**תרגיל בית 1: שימוש באלגוריתמי חיפוש היוריסטיים לתכנון מסלולי חלוקה אופטימליים**

## חלק א' – מבוא והנחיות (3 נק׳ יבש + 3 נק' בונוס)

1. יבש (1 נק'): מצאו ביטוי מתמטי עבור מספר המסלולים האפשריים תחת ההנחות הללו. על הביטוי להיות תלוי בפרמטרים בלבד. אנו מצפים לביטוי מתמטי סגור וללא סכומים (סיגמא). הביטוי צריך להיות כתוב בשפה מתמטית פורמלית ללא שימוש בסימוני עזר וללא תיאורים מילוליים. לא יינתנו נקודות עבור תשובות שאינן מדויקות. תנו כותרת קצרה עבור כל אחד מהמרכיבים הכפליים בביטוי.

**פתרון:**

מתוך מענה שקיבלתי מהצוות במייל, אניח שהמפה קליקה (כל הדירות והמעבדות קשורות) ואז מספר המסלולים האפשריים הינו מספר סידורי הבניינים שהוא עוצר בהן: כאשר הביטוי הראשון מתייחס למספר סידורי הדירות שיעבור בהן (סידור עצמים עם סדר בלי חזרות), הביטוי השני מתייחס למעבדה (סידור עצמים עם סדר ועם חזרות) (1 מתייחס לאפשרות שיחליט לא לעבור באף מעבדה ו-m מתייחס לm האפשרויות שיש לו לבחור במעבדה לבקר בה), והביטוי האחרון מתייחס למספר האפשרויות לבחירת המעבדה האחרונה שיגמור בה.

1. יבש בונוס (3 נק'): הנחות חדשות: נתרכז אך ורק בכל המסלולים מהסוג הבא (ונניח שכולם חוקיים): עוברים בכל דירה פעם אחת בדיוק, חייבים לסיים במעבדה (אחרי הביקור האחרון בדירה עוברים במעבדה אחת בדיוק), לא חייבים לבקר בכל המעבדות (ייתכן שקיימות מעבדות שלא נעבור בהן), ניתן לבקר במעבדה רק אם (א) טרם עברנו בה או (ב) ביקרנו בדירה ממש לפני הביקור במעבדה זו, בתחילת המסלול (לפני הביקור הראשון בדירה כלשהי) ובין זוג ביקורים רצוף בדירות ניתן לבקר ב- 0 או יותר מעבדות – ייתכן רצף של ביקורים במעבדות בלבד בתנאי שמתקיימים כל התנאים הקודמים. מצאו ביטוי מתמטי עבור מספר המסלולים האפשריים תחת ההנחות הללו. על הביטוי להיות תלוי בפרמטרים בלבד. הביטוי יכול להכיל סכומים (סיגמא) וביטויים קומבינטוריים מוכרים אחרים. הביטוי צריך להיות כתוב בשפה מתמטית פורמלית ללא שימוש בסימוני עזר וללא תיאורים מילוליים. לא יינתנו נקודות עבור תשובות שאינן מדויקות. תנו כותרת קצרה עבור כל אחד מהמרכיבים בביטוי (כפליים/חיבוריים/משתני סכום וכו'). הסבירו בקצרה (עד 3 שורות).

**פתרון:**

נפריד למקרים זרים ונסכום – המקרה של סעיף 1 (כתוב בהתחלה) והמקרה שישנם מעבדות שבוקרו ללא מבחנות. במקרה השני נפריד למקרים לפי מספר הביקורים החוזרים למעבדות אלה r ולמספר הדירות d שבוקרו ללא מעבדה אחריהם. כעת נפתור בעזרת הכלה והפרדה (הסכימה לפי j).

1. יבש (2 נק׳): מלאו את הטבלה הבאה. השתמשו בנוסחה שמצאתם בסעיף 1. הזינו את מספר הפרמוטציות האפשריות עבור ערכי המופיעים בטבלה. נניח שמחשב יחיד יכול לבחון מסלולים בשנייה (הסבר הנחה: המחשב יכול לעשות פעולות בסיסיות בשנייה וצריך פעולות בסיסיות ע"מ לבחון מסלול בודד). מלאו בעמודה האחרונה כמה זמן ייקח למחשב כזה לבדוק כל אחד מהמסלולים (לפי היחידות המפורטות).

**פתרון:**

|  |  |  |  |
| --- | --- | --- | --- |
| Estimated calculation time |  |  |  |
| ~18.48 [secs] | ~22.04 × 10^6 | 2 | 7 |
| ~3.845 [mins] | ~24.77 × 10^7 | 3 | 7 |
| ~2.256 [hours] | ~79.27 × 10^8 | 3 | 8 |
| ~19.56 [hours] | ~63.00 × 10^9 | 4 | 8 |
| ~3.691 [days] | ~28.54 × 10^10 | 3 | 9 |
| ~5.332 [months] | ~11.42 × 10^12 | 3 | 10 |
| ~21.05 [years] | ~50.23 × 10^13 | 3 | 11 |
| ~1.083 [thousand years] | ~24.11 × 10^15 | 3 | 12 |
| ~22.41 [thousand years] | ~46.78 × 10^16 | 4 | 12 |
| ~1.548 [million years] | ~30.41 × 10^18 | 4 | 13 |

## חלק ג' – הגדרת מרחב החיפוש של בעיית מד״א (6 נק׳ יבש)

### תרגילים

לטובת הסעיפים בחלק זה הנח שלאו דווקא קיים פתרון ישיג במרחב.

1. יבש (1 נק׳): מהם ערכי הקיצון (המקסימלי והמינימלי) האפשריים של דרגת היציאה במרחב החיפוש? ספקו ביטוי מתמטי כפונק' של הפרמטרים של השאלה בלבד. נמקו בקצרה (שורה אחת לכל מקרה).

