==> 96542 ==> 96543 ==> 96544 ==> 96545 ==> 96546 ==> 96547 ==> 82911 ==> 82928 ==> 24841 ==> 24842 ==> 24843 ==> 5215 ==> 24844 ==> 9274 ==> 24845 ==> 24846 ==> 24847 ==> 24848 ==> 24849 ==> 24850 ==> 24851 ==> 24852 ==> 24853 ==> 24854 ==> 24855 ==> 24856 ==> 24857 ==> 24858 ==> 24859 ==> 24860 ==> 24861 ==> 24862 ==> 24863 ==> 24864 ==> 24865 ==> 24866 ==> 82208 ==> 82209 ==> 82210 ==> 21518 ==> 21431 ==> 21432 ==> 21433 ==> 21434 ==> 21435 ==> 21436 ==> 21437 ==> 21438 ==> 21439 ==> 21440 ==> 21441 ==> 21442 ==> 21443 ==> 21444 ==> 21445 ==> 21446 ==> 21447 ==> 21448 ==> 21449 ==> 21450 ==> 21451 ==> 621 ==> 21452 ==> 21453 ==> 21454 ==> 21495 ==> 21496 ==> 539 ==> 540 ==> 541 ==> 542 ==> 543 ==> 544 ==> 545 ==> 546 ==> 547 ==> 548 ==> 549]

## חלק ה': אלגוריתם A\*

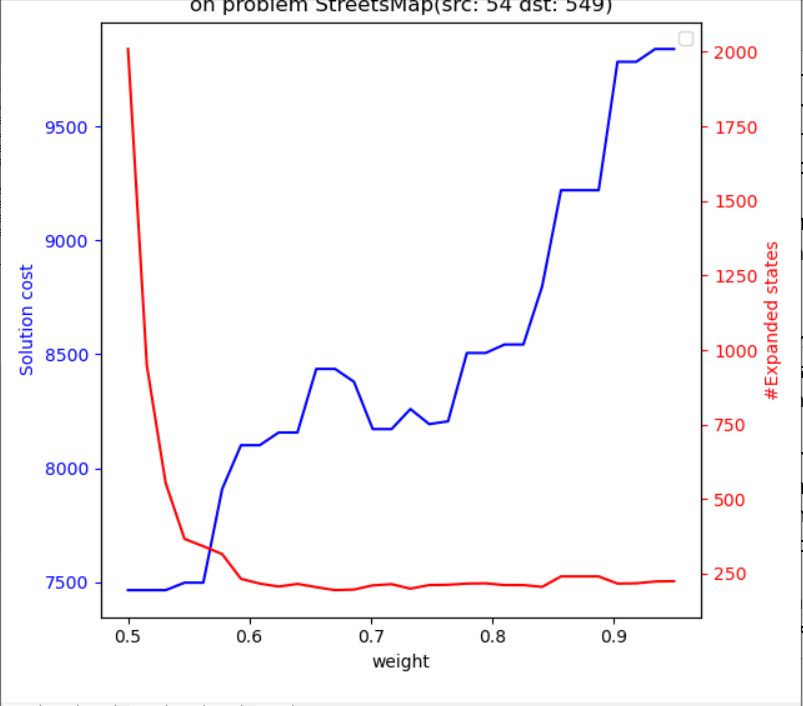
1. רטוב.
2. רטוב.
3. רטוב.
4. רטוב + יבש.  
   פלט:  
   מספר פיתוחי המצבים היחסי שחסכנו לעומת הריצה העיוורת:  
   מספר הפיתוחים בריצה עיוורת: 17473  
   מספר הפיתוחים ב AirHeuristic: 2015  
     
   החיסכון: 0.885

Solve the map problem.

StreetsMap(src: 54 dst: 549) UniformCost time: 0.86 #dev: 17354 |space|: 17514 total\_g\_cost: 7465.52560 |path|: 136 path: [ 54 ==> 55 ==> 56 ==> 57 ==> 58 ==> 59 ==> 60 ==> 28893 ==> 14580 ==> 14590 ==> 14591 ==> 14592 ==> 14593 ==> 81892 ==> 25814 ==> 81 ==> 26236 ==> 26234 ==> 1188 ==> 33068 ==> 33069 ==> 33070 ==> 15474 ==> 33071 ==> 5020 ==> 21699 ==> 33072 ==> 33073 ==> 33074 ==> 16203 ==> 9847 ==> 9848 ==> 9849 ==> 9850 ==> 9851 ==> 335 ==> 9852 ==> 82906 ==> 82907 ==> 82908 ==> 82909 ==> 95454 ==> 96539 ==> 72369 ==> 94627 ==> 38553 ==> 72367 ==> 29007 ==> 94632 ==> 96540 ==> 9269 ==> 82890 ==> 29049 ==> 29026 ==> 82682 ==> 71897 ==> 83380 ==> 96541 ==> 82904 ==> 96542 ==> 96543 ==> 96544 ==> 96545 ==> 96546 ==> 96547 ==> 82911 ==> 82928 ==> 24841 ==> 24842 ==> 24843 ==> 5215 ==> 24844 ==> 9274 ==> 24845 ==> 24846 ==> 24847 ==> 24848 ==> 24849 ==> 24850 ==> 24851 ==> 24852 ==> 24853 ==> 24854 ==> 24855 ==> 24856 ==> 24857 ==> 24858 ==> 24859 ==> 24860 ==> 24861 ==> 24862 ==> 24863 ==> 24864 ==> 24865 ==> 24866 ==> 82208 ==> 82209 ==> 82210 ==> 21518 ==> 21431 ==> 21432 ==> 21433 ==> 21434 ==> 21435 ==> 21436 ==> 21437 ==> 21438 ==> 21439 ==> 21440 ==> 21441 ==> 21442 ==> 21443 ==> 21444 ==> 21445 ==> 21446 ==> 21447 ==> 21448 ==> 21449 ==> 21450 ==> 21451 ==> 621 ==> 21452 ==> 21453 ==> 21454 ==> 21495 ==> 21496 ==> 539 ==> 540 ==> 541 ==> 542 ==> 543 ==> 544 ==> 545 ==> 546 ==> 547 ==> 548 ==> 549]