**פתרון:**

הערך המינימלי לדרגת היציאה הוא 0, אם אין מספיק מטושים לכל הדיירים והאוטובוס נתקע במעבדה. הערך המקסימלי הוא m+k במקרה שהאמבולנס במצב ההחתלה עם מספיק מטושים לכולם.

1. יבש (1 נק׳): האם ייתכנו מעגלים במרחב המצבים שלנו? אם כן תנו דוגמה למעגל כזה, אחרת נמקו. (עד 5 שורות).

**פתרון:**

לא יתכנו מעגלים במרחב המצבים כי בכל מעבר ממצב למצב אנו מבקרים בדירה (שאז הדירה אף פעם לא תחזור להיות ריקה ולכן המצב לא ניתן לחזרה כי הדירה תמיד תמצא בtaken או transferred) או מבקרים במעבדה שאז אנחנו מוסיפים מבחנות למעבדה (ואין דרך להוציאם) או שמספר המטושים עולה (ואי אפשר להוריד מספר זה בלי לבקר בדירה, שכאמור יפסול כל מעגל).

1. יבש (1 נק׳): כמה מצבים יש במרחב זה (כפי שהוגדר)? האם כולם ישיגים (ציינו כן/לא)? נמקו (עד 3 שורות).

**פתרון:**

ישנם אינסוף מצבים במרחב, כיוון שבהגדרה הפורמלית של הבעיה כתוב שיש מצב אחד לכל מספר טבעי של מטושים זמינים באמבולנס (לכל מיקום אמבולנס, מצב דיירים, וכו'), ויש אינסוף מספרים טבעיים. כמובן שלא כל המצבים ישיגים כי יש מספר מוגבל של מטושים במרחב סה"כ.

1. יבש (1 נק׳): האם ייתכנו בורות ישיגים מהמצב ההתחלתי שאינם מצבי מטרה במרחב המצבים? אם כן – איך זה ייתכן? אם לא – למה? (נימוק לכל היותר שורה אחת)

**פתרון:**

ייתכנו, במרחב שאין מספיק מטושים לכל הדיירים, שאז נתקע באיזשהו שלב ללא מטושים לאבחונים.

1. יבש (1 נק׳): מהו טווח האורכים האפשריים של מסלולים במרחב ממצב התחלתי אל מצב סופי? (אורך מסלול = מס׳ הקשתות) (לכל היותר 7 שורות סה"כ).

**פתרון:**

מסלול מינימום באורך k+1 הוא המסלול (תחת ההנחה שיש מספיק מטושים לכך) שעובר מהמצב ההתחלתי דרך כל הדירות (K קשתות) ואז ישר למעבדה. כל מסלול קצר מזה לא מתחיל ממצב ההתחלה או לא מסיים במעבדה או לא עובר בכל הדירות ולכן אינו מסלול גמר חוקי. מסלול מקסימום באורך m+2k הוא המסלול שעובר קודם בכל המעבדות (M קשתות) ולאחר מכן עובר בכל הדירות (K קשתות) כשלאחר כל דירה (K קשתות) הוא מבקר במעבדה כלשהו. מסלול ארוך מזה יבקר שוב במצב ההתחלה או יבקר בדירה כלשהו פעמיים (שזה אסור) או יבקר במעבדה ישר לאחר ביקור אחר במעבדה, שאת זה ניתן רק לעשות פעם אחת לכל מעבדה (ואת זה עשינו במידה המקסימלי במסלול המתואר, ולכן יותר מזה אסור).

1. יבש (1 נק׳): הגדירו פורמלית ובצורה ישירה את פונקציית העוקב המתאימה לבעיה זו (ללא שימוש בקבוצת האופרטורים ).

שימו לב, אנו מצפים לביטוי מהצורה:

**פתרון:**

עם כאשר

## חלק ה' – אלגוריתם A\* (2 נק׳ יבש)

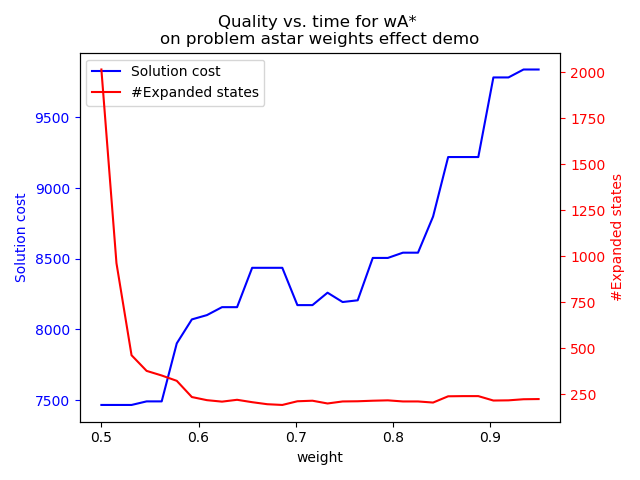
1. יבש (1 נק'): כתוב בדו״ח את מס׳ פיתוחי המצבים ה**יחסי** שחסכנו בריצה בסעיף קודם לעומת הריצה העיוורת (ההפרש חלקי מס׳ הפיתוחים בריצה **בלי** ההיוריסטיקה).

**פתרון:**

1. יבש (1 נק׳): צרפו לדו"ח את הגרף שנוצר בריצה מהסעיף הקודם. הסבירו את הגרף שהתקבל. ציינו אילו איזורים בגרף הם יותר כדאיים ואילו פחות – ציינו למה (עד 2 שורות). בכיתה למדתם כלל אצבע לפיו ״ככל ש- w קטן יותר כך הפתרון איכותי יותר ומס׳ הפיתוחים גדול יותר״. הכלל הנ״ל מצביע על מגמה כללית, אך איננו נכון באופן גורף (כלומר ייתכנו זוג ערכים עבורם הפתרון המתקבל עם פחות טוב מאשר הפתרון המתקבל עם ו/או מס׳ הפיתוחים עם גדול יותר ממס׳ הפיתוחים עם ). כיצד הכלל שהוזכר והדגש הנ״ל באים לידי ביטוי בתרשים שקיבלתם? (תשובה עד 4 שורות). על התרשים להראות כמו בדוגמה הזו (צורת העקומות עצמן עשויה להשתנות כמובן(:  
   A close up of a map

   Description automatically generated

**פתרון:**



כדאי לקחת את האזור בצד שמאל כאשר חשיבות מקום וזמן הריצה זניחה לעומת חשיבות איכות הפתרון, וכדאי לקחת את האזור בצד ימין למקרה ההפוך. אזור מפגש הגרפים בסביבות weight=0.57 שילוב טוב.

כלל האצבע בא לידי ביטוי בכך שגרף מחיר הפתרון בעיקר יורד (איכות פתרון עולה) עם w יותר גדול במגמה כללית ולהיפך עבור מספר הפיתוחים (בעיקר עולה עם משקל גבוה יותר), אך הדגש בא לידי ביטוי שאמירה זו איננה נכונה באופן מדויק – הגרפים אינם מונוטוניים, אלא רק קרובים לכך במבט מרחוק.

## חלק ו׳ – מימוש בעיית מד״א (15 נק׳ יבש)

1. שאלה יבש (2 נק'): בחלק ב' הגדרנו את מרחב מפת הכבישים, ובחלק ג' הגדרנו את מרחב מד"א. נסתכל על זוג של מצב ועוקב כלשהו שלו ­ במרחב מד"א. שימו לב כי ייתכן שלא קיימת קשת ישירה בין זוג הצמתים ל- במרחב מפות הכבישים. עם זאת, קיים מסלול כלשהו בניהם במרחב המפה. לכן, כל פעם שצריך לחשב את עלות האופרטור (כפי שהוגדר לעיל בחלק ג'), למעשה פותרים "בעיית ביניים" במרחב מפות הכבישים .  
   עקרונית, היה אפשר לשלב את שני המרחבים המדוברים למרחב-על אחד. במרחב-העל הנ"ל עבור מצב כלשהו, המיקום שלו היה יכול להיות כל נקודה ברשת הכבישים. בנוסף, היינו מוסיפים אופרטורים שמאפשרים מעבר למצב עוקב שבו רק המיקום של האמבולנס היה משתנה (לאחת מהנקודות על רשת הכבישים).  
   שאלה: מה יכול להיות חסרון בגישה שכזאת מבחינת יעילות הפתרון? על תשובתכם להתייחס לטכניקה ספציפית שהשתמשנו בה במימוש. תשובה עד 3 שורות.

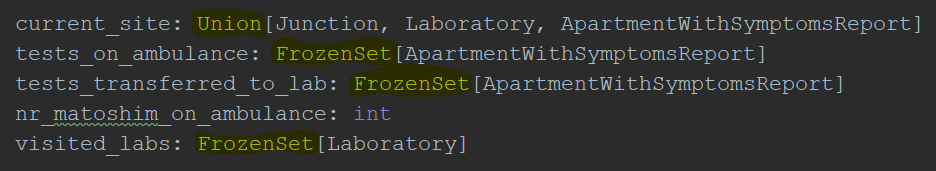
**פתרון:**

החיסרון בגישה זו הוא חוסר יעילות שבמקום להשתמש בפונקציה Cached\_map\_distance\_finder() שמחזירה לנו את המרחק במפה ב-O(1) מתוצאות חישוב קודמות שביצעו ושמרנו, כעת אנו נאלצים כל פעם לחשב מחדש מפני שכל פעם אנחנו פותרים בעיה אחרת עם מצבים אחרים ולכן לא ניתן לשמור בCache (עלות הזיכרון מאוד גדולה).