StreetsMap(src: 54 dst: 549) A\* (h=0, w=0.500) time: 0.85 #dev: 17354 |space|: 17514 total\_g\_cost: 7465.52560 |path|: 136 path: [ 54 ==> 55 ==> 56 ==> 57 ==> 58 ==> 59 ==> 60 ==> 28893 ==> 14580 ==> 14590 ==> 14591 ==> 14592 ==> 14593 ==> 81892 ==> 25814 ==> 81 ==> 26236 ==> 26234 ==> 1188 ==> 33068 ==> 33069 ==> 33070 ==> 15474 ==> 33071 ==> 5020 ==> 21699 ==> 33072 ==> 33073 ==> 33074 ==> 16203 ==> 9847 ==> 9848 ==> 9849 ==> 9850 ==> 9851 ==> 335 ==> 9852 ==> 82906 ==> 82907 ==> 82908 ==> 82909 ==> 95454 ==> 96539 ==> 72369 ==> 94627 ==> 38553 ==> 72367 ==> 29007 ==> 94632 ==> 96540 ==> 9269 ==> 82890 ==> 29049 ==> 29026 ==> 82682 ==> 71897 ==> 83380 ==> 96541 ==> 82904 ==> 96542 ==> 96543 ==> 96544 ==> 96545 ==> 96546 ==> 96547 ==> 82911 ==> 82928 ==> 24841 ==> 24842 ==> 24843 ==> 5215 ==> 24844 ==> 9274 ==> 24845 ==> 24846 ==> 24847 ==> 24848 ==> 24849 ==> 24850 ==> 24851 ==> 24852 ==> 24853 ==> 24854 ==> 24855 ==> 24856 ==> 24857 ==> 24858 ==> 24859 ==> 24860 ==> 24861 ==> 24862 ==> 24863 ==> 24864 ==> 24865 ==> 24866 ==> 82208 ==> 82209 ==> 82210 ==> 21518 ==> 21431 ==> 21432 ==> 21433 ==> 21434 ==> 21435 ==> 21436 ==> 21437 ==> 21438 ==> 21439 ==> 21440 ==> 21441 ==> 21442 ==> 21443 ==> 21444 ==> 21445 ==> 21446 ==> 21447 ==> 21448 ==> 21449 ==> 21450 ==> 21451 ==> 621 ==> 21452 ==> 21453 ==> 21454 ==> 21495 ==> 21496 ==> 539 ==> 540 ==> 541 ==> 542 ==> 543 ==> 544 ==> 545 ==> 546 ==> 547 ==> 548 ==> 549]

StreetsMap(src: 54 dst: 549) A\* (h=AirDist, w=0.500) time: 0.16 #dev: 2015 |space|: 2229 total\_g\_cost: 7465.52560 |path|: 136 path: [ 54 ==> 55 ==> 56 ==> 57 ==> 58 ==> 59 ==> 60 ==> 28893 ==> 14580 ==> 14590 ==> 14591 ==> 14592 ==> 14593 ==> 81892 ==> 25814 ==> 81 ==> 26236 ==> 26234 ==> 1188 ==> 33068 ==> 33069 ==> 33070 ==> 15474 ==> 33071 ==> 5020 ==> 21699 ==> 33072 ==> 33073 ==> 33074 ==> 16203 ==> 9847 ==> 9848 ==> 9849 ==> 9850 ==> 9851 ==> 335 ==> 9852 ==> 82906 ==> 82907 ==> 82908 ==> 82909 ==> 95454 ==> 96539 ==> 72369 ==> 94627 ==> 38553 ==> 72367 ==> 29007 ==> 94632 ==> 96540 ==> 9269 ==> 82890 ==> 29049 ==> 29026 ==> 82682 ==> 71897 ==> 83380 ==> 96541 ==> 82904 ==> 96542 ==> 96543 ==> 96544 ==> 96545 ==> 96546 ==> 96547 ==> 82911 ==> 82928 ==> 24841 ==> 24842 ==> 24843 ==> 5215 ==> 24844 ==> 9274 ==> 24845 ==> 24846 ==> 24847 ==> 24848 ==> 24849 ==> 24850 ==> 24851 ==> 24852 ==> 24853 ==> 24854 ==> 24855 ==> 24856 ==> 24857 ==> 24858 ==> 24859 ==> 24860 ==> 24861 ==> 24862 ==> 24863 ==> 24864 ==> 24865 ==> 24866 ==> 82208 ==> 82209 ==> 82210 ==> 21518 ==> 21431 ==> 21432 ==> 21433 ==> 21434 ==> 21435 ==> 21436 ==> 21437 ==> 21438 ==> 21439 ==> 21440 ==> 21441 ==> 21442 ==> 21443 ==> 21444 ==> 21445 ==> 21446 ==> 21447 ==> 21448 ==> 21449 ==> 21450 ==> 21451 ==> 621 ==> 21452 ==> 21453 ==> 21454 ==> 21495 ==> 21496 ==> 539 ==> 540 ==> 541 ==> 542 ==> 543 ==> 544 ==> 545 ==> 546 ==> 547 ==> 548 ==> 549]

1. הגרף שנוצר:  
   

הסבר הגרף:  
תיאור מספר הצמתים שפותחו ואיכות הפתרון של ריצת אלגוריתם A\* המשתמש ביוריסטיקת AirDistHeuristic עבור ערכי w שונים בטווח (0.5, 1) על בעיית map\_problem.  
ניתן לראות כי חלה ירידה תלולה במספר הצמתים המפותחים עבור ערכי w מ 0.5 עד לאיזור של הנקודה הירוקה, ומשם היא מתחילה להתמתן עד שמתייצבת יחסית.  
לעומת זאת, נראה כי מחיר הפתרון המתקבל עבור ערכי w שונים נמצא במגמת עלייה אך היא אינה אחידה ויש הרבה קפיצות. העלייה המשמעותית במחיר חלה סביב איזור w = 0.6.