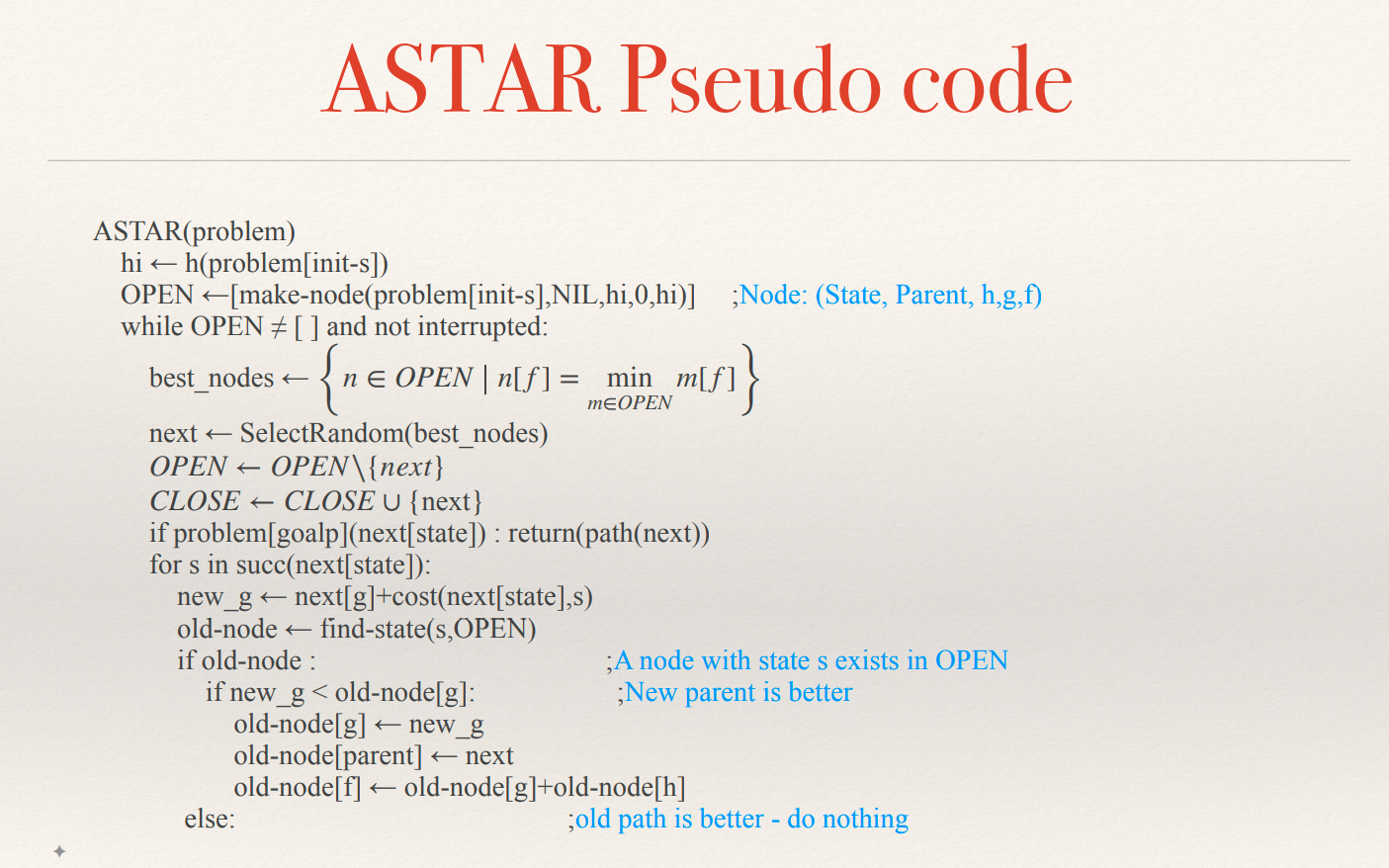
1. שאלה יבש (3 נק׳): בתכנות לפעמים אנחנו רוצים לכפות על מבני נתונים / טיפוסים מסוימים להיות immutable/frozen. הכוונה היא שאחרי יצירת אובייקט מטיפוס שכזה לא יהיה ניתן לשנותו. הצהרה על טיפוס כ״קפוא״ מגבילה אותנו, אך יחד עם זאת היא גם מגינה עלינו.  
   (i) העתק לדו״ח את שורת הקוד הרלוונטית שקובעת שאובייקטים מהטיפוס MDAState יהיו בלתי ניתנים לשינוי.  
   (ii) האם שורה זו מספיקה? מה עוד בקוד מבטיח שלא יהיה ניתן לשנות בטעות את האובייקט ו/או את המבנים שהוא מחזיק?  
   (iii) האם ייתכן באלג' A\* שנפגוש מצב בשנית גם לאחר שפיתחנו אותו? (אם כן – ציין את השורה באלג׳ A\* מההרצאה שבה זה קורה).  
   (iv) הסבר למה אנחנו רוצים לעשות זאת ספציפית עבור הטיפוס MDAState – תן דוגמא למימוש שגוי של המתודה expand\_state\_with\_costs במחלקה MDAProblem שממחיש את הצורך בטיפוסים ״קפואים״. על הבאג להיגרם מכך שבפיתון משתנה מחזיק בפועל מצביע לאובייקט ולא העתק שלו. יש להשתמש בתשובה מהסעיף הקודם כדי לענות על סעיף זה. הבאג צריך להיות שגיאה תמימה של מתכנת שלא רגיל לשפות תכנות בהן מתבצע copy-by-reference. תשובה ל- (iv) עד 5 שורות.