ערך ה w (בהנחה ואנחנו רוצים לשמור על קבילות הפתרון):  
0.5292~, ערך ה x של הנקודה המסומנת בירוק.  
ניתן לראות כי בטווח מ w לנקודה הירוקה המחיר המתקבל הוא מחיר הפתרון האופטימלי (w=0.5).  
ע"פ כלל האצבע, ככל שנגדיל את ערך ה w שנבחר נקבל (מגמתית) מספר פיתוחים קטן יותר של צמתים.  
אם כך, נבחר את ערך ה w המקסימאלי עבורו מחיר הפתרון עדיין אופטימלי. זהו ה w של הנקודה המסומנת.

ניתן לראות בגרף את ביטוי כלל האצבע לפיו "ככל ש w קטן יותר כך הפתרון איכותי יותר ומספר הפיתוחים גדול יותר". הגרף המייצג את מספר הפיתוחים נמצא במגמת ירידה מ w=0.5 המייצג את הפתרון האופטימלי במקביל לכך שגרף עלות הפתרון הולך וגדל.  
הדגש כי הכלל הנ"ל מצביע על מגמה כללית ואינו נכון באופן גורף, בא לידי ביטוי באיזור בין 0.7 ל 0.8: ניתן לראות כי קיים שם מינימום לוקאלי בגרף מספר הצמתים שפותחו, נסמן את משקלו w. ניתן לראות שקיים w1 קטן מ w מספר הצמתים שפותחו גדול יותר מאשר מספר הצמתים שפותחו עבור w.

## חלק ו': מימוש בעיית מד"א

1. רטוב.
2. רטוב.
3. שורת הקוד הרלוונטית שקובעת שאובייקטים מהטיפוס MDAState יהיו בלתי ניתנים לשינוי:  
   שורה 14 בקובץ mda\_problem.py:

@dataclass(frozen=True)

**האם שורה זו מספיקה? מה עוד מבטיח שלא יהיה ניתן לשנות בטעות את האובייקט ו\או המבנים שהוא מחזיק?**שורה זו אינה מספיקה.  
שורה זו מונעת ניסיונות שינוי של ערך השדות של MDAState.  
בשדות מסוג value אכן תהיה אכיפה כמצופה.  
בשדות מסוג reference כגון set, תהיה אכיפה של שינוי הכתובת המוחזקת בשדה זה בלבד.  
שינוי בערכי ה set עצמו לא יגרום לשינוי הכתובת, ולכן אין מניעה של שינויים כאלו.

הפתרון לכך הוא הגדרת כל אחד מה set הקיימים ב MDAState כ frozenset בעצמו (כפי שנעשה בשורות 40-43).

**הסבר למה אנחנו רוצים לעשות זאת ספציפית עבור הטיפוס MDAState, תן דוגמא למימוש של המתודה expand\_state\_with\_cost במחלקה MDAProblem שממחיש את הצורך בטיפוסים "קפואים".**

הצורך בהגדרת MDAState כ frozen נובע מכך שבמהלך פיתוח צמתים עוקבים עבור צומת מסוים לא נרצה לאפשר למצב עוקב שמפותח לשנות ערכים של מצב האבא שלו, מאחר וכל צומת עוקב הוא חלק ממסלול שונה. שינויים כאלה עלולים לגרום להגדרת מצבים שגויים ולשגיאות במהלך החיפוש.

דוגמא מהמתודה MDAProblem.expand\_state\_with\_cost הממחישה את הצורך ב frozen:  
הפונקציה מחזירה צמתים עוקבים עבור צומת מסוים לפיתוח תוך שימוש ב yield.  
בפיתוח עוקב הנוצר מהפעלת אופרטור ביקור במעבדה x,  
ניתן לשגות ולשנות ישירות את הרשימה המוחזקת בשדה visited\_labs של state\_to\_expand, כלומר להוסיף את המעבדה x ל state\_to\_expand.visited\_lab, ולבצע השמה שלה ל visited\_labs במצב החדש שפיתחנו. כתוצאה משימוש ישיר זה, גם מצב האבא וגם המצב העוקב חולקים מצביע לרשימה visited\_lab כאשר הרשימה הזו שגויה עבור מצב האבא.  
בפעם הבאה שנרצה לפתח עוקב ל state\_to\_expand, השגיאה הנ"ל תגרום לפיתוח של מצבים עוקבים שגויים.

1. רטוב (מימוש יוריסטיקת MDAMaxAirDistHeuristic).
2. רטוב.
3. **הוכח\הפרך: היוריסטיקה MDAMaxAirDistHeuristic הינה קבילה. לטובת סעיף זה, הנח שכל הנקודות במפת הכבישים הן נקודות ב והמרחק בין זוג נק' הוא המרחק האוקלידי.**היוריסטיקה הינה קבילה.  
   יוריסטיקה קבילה היא יוריסטיקה עבורה מתקיים: , כלומר היא מעריכה "אופטימית" את מחיר המסלול מכל מצב s למצב מטרה.

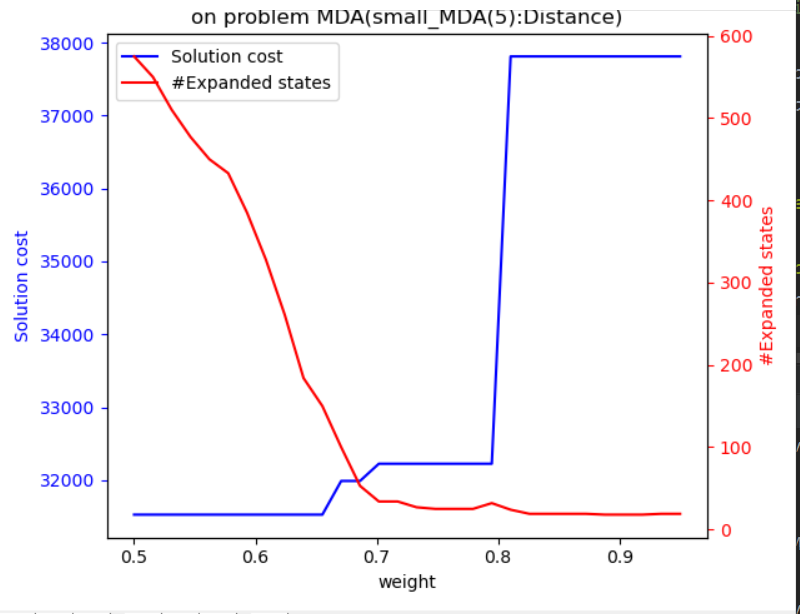
הוכחה:  
יוריסטיקה זו נותנת לכל צומת ערך יוריסטי של המרחק האווירי הכי גדול בין שתי נקודות בהן המסלול צריך לעבור שעוד לא עברו בהם, כלומר המרחק המקסימאלי בין שתי דירות שעדיין לא בוקרו.  
במקרה בו נשארה דירה אחת בלבד שעוד לא בוקרה, מתקיים כי h(s) <= h\*(s), מאחר והמרחק האווירי הוא יוריסטיקה אופטימית ולכן קבילה (הוכח בהרצאה).  
ע"פ אי שוויון המשולש, כל הוספה של דירה נוספת תאריך את המסלול (אפילו אם מרחקה מ s קטן יותר) ולכן נקבל כי לכל s מתקיים h(s) <= h\*(s). אם כך קיבלנו כי היוריסטיקה MDAMaxAirDistHeuristic קבילה ע"פ הגדרה.

1. רטוב (מימוש יוריסטיקת MDASumAirDistHeuristic).
2. רטוב.
3. **הוכח\הפרך: היוריסטיקה MDASumAirDistHeuristic הינה קבילה (עבור פונ' המחיר ). לטובת סעיף זה, הנח שכל הנקודות במפת הכבישים הן נקודות ב והמרחק בין זוג נקודות הוא המרחק האוקלידי.**  
   היוריסטיקה אינה קבילה.  
   יוריסטיקה קבילה היא יוריסטיקה עבורה מתקיים: .  
   *נסתכל על מקרה בו k=3, כאשר מיקומי הדירות הם הנקודות (-3, 0), (1, 0), (10, 0) ומיקום האמבולנס הוא (0,0). נסמן את המצב ההתחלתי s.  
   ע"פ היוריסטיקה, h(s) = sum\_air\_cost((0,0) => (1, 0) => (-3, 0) => (10, 0)) = 17.  
   לעומת זאת, המסלול האופטימלי עבור דירות אלו מ s הוא (0,0) => (-3 0) => (1, 0) => (10, 0),  
    כלומר h\*(s), הוא 16.*

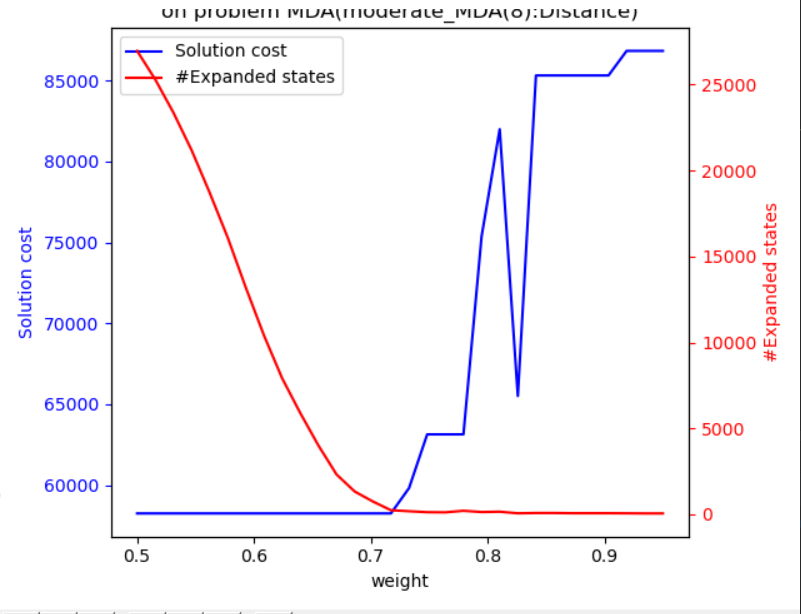
קיבלנו שקיים s עבורו h(s) > h\*(s), ולכן יוריסטיקה זו אינה קבילה ע"פ הגדרה.

1. רטוב (מימוש יוריסטיקת MDAMSTAirDistHeuristic).
2. רטוב.
3. **הוכח\הפרך: היוריסטיקה MDAMSTAirDistHeuristic הינה קבילה (עבור פונ' המחיר ). לטובת סעיף זה, הנח שכל הנקודות במפת הכבישים הן נקודות ב והמרחק בין זוג נקודות הוא המרחק האוקלידי.**

היוריסטיקה הינה קבילה.  
יוריסטיקה קבילה היא יוריסטיקה עבורה מתקיים: .  
  