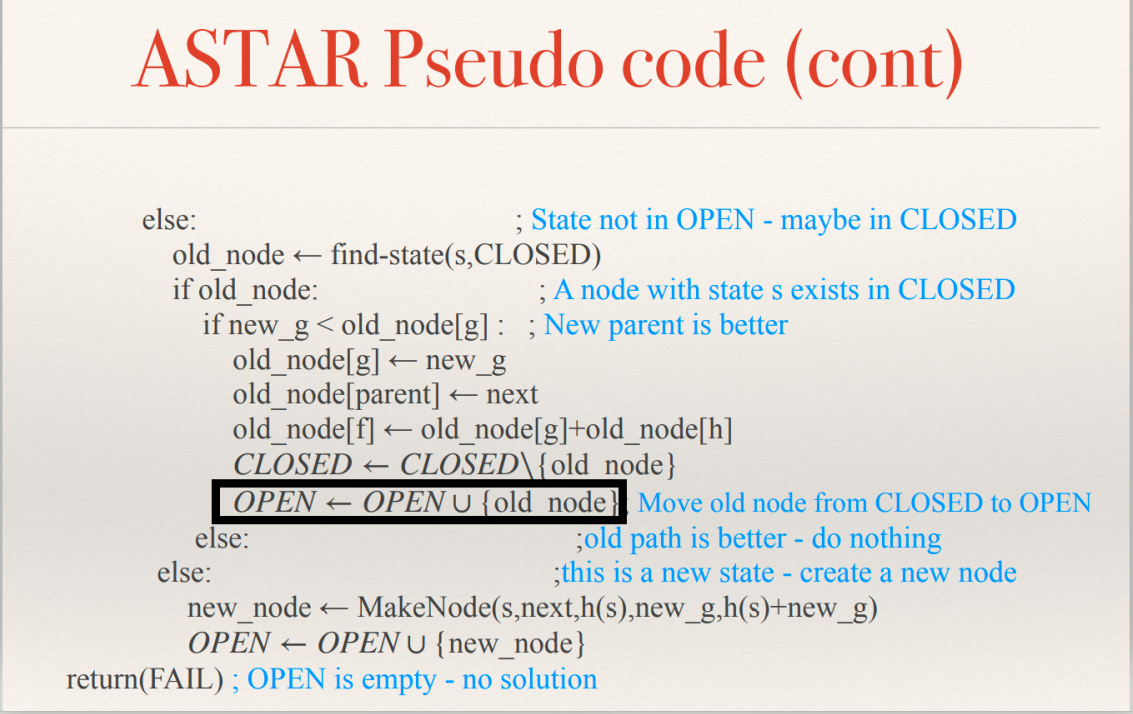
**פתרון:**

1. השורה היא ( צילום מסך ):
2. נשים לב שיש שימוש בפקודות FrozenSet + Union על FrozenSets שמחזירה כפלט FrozenSet שהוא טיפוס Immutable (מצורף צילום מסך עם הדגשות רלוונטיות ):

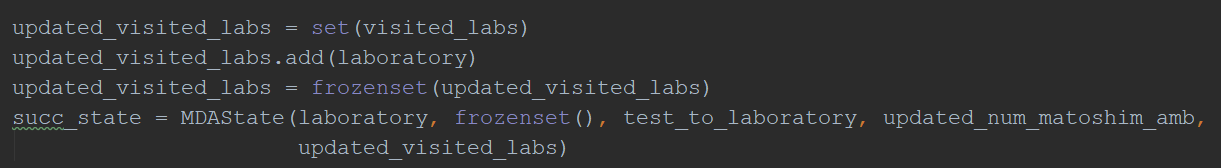


1. אכן בהסתכלות על הפסאודו קוד של אלגוריתם A\*, מההרצאה ניתן לראות שניתן לפגוש מצב בשנית לאחר שפיתחנו אותו נצרף צילום מסך שבו השורה הרלוונטית מסומנת:

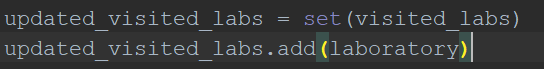




ניתן לראות שבשורה המסומנת בעצם אנחנו מחזרים Node קודם שהיה ב-Closed ומחזירים אותו ל-Open ובמקרה זה נפגוש אותו בשנית.

1. המימוש המקורי שלנו (שימוש ב- Frozenset) :

מימוש שגוי :



כלומר ניתן לראות שהשינוי הוא לא להשתמש ב-Frozenset, במצב זה בעצם נבצע עדכון של MDAState, ולא יצירה של מצב חדש כי המשתנה ה"חדש" שניצור עדיין יחזיק מצביע לאותו אובייקט שהיה בצומת המקורי, כלומר יצרנו בעצם תרחיש בו הצומת המקורי והצומת העוקב לו שניהם מצביעים על אותו Visited\_Labs למרות שלצומת הנוכחית לא היה אמור להתווסף מעבדה נוספת.

המצב בעייתי מאחר ויכול להיות תרחיש שנגיע לאותו מצב משני מצבים שונים וכאשר נגיע למצב בפעם השניה הוא יכיל שדות לא רלוונטיים (כי שינינו אותם בפעם הראשונה שהגענו אליו) והמידע בו יהיה שגוי וישפיע על התוצאה.

עתה, כדי להריץ את A\* על הבעיה, יש ראשית להגדיר (ולממש) היוריסטיקות עבור הבעיה.

1. יבש (4 נק׳): הוכח/הפרך: ההיוריסטיקה MDAMaxAirDistHeuristic הינה קבילה (עבור פונק׳ המחיר ). ראה בעמוד השני במסמך את ההערות המתייחסות לשאלות הוכח/הפרך.

**פתרון:**

היוריסטיקה קבילה, הוכחה:

יהי האמבולנס בצומת שנסמנה J ונניח ששאר הצמתים בהן הוא צריך לעבור הם הקבוצה Junc = {j1,j2,……,jn}.

נסמן את המרחק המקסימלי בין שני צמתים בקבוצה Junc על ידי L , לכן כפי שלמדנו h(J) = L ולכן אורך המסלול אותו האמבולנס צריך לעבור הוא גדול או שווה ל- L מכיוון ומרחק אוקלידי הוא המרחק המינימלי בין שני צמתים כלשהם ולכן אם קיים מסלול כלשהו בין שני צמתים הוא בהכרח לפחות בגודל L ולכן מתקיים : h(J) קטן שווה ל- h\* (J) , בנוסף בוודאות היוריסטיקה גדולה שווה לאפס מכיוון וכל הערכים של המרחקים הם אי שליליים ולכן לסיכום מתקיים :

1. יבש (4 נק׳): הוכח/הפרך: ההיוריסטיקה MDASumAirDistHeuristic הינה קבילה (עבור פונק׳ המחיר ). ראה בעמוד השני במסמך את ההערות המתייחסות לשאלות הוכח/הפרך.

**פתרון:**

היוריסטיקה אינה קבילה, נפריך באמצעות דוגמא נגדית:

לפי היוריסטיקה הנ"ל מתקיים שמצומת ההתחלה תחילה זזים שמאלה לצומת 2 ואז חוזרים ימינה לצומת 1 ולבסוף ממשיכים שמאלה לצומת 3 ובסה"כ המרחקים הם : L = 3 + 7 + 15 = 25 , לעומת זאת קיים מסלול קצר יותר כאשר קודם כל מצומת ההתחלה זזים ימינה ומשם שמאלה עד הסוף ובסה"כ המרחקים הם : L = 4 + 7 + 8 = 19 . לכן הראינו כי לא מתקיים התנאי והפרכנו את הטענה.

1. יבש (4 נק׳): הוכח/הפרך: ההיוריסטיקה MDAMSTAirDistHeuristic הינה קבילה (עבור פונק׳ המחיר ). ראה בעמוד השני במסמך את ההערות המתייחסות לשאלות הוכח/הפרך.