הערך היוריסטי של מצב s כלשהו בגרף החיפוש, h(s), מוגדר להיות סכום קשתות ה MST של גרף G כאשר הצמתים ב G הם הדירות שעוד לא בוקרו במצב s, ומשקלה של כל קשת ב G מוגדרת להיות במרחק האווירי בין שני צמתים. ע"מ להגיע מ s למצב מטרה, יהיה עלינו אכן לעבור על כל הצמתים בגרף G. הערך היוריסטי של s הוא הערכה אופטימית של מחיר המעבר בכל הדירות, מאחר ולא לקחנו בחשבון את המעבר במעבדות (שיוסיף בוודאות למרחק ע"פ אי שוויון המשולש), ומקבילות יוריסטיקת המרחק האווירי (בפועל המרחק האמיתי יהיה גדול\שווה לו). אם כך, קיבלנו ש h(s) <= h\*(s) כנדרש.

1. **הגרף שנוצר עבור small\_mda\_problem עם יוריסטיקת MDAMSTAirDistHeuristic:**

הסבר הגרף:  
תיאור מספר הצמתים שפותחו ואיכות הפתרון של ריצת אלגוריתם A\* המשתמש ביוריסטיקת MDAMSTAirDistHeuristic עבור ערכי w שונים בטווח (0.5, 1) על בעיית mda\_problem.  
ניתן לראות שוב את ביטוי כלל האצבע האומר כי ככל ש w קטן יותר כך הפתרון איכותי יותר ומספר הפיתוחים גדול יותר בצורה מגמתית, אך הפעם ניתן לראות כי ב w = 0.8 יש זינוק במחיר הפתרון,  
והוא זהה בתחום w > 0.8. כמו כן ניתן לראות כי מספר הצמתים המפותחים יורד בצורה יחסית מתונה עם עליית ערך w, והוא מתייצב מאיזור w = 0.7.  
  
ערך ה w שנבחר (שיתן את זמן הריצה הטוב ביותר מבין ערכי w שמחזירים פתרון אופטימלי):  
~0.655, ערך ה w של הנקודה הירוקה.  
זהו ערך ה w המקסימאלי בו הפתרון המוחזר עדיין אופטימלי, כך שנקבל עבור w הנ"ל את הפתרון האופטימלי בזמן היעיל ביותר עבור ערכי w השונים.

**הגרף שנוצר עבור small\_mda\_problem עם יוריסטיקת MDASumAirDistHeuristic:**הסבר הגרף:

תיאור מספר הצמתים שפותחו ואיכות הפתרון של ריצת אלגוריתם A\* המשתמש ביוריסטיקת MDASumAirDistHeuristic עבור ערכי w שונים בטווח (0.5, 1) על בעיית mda\_problem.  
ניתן לראות שוב את ביטוי כלל האצבע האומר כי ככל ש w קטן יותר כך הפתרון איכותי יותר ומספר הפיתוחים גדול יותר בצורה מגמתית. ניתן לראות כי אחרי הנקודה הירוקה מספר הצמתים המפותחים משתטח וקרוב מאוד ל 0. מחיר הפתרון במגמת עלייה אך ישנה צניחה משמעותית עבור w מסוים.

ערך ה w שנבחר (שיתן את זמן הריצה הטוב ביותר מבין ערכי w שמחזירים פתרון אופטימלי):  
~0.716, ערך ה w של הנקודה הירוקה.

נבחר בערך ה w הנותן לנו פתרון אופטימלי ובנוסף גם מפתח כמה שפחות צמתים. מכיוון שגרף הצמתים המפותחים יורד ככל שערכי ה w גדלים והתחום של ה w בו הפתרון הוא אופטימלי הוא , נבחר .

## חלק ז': מימוש והשוואת פונקציות עלות שונות

מסתבר שהמקרר באמבולנס אינו אידיאלי עבור אחסון ממושך של הדגימות. ככל שעובר יותר זמן שבו הבדיקות

מאוחסנות באמבולנס, כך יורדת אםקטיביות ואמינות הבדיקה. פונקציית העלות מתארת את המדד

הנ"ל: .

1. **הוכח\הפרך: היוריסטיקה MDAMaxAirDistHeuristic הינה קבילה (עבור פונ' המחיר ). לטובת סעיף זה, הנח שכל הנקודות במפת הכבישים הן נקודות ב והמרחק בין זוג נקודות הוא המרחק האוקלידי.**  
   היוריסטיקה אינה קבילה.  
   יוריסטיקה קבילה היא יוריסטיקה עבורה מתקיים: , כלומר היא מעריכה "אופטימית" את מחיר המסלול מכל מצב s למצב מטרה תחת פונקציית העלות .

דוגמא נגדית:  
נסתכל על גרף בו יש מעבדה במרחק 1 מכל דירה, והמרחק האווירי בין כל שתי דירות הוא בדיוק 2\*.  
יהי s מצב התחלתי. נסתכל על מסלול מ s למצב מטרה שמבקר במעבדה אחרי כל ביקור בדירה, ונקבל שמחירו הוא .  
הערך היוריסטי ש s יקבל הוא המרחק האווירי המקסימאלי בין שתי דירות בגרף שעוד לא בוקרו, כלומר 2\* h(s) =.  
לעומת זאת המסלול הנ"ל הוא מסלול מ s למצב מטרה שעלותו היא .  
אם כך קיים לפחות מצב מטרה אחד שעלות מסלולו מ s קטנה מהערך היוריסטי ש s תקבל,  
ולכן בהכרח h(s) > h\*(s) והיוריסטיקה אינה קבילה ע"פ הגדרה.

1. **הוכח\הפרך: היוריסטיקה MDASumAirDistHeuristic הינה קבילה (עבור פונ' המחיר ). לטובת סעיף זה, הנח שכל הנקודות במפת הכבישים הן נקודות ב והמרחק בין זוג נקודות הוא המרחק האוקלידי.**  
   היוריסטיקה אינה קבילה.  
   יוריסטיקה קבילה היא יוריסטיקה עבורה מתקיים: .  
     