**פתרון:**

יהי J צומת כלשהו, יהי G גרף הצמתים הנותרים כפי שתואר בתרגיל, והמשקל של כל קשת הוא w(e) = dist(j1,j2) (מרחק הדרכים בין הצמתים הרלוונטיים לקשת) ויהי T עפ"מ של G כאשר המשקלים על הקשתות הם המרחקים האוויריים בין הצמתים. יהי מסלול אופטימלי (מבחינת מרחק) המתחיל ב ומסתיים בצומת מטרה (מעבדה) כלשהו m’ שהדירה שלפניו הוא (המסלול האופטימלי אינו עובר ממעבדה למעבדת הסוף כי זה יוסיף מרחק לערך המסלול כאשר היה יכול לסיים במעבדה הראשונה ולסיים, בסתירה לאופטימליות). נגדיר מסלול P' כמסלול P לאחר שמורידים ממנו את כל המעברים החוזרים בדירות (כלומר בכל דירה עוברים רק פעם אחת, וכל פעם שרוצים להגיע מדירה לדירה אחרת פשוט "נדלג אליה" בלי לעבור בדירות הביניים שכבר הופיעו במסלול מקודם) וגם הורדנו ממנו את המעבר האחרון למעבדה m’. את כל מעברי ה"דילוגים" האלה נמשקל לפי המרחק האווירי בין הדירות ונשים לב שסכום המרחק ב'P אינו עולה על סכום המרחקים בP (כי לפי אי"ש המשולש, מרחק אווירי בין 2 נקודות אינו גדול ממרחק שעובר דרך מיקומים נוספים בדרך, לפי מדד מרחק אוקלידי) (וכי הורדנו ממנו את המרחק האי-שלילי ). 'P קשיר (בתור מסלול המילטוני) וגרף התשתית שלו חסר מעגלים (עקב הדילוגים מעל צמתים שהיינו כבר בהם) ולכן הוא עץ פורש (כיוון שP עבר **בכל** הצמתים הנותרים ולכן גם P' (כי הורדנו רק **חזרות** על צמתים)), ומשקלו אינו קטן ממשקל T (לפי הגדרת עפ"מ כעף פורש **מינימום**). קיבלנו

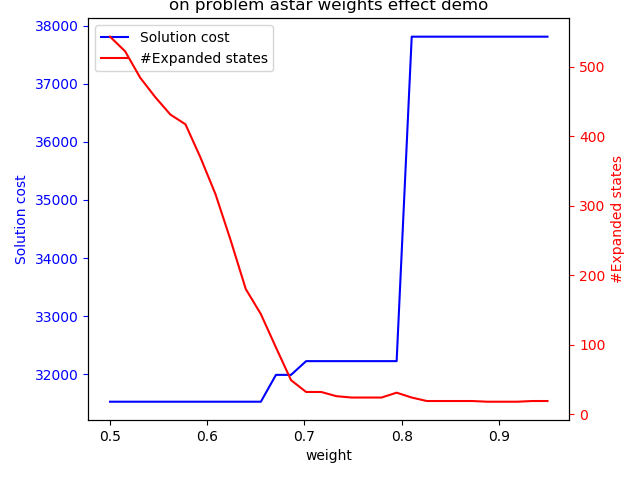
(הגדרת משקל גרף W הינו סכום כל משקלי הקשתות שבגרף) ולכן ההיוריסטיקה קבילה.

1. רטוב + יבש (1 נק׳): עתה נריץ את wA\* עם ערכי w שונים כדי לצייר גרף שמציג את מגמת מחיר הפתרון מגמת מס׳ הפיתוחים כאשר w משתנה בתחום . לצורך כך נשתמש בפונק׳ run\_astar\_for\_weights\_in\_range() שכבר מימשנו בשלבים מוקדמים. השלימו בקובץ main.py את הקוד תחת ההערה הרלוונטית לסעיף זה. צרפו את הגרף שנוצר לדו"ח. ציינו אילו איזורים בגרף הם יותר כדאיים ואילו פחות.

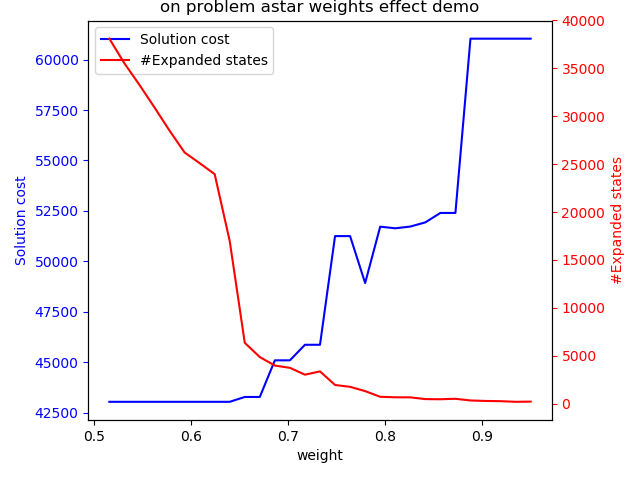
**פתרון:**

להלן מצורפים הגרפים שהתקבלו:

גרף ראשון:



גרף שני:



כפי שתואר לנו בסעיף קודם משמעות הגרפים היא :

העקומה הכחולה מתארת את טיב הפתרונות בציר ה-Y כפונקציה של המשקל.

העקומה האדומה מתארת את מספר המצבים שפותחו כפונקציה של המשקל.

נשים לב שערך המשקולת האופטימלי כתלות בעלות הוא בערך 0.67 – 0.68 , כאשר בערכים אלה מספר המצבים המפותחים הוא יחסית קטן, כאשר איכות הפתרון יחסית גבוהה והעלות של הפתרון יחסית נמוכה.