   דוגמא נגדית:  
   נסתכל על הגרף בדוגמא בשאלה 26.  
   הפעם הערך היוריסטי ש s תקבל הוא (k-1)\*2\* h(s) =   
   (k מספר הדירות שדווחו). המחיר האמיתי של המסלול מ s למצב המטרה במסלול הנתון (ביקור במעבדה אחרי כל ביקור בדירה) הוא עדיין .  
   קיים לפחות מצב מטרה אחד שעלות מסלולו מ s קטנה מהערך היוריסטי של s, ולכן בהכרח h(s) > h\*(s) והיוריסטיקה אינה קבילה ע"פ הגדרה.
2. **הוכח\הפרך: היוריסטיקה MDAMSTAirDistHeuristic הינה קבילה (עבור פונ' המחיר ). לטובת סעיף זה, הנח שכל הנקודות במפת הכבישים הן נקודות ב והמרחק בין זוג נקודות הוא המרחק האוקלידי.**  
   היוריסטיקה אינה קבילה.  
   יוריסטיקה קבילה היא יוריסטיקה עבורה מתקיים: .  
     
   דוגמא נגדית:  
   נסתכל על הגרף בדוגמא בשאלה 26.  
   מאחר והמרחק האווירי בין כל שתי דירות זהה, וערכו 2\*,  
   MST מתאים לדירות שעוד לא בוקרו (ע"פ הגדרת היורסטיקה, כאשר המשקלות הם מרחק אווירי) הוא "שרוך" המכיל את כל הצמתים בגרף.  
   מחיר עץ כזה הוא (k-1)\*2\* h(s) =.  
   המחיר האמיתי של המסלול מ s למצב המטרה במסלול של ביקור במעבדה אחרי כל ביקור בדירה הוא .  
   קיים לפחות מצב מטרה אחד שעלות מסלולו מ s קטנה מהערך היוריסטי של s, ולכן בהכרח h(s) > h\*(s) והיוריסטיקה אינה קבילה ע"פ הגדרה.
3. רטוב.
4. **הוכח\הפרך: היוריסטיקה MDATestsTravelTimeToNearestLabHeuristic הינה קבילה (עבור פונ' המחיר ). לטובת סעיף זה, הנח שכל הנקודות במפת הכבישים הן נקודות ב והמרחק בין זוג נקודות הוא המרחק האוקלידי.**

היוריסטיקה הינה קבילה.  
יוריסטיקה קבילה היא יוריסטיקה עבורה מתקיים: .

יהי :

אם : אז לא נותרו דירות לבקר בהן ולכן .

אם : לכל דירה שנותר לנו לבקר נסמן ו את המרחק למעבדה הקרובה ביותר מהדירה ה-i והמרחק שהאמבולנס נסע עם הבדיקות בפתרון האופטימלי מהדירה ה-i בהתאמה.

לכל דירה מהדירות שנותר לנו לבדוק, ,  *ומכיוון ש*

*נקבל שהיוריסיקה שלנו נותנת הערכה אופטימית לפתרון האופטימלי ולכן היא קבילה.*

**מה אפשר לומר על היחס בין MDATestsTravelTimeToNearestLabHeuristic ל ?**היחס הוא יחס של חסם תחתון.  
 לא ניתן "לטייל" פחות עם הבדיקות שנאספו מאשר ההערכה שהיוריסטיקה  
MDATestsTravelTimeToNearestLabHeuristic מציעה לנו, כיוון שהיא מציעה לנסוע ישירות למעבדה הכי קרובה וזהו המקרה הכי זול.

1. **עלויות הפתרון של תוצאת הריצה:**  
    **הדגישו איך רואים בתוצאות שהפתרון המתקבל אכן ממזער את המדד הרלוונטי בהתאם לפונקציית העלות שהופעלה:**  
   המרחק שהבדיקות עברו בפתרון עם היוריסטיקה הנ"ל קטן משמעותית מהפתרונות הקודמים בהם פתרנו את הבעיה ע"פ מדד המרחק.

MDA(moderate\_MDA(8):TestsTravelDistance) A\* (h=MDA-TimeObjectiveSumOfMinAirDistFromLab, w=0.500) time: 32.43 #dev: 29180 |space|: 41799 total\_g\_cost: 104387.48471 total\_cost: MDACost(dist= 123573.833m, tests-travel= 104387.485m) |path|: 18

בפתרון זה קיבלנו לעומת שלושת הפתרונות הקודמים בהם קיבלנו .

### שילוב בין שני המדדים

לצורך הסעיף היבש הבא, נגדיר את הפעולה הכללית שמקבלת מרחב ומגדירה מרחב חדש באופן הבא:

* כאשר

אלג׳ מריץ על המרחב עם פונק׳ העלות הבאה:

אם בסיום ריצת נמצא פתרון, מוחזר המצב הסופי (במרחב ), שהינו למעשה מסלול – סדרה של מצבים במרחב המקורי (באותה התצורה שמוחזר מסלול ע״י ריצת על המרחב המקורי ). בסעיפים היבשים בחלק זה, הנח שכל הנקודות במפת הכבישים הן נקודות ב- והמרחק בין זוג נק׳ הוא המרחק האוקלידי. אם אתם מספקים דוגמא נגדית, היא צריכה להיות קטנה ככל הניתן.

1. **הוכח\הפרך: אם קיים פתרון במרחב (המקורי), אלגוריתם בהכרח מחזיר פתרון.**  
   הוכחה:

נניח וקיים פתרון במרחב המקורי המורכב מהמסלול p = .  
מסלול זה מקביל למסלול .... P = שמסתיים במצב מקבל במרחב .  
נסמן את המצב הסופי ב P s =. ע"פ הגדרת הפתרון במרחב המקורי, s מקיים כי ערך פונקציית העלות שלו, , היא בעלת הערך המינימלי מבין מצבי המטרה s1 במרחב המקיימים .

מריץ על המרחב , ולכן האלגוריתם יעצור ויחזיר פתרון כאשר צומת בעל מחיר מסלול מינימלי הינו צומת מטרה, כלומר כאשר הוא מוציא מ OPEN צומת שהיא מצב מטרה.  
  
מכך שיש מסלול P של פתרון המקיים את התנאי על מובטח לנו שתמיד יהיה ב OPEN צומת שמחירו קטן מ (s מקיים c. מאחר והמחיר של צומת ע"פ c הולך וגדל ככל שנעמיק בגרף, והתנאי מתקיים עבור s, כל מצב על המסלול P מקיים גם הוא את התנאי ולכן מחירו שונה מ ).  
אם כך, בכל שלב של ריצת האלגוריתם קיים ב OPEN מצב שניתן להמשיך ולפתח ולהגיע למצב מטרה (לכל הפחות אחד מהצמתים במסלול P נמצא ב OPEN), ולכן ניתן להגיע לשלב באלגוריתם בו נוציא מ OPEN צומת שהוא מצב מטרה ( מצב מטרה s, אלא אם כן נמצא פתרון אחר ואז מש"ל).  
קיבלנו שאם קיים פתרון במרחב המקורי, אלגוריתם בהכרח מחזיר פתרון.

1. **הוכח/הפרך: אם אלג׳ מחזיר פתרון אז הפתרון המוחזר בהכרח אופטימלי ע״פ הקריטריון המשולב שהוגדר מעלה.**

הוכחה:

ע"פ אופן הגדרת ,  
ישנו לפחות מסלול אחד בגרף p שכל צמתיו מקיימים את התנאי לכל (פונקציית העלות גדלה ככל שנתקדם ב p ומאחר והיא מקיימת את התנאי עבור המסלול כולו היא מתקיימת לכל אחד מהצמתים במסלול).  
ע"פ הגדרת פונקציית המחיר של האלגוריתם, המחיר של מסלולים המקיימים את התנאי בהכרח נמוך משל מסלולים שאינם מקיימים את התנאי (כיון שהוא יגיע ל ).   
לא ייתכן שנוציא מצב מ OPEN שאינו מקיים את התנאי כיון שמחירו יהיה אינסוף ובהכרח קיים ב OPEN קודקוד כלשהו השייך למסלול p המקיים את התנאי ולכן מרחקו יהיה קטן מ , ו OPEN תמיד בוחר את המינימלי.  
באופן דומה, מכל הצמתים מקיימים את התנאי הצומת שנבחר לפתח (כלומר הצומת שנוציא מ OPEN) הוא כאמור הצומת בעל ערך המחיר המינימלי, מאחר ומדובר על צמתים המקיימים את התנאי, הצומת שנוציא הוא בעל המינימלי מכל הצמתים המקיימים את התנאי הנמצאים ב OPEN.  
נובע שאם הוצאנו צומת מ OPEN והוא מצב מטרה, זה אומר שהוא מצב המקיים המקיימים את התנאי , וכי הוא מצב מטרה שערך ה שלו מינימלי מבין כל מצבי המטרה האפשריים הקיימים ב OPEN וע"פ קבילות UCS המשמעות היא שמחירו מינימלי מבין כל מצבי המטרה p1 המקיימים את התנאי , כאשר פונקציית המחיר היא .  
זוהי בדיוק ההגדרה של מסלול אופטימלי ע"פ הקריטריון המשולב.

עתה נציע את אלג׳ שפועל באופן הבא:

1. *הרץ (עם היוריסטיקה קבילה) על המרחב עם פונק׳ העלות .*
2. *שמור את עלות הפתרון המוחזר במשתנה .*
3. *הרץ (עם היוריסטיקה קבילה) על המרחב עם פונק׳ העלות . במהלך הריצה, סכום בצמתי עץ החיפוש גם את העלות בשדה נפרד. במהלך הריצה, מיד לאחר יצירת צומת חיפוש חדש, הוסף את הבדיקה הבאה: אם העלות dist שלו גדולה מ- , מחק את הצומת הזה ואל תוסיף אותו ל- open.*
4. רטוב+יבש.  
   **תוצאת הריצה:**

[Grab your reader’s attention with a great quote from the document or use this space to emphasize a key point. To place this text box anywhere on the page, just drag it.]

**השוואה לתוצאות הריצה מסעיפים קודמים (על אותה הבעיה עם שתי פונק׳ עלות השונות):**טבלה...

1. **הוכח/הפרך: אם קיים פתרון במרחב, אלג׳ בהכרח מחזיר פתרון.  
   טיפ: כדי לקבל קצת יותר אינטואיציה, אתם יכולים להריץ את הדוגמא מסעיף קודם עם ערכי שונים.**

נפריך באמצעות דוגמא נגדית:

נתאר גרף שצמתיו ב הם: apar1.location = (0,100) , apar2.location(0,200) ,

lab1.location = (0,20), lab2.location=(0,300) , initial.location=(0,0).

כאשר המצב ההתחלתי שלנו הוא initial.

נריץ את אלגוריתם עם :

*- נריץ את אלגורית עם יוריסטיקה קבילה על המרחב המתואר עם פונ' עלות*

*- נשמור את המשתנה :*  עם המסלול האופטימלי *עבור המרחב המתואר*

*- נריץ שוב עם יוריסטיקה קבילה עם פונ' העלות ובמהלך הריצה נסכום בצמתי עץ החיפוש גם את בשדה נפרד.אם העלות dist גדולה מ נמחק את הצומת ולא נוסיף אותו לopen .*

*המסלול האופטימלי עבור יוריסטיקה קבילה עבור הוא:*

*מתקבל שעבור מסלול זה*

*אך עבור האפסילון הנתון(0.5) הצומת lab2 לא יוכנס לפיתרון האופטימלי והאלגוריתם לא ימצא את המסלול האופטימלי, ובפרט, לא ימצא אף פתרון. (הערה גם בתרגיל שלנו עבור לא מצאנו פתרון בעוד שאנו יודעים שלבעיה הזו קיים פתרון)*

1. **הוכח/הפרך: אם אלג׳ מחזיר פתרון אז הפתרון המוחזר בהכרח אופטימלי ע״פ הקריטריון המשולב שהוגדר מעלה.**

הטענה נכונה.