כאשר ערך המשקולת גדול יותר נוצר מצב שהעלות הנוספת משפרת במידה נמוכה יחסית את איכות הפתרון ולכן נוצר מצב "לא משתלם".

## חלק ז' – מימוש והשוואת פונק׳ עלות שונות (21 נק׳ יבש)

1. יבש (2 נק׳): סמן בכל אחד מהתאים כן/לא. האם כל אחת מההיוריסטיקות הנקובות הינה קבילה ביחס לפונק׳ המחיר הנקובה? (אין צורך בנימוק).

**פתרון:**

|  |  |  |  |
| --- | --- | --- | --- |
|  | MDAMaxAirDistHeuristic | MDASumAirDistHeuristic | MDAMSTAirDistHeuristic |
|  | לא | לא | לא |
|  | לא | לא | לא |

1. רטוב + יבש (0.5 נק' יבש): כעת נפתור את הבעיה עם פונק' העלות . השלימו בקובץ main.py את הקוד תחת ההערה הרלוונטית לסעיף זה. השוו כאן בדו"ח את התוצאות עם תוצאות מסעיפים קודמים של פתרון בעיה זו עם מדד המרחק כ- optimization objective. הראו בדו"ח איך רואים בתוצאות שהפתרון המתקבל אכן ממזער את המדד הרלוונטי בהתאם לפונק׳ העלות שהופעלה (אין צורך לצרף את כל הפלט עם המסלול, רק את העלויות).

**פתרון:**

התוצאה של סעיף זה אינו מנסה למזער את המרחק שנוסעים בו, אלא רק את המחיר הכולל. על כן, ניתן לראות בתוצאות שיש לו מרחק כולל יותר גדול (31923.809m בבעיה הקטנה למשל, במקום הערך האופטימלי שהוציאה הUCS על אותה בעיה 31528.659m). עם זאת, ניתן לראות שמתוך כל התוצאות הקודמות והנוכחי, הערך הכספי נמוך ביותר בתוצאה הזה (41.440NIS לעומת 49.107NIS בהרצת האלגוריתם UCS הנ"ל) ומתוך כך ניתן לראות שהפתרון המתקבל אכן ממזער במקרה זה את המחיר הכספי כיוון שהוא הקטן ביותר והאחרים לא ממוזערים למרות השימוש באלגוריתם הקביל UCS. ניתן גם כמובן להשוות את הg\_cost (לא מצורף כאן) לMDACost ולראות לאיזה ערך הוא מתאים, וכותרת מדד זה יהיה המדד שמוזער.

התוצאות הן:

MDA(small\_MDA(5):Distance)

UniformCost

MDACost(dist= 31528.659m, money= 49.107NIS, tests-travel= 52112.429m(

MDA(moderate\_MDA(8):Distance)

A\* (h=MDA-Max-AirDist, w=0.500)

MDACost(dist= 43082.235m, money= 95.715NIS, tests-travel= 177571.274m(

MDA(moderate\_MDA(8):Distance)

A\* (h=MDA-Sum-AirDist, w=0.500)

MDACost(dist= 43082.235m, money= 95.715NIS, tests-travel= 177571.274m)

MDA(small\_MDA(5):Monetary)

UniformCost

MDACost(dist= 31923.809m, money= 41.440NIS, tests-travel= 53317.118m)

MDA(moderate\_MDA(8):Monetary)

UniformCost

MDACost(dist= 55101.186m, money= 77.118NIS, tests-travel= 174498.879m(

1. יבש (2 נק׳): הוכח/הפרך: ההיוריסטיקה *MDATestsTravelTimeToNearestLabHeuristic* הינה קבילה עבור פונק׳ המחיר . ראה בעמוד השני במסמך את ההערות המתייחסות לשאלות הוכח/הפרך.

**פתרון:**

הוכחה: יהי מצב המתחיל במיקום ויהי מסלול פתרון אופטימלי לפי , נביט בתרומת מבחנת דייר יחיד למחיר: אבל שזהו בדיוק תרומת אותו המבחן לערך ההיוריסטיקה (מעבר ראשון-אי"ש המשולש המתקיים עבור מרחקים אוקלידיים במפה, מעבר שני עקב מינימליות המרחק של המעבדה הקרובה ביותר). זה מתקיים לכל מבחנה ולכן ערך ההיוריסטיקה=סכום התרומות של המבחנות להיוריסטיקה אינו גדול מערך הפתרון האופטימלי החל מ ולכן ההיוריסטיקה מוגדרת קבילה.

1. רטוב + יבש (0.5 נק' יבש): כעת נפתור את הבעיה עם פונק' העלות . השלימו בקובץ main.py את הקוד תחת ההערה הרלוונטית לסעיף זה. השוו כאן בדו"ח את התוצאות עם תוצאות מסעיפים קודמים של פתרון בעיה זו עם מדד המרחק כ- optimization objective. הראו בדו"ח איך רואים בתוצאות שהפתרון המתקבל אכן ממזער את המדד הרלוונטי בהתאם לפונק׳ העלות שהופעלה (אין צורך לצרף את כל הפלט עם המסלול, רק את העלויות).

**פתרון:**

התוצאה של סעיף זה אינו מנסה למזער את המרחק שנוסעים בו, אלא רק את סכום מרחקי נסיעות המבחנות. על כן, ניתן לראות בתוצאות שיש לו מרחק כולל יותר גדול (87850.369m בבעיה הגדולה למשל, במקום הערך האופטימלי שהוציאה הUCS על אותה בעיה 43082.235m). עם זאת, ניתן לראות שמתוך כל התוצאות הקודמות והנוכחי, מדד הtest travel הנמוך ביותר בתוצאה הזה (131265.153m לעומת 177571.274m בהרצת האלגוריתם UCS הנ"ל) ומתוך כך ניתן לראות שהפתרון המתקבל אכן ממזער במקרה זה את מדד מרחקי המבחנות כיוון שהוא הקטן ביותר והאחרים לא ממוזערים. ניתן גם כמובן להשוות את הg\_cost (לא מצורף כאן) לMDACost ולראות לאיזה ערך הוא מתאים, וכותרת מדד זה יהיה המדד שמוזער.