על פי אופן הפעלת האלגוריתם הפתרון המוחזר הוא פתרון על פי יוריסטיקה קבילה עם פונק' עלות ולכן הפתרון יהיה אופטימלי עבור מדד זה. ובנוסף, הפתרון בוודאות מקיים

כי צמתים שלא מקיימים את התנאי לא יכנסו לopen ולכן לא יפותחו. לכן הפתרון המוחזר הוא בהכרח אופטימלי ע"פ הקריטריון המשולב שהוגדר למעלה.

1. **יתרון של ע״פ במובנים של זמני ריצה:** שומר פחות צמתים ב open מ , ברגע שהוא מגלה שעלות ה dist של צומת מסוים לא עומדת בתנאי הוא מוחק אותה ולא מוסיף ל open לעומת שכן יוסיף צמתים כאלו ל open אך יגדיר את מחירם להיות אינסוף. צמתים כאלו מן הסתם לא יפותחו, אך הם מעמיסים על open, ולכן ב  ה open יהיה קטן יותר והפעולות עליו יעילות יותר.

## חלק ח: מימוש האלג׳ A\* והרצתו

1. רטוב.
2. רטוב + יבש.  
   **תוצאת הריצה:**

MDA(small\_MDA(5):Distance) A\* (h=MDA-MST-AirDist, w=0.500) time: 11.15 #dev: 575 |space|: 947 total\_g\_cost: 31528.65909 total\_cost: MDACost(dist= 31528.659m, tests-travel= 52112.429m) |path|: 8

MDA(small\_MDA(5):Distance) A\*eps (h=MDA-MST-AirDist, w=0.500) time: 56.12 #dev: 564 |space|: 933 total\_g\_cost: 31528.65909 total\_cost: MDACost(dist= 31528.659m, tests-travel= 52112.429m) |path|: 8

**האם חסכנו בפיתוחים? אם כן, בכמה? הסבירו למה בכלל ציפינו מראש ש- A\* יוכל לחסוך במס׳ הפיתוחים בתצורה שבה הרצנו אותו.**כן, בריצה של אלגוריתם A\*eps חסכנו פיתוח של 11 צמתים לעומת ריצה עם אלגוריתם חיפוש A\*.  
  
אלגוריתם A\*eps יוצר רשימת את רשימת ה best\_nodes שלו באופן שונה מאלגוריתם A\*.  
רשימת ה focal מחזיקה צמתים שערך ה f שלהם מינימלי עד כדי כפל ב (1 + eps), כלומר ערך ה f שלהם אינו המינימלי ממש, אבל הוא "מספיק טוב" כדי לבחון אפשרות שמחיר המסלול שנפתח ממנו יהיה זול.  
מתוך הרשימה הזאת, נבחר את הצומת בעלת הערך היוריסטי הנמוך ביותר, כלומר הצומת שמוערך על ידי h כצומת הכי קרוב למטרה.

ערך f של צומת תלוי גם ב g וגם ב h. H היא הערכה אופטימית בלבד, ולכן על אף שצומת מקבל ערך f מינימלי מכל הצמתים ב OPEN לא מובטח לנו שאכן הוא באמת זה שיתן את המסלול האופטימלי (ייתכן שהיוריסטיקה עבורו הייתה אופטימית מדי, והחמירה עם צומת אחר). כיון שאנחנו כן נותנים משקל ליוריסטיקה, כן הגבלנו את ערך ה f של הצמתים ב focal להיות מינימלי עד כדי כפל ב (1 + eps) ובכך כן ניצלנו את הידע שהיוריסטיקה נותנת לנו (כמו כן גם הבחירה מתוך focal היא של הצומת בעל הערך היוריסטי המינימלי).

מצד שני איפשרנו גמישות כדי לכסות על מקרים בהם היא אינה מדויקת.

גמישות זו מאפשרת לנו להגיע לפתרון אופטימלי תוך פיתוח של פחות צמתים, כיון שנתנו "הזדמנות" לצמתים שכן הובילו למסלולים אופטימליים אך לא בהכרח נראו הכי מבטיחים, בשל אי הדיוק של f, להיות מפותחים באיטרציות מוקדמות יותר של האלגוריתם.

## חלק ט' – מימוש האלג׳ Anytime A\* והרצתו

1. רטוב.
2. רטוב + יבש.  
   **תוצאות הריצה:**

MDA(moderate\_MDA(8):Distance) Anytime-A\* (h=MDA-MST-AirDist, w=0.800) time: 2.39 #dev: 1027 |space|: 740 total\_g\_cost: 64055.65000 total\_cost: MDACost(dist= 64055.650m, tests-travel= 131870.337m) |path|: 13

**הסבירו איך עזר לנו להריץ את הווריאציה הזו של Anytime A\* במקרה זה. מה בעצם קיבלנו? שימו לב לגודל הבעיה אותה פתרנו. חזרו לחלק א׳ והיזכרו כמה זמן ייקח למחשב בודד לעבור על כל הסידורים האפשריים.**הרצת אלגוריתם Anytime A\* מצאה פתרון שמחירו 64055.65000 בזמן 2.39 שניות.  
בשאלה 17 ו 23, הרצנו אלגוריתם A\* עם היורסטיקות קבילות שונות, כאשר מחיר הפתרון האופטמילי שנמצא הוא 58254.18667 בזמן ~50 שניות.  
קיבלנו זמן ריצה פי 20 טוב יותר, לעומת הרעה של 10% בלבד באיכות הפתרון.

גודל הבעיה אותה פתרנו הוא 13, סידור של 4 מעבדות ו 8 דירות.  
בחלק א ראינו כי למחשב הבוחן 230 סידורים בשנייה יקח 5.799 שניות לחשב