התוצאות הן:

MDA(small\_MDA(5):Distance)

UniformCost

MDACost(dist= 31528.659m, money= 49.107NIS, tests-travel= 52112.429m(

MDA(moderate\_MDA(8):Distance)

A\* (h=MDA-Max-AirDist, w=0.500)

MDACost(dist= 43082.235m, money= 95.715NIS, tests-travel= 177571.274m(

MDA(moderate\_MDA(8):Distance)

A\* (h=MDA-Sum-AirDist, w=0.500)

MDACost(dist= 43082.235m, money= 95.715NIS, tests-travel= 177571.274m)

MDA(moderate\_MDA(8):TestsTravelDistance(

A\* (h=MDA-TimeObjectiveSumOfMinAirDistFromLab, w=0.500)

MDACost(dist= 87850.369m, money=123.398NIS, tests-travel= 131265.153m)

### שילוב בין 2 מדדים

1. יבש (3 נק׳): הוכח/הפרך: אם קיים פתרון במרחב המקורי , אלג׳ בהכרח מחזיר פתרון. ראה בעמוד השני במסמך את ההערות המתייחסות לשאלות הוכח/הפרך.

**פתרון:**

הוכחה: נניח שקיים פתרון במרחב המקורי. כיוון שA\* אלגוריתם שלם (בעזרת היוריסטיקה קבילה), השלב הראשון באלגוריתם יסתיים והפתרון המוחזר אכן אינו קטן מערך הפתרון האופטימלי (בהקשר של מרחק נסיעה). יהי מסלול מהמצב ההתחלתי למצב סוף בעל ערך מרחק **אופטימלי**. המסלול הוא מסלול בעל אותו ערך מרחק אופטימלי (כאשר ערך מרחק מוגדר לפי סכום מרחק המסלול האחרון p כמו בתנאי המחיר של בתרגיל) ולכן ערך מרחקו אינו עולה על (כיוון שאפסילון נתון כחיובי) ולכן הינו מסלול ממצב ההתחלה למצב סוף במרחב P(S). לכן קיים מסלול חוקי כזה ולכן אלגוריתם UCS בשלב 3 ימצא מסלול חוקי (ראינו בהרצאה שאלגוריתם UCS הינו שלם) ויחזיר אותו.

1. יבש (3 נק׳): הוכח/הפרך: אם אלג׳ מחזיר פתרון אז הפתרון המוחזר בהכרח אופטימלי ע״פ **הקריטריון המשולב** שהוגדר מעלה. ראה בעמוד השני במסמך את ההערות המתייחסות לשאלות הוכח/הפרך.

**פתרון:**

הוכחה: נניח שהאלגוריתם החזיר פתרון במחיר P, אז מהגדרת האלגוריתם מתקיים כי המרחק הכולל שבו אינו עולה על (אחרת היה עולה ועל כן בלתי חוקי לפי ההבהרה) ומקבילות של אלגוריתם UCS מתברר כי הוחזר המרחק בעל מרחק הסעת מבחנות האופטימלי מבין הפתרונות החוקיים לפי מרחק כולל, ולכן אופטימלי לפי הקריטריון המשולב.

1. רטוב + יבש (0.5 נק׳): בשלב זה נממש ונריץ ווריאציה של (השינוי הוא שבמימוש נשתמש ב- עם היוריסטיקה קבילה במקום ב- ). השלימו בקובץ main.py את הקוד תחת ההערה הרלוונטית לסעיף זה. צרפו את התוצאות שקיבלתם לדו"ח (אין צורך במסלולים – מספיק עלויות הפתרון). השוו בטבלה לתוצאות הריצה מסעיפים קודמים (על אותה הבעיה עם שתי פונק׳ עלות השונות) ובדקו מספרית האם הפתרון המתקבל בסעיף זה אכן מקיים איזון בין שני המדדים. חשבו וצרפו לדו״ח את הערך . האם אכן נשמר ערך ה- הנקוב?

**פתרון:**

התוצאות הן:

מדד מרחק כולל

MDA(moderate\_MDA(8):Distance)

A\* (h=MDA-Sum-AirDist, w=0.500)

MDACost(dist= 43082.235m, money= 95.715NIS, tests-travel= 177571.274m)

מדד מרחקי נסיעת מבחנות:

MDA(moderate\_MDA(8):TestsTravelDistance(

A\* (h=MDA-TimeObjectiveSumOfMinAirDistFromLab, w=0.500)

MDACost(dist= 87850.369m, money=123.398NIS, tests-travel= 131265.153m)

מדד משולב:

MDA(moderate\_MDA(8):TestsTravelDistance)

A\* (h=MDA-TimeObjectiveSumOfMinAirDistFromLab, w=0.500

MDACost(dist= 65577.980m, money= 110.443NIS, tests-travel= 134889.839m)

|  |  |  |  |
| --- | --- | --- | --- |
| מדד כספי | מדד מרחקי נסיעת מבחנות | מדד מרחק כולל |  |
| 95.715NIS | 177571.274m | 43082.235m | סעיף 32 |
| 123.398NIS | 131265.153m | 87850.369m | סעיף 35 |
| 110.443NIS | 134889.839m | 65577.980m | סעיף 38 |

אכן נשמר האיזון בסעיף 38 כיוון שמדד המרחק נמצא בין שני הסעיפים הקודמים, וכנ"ל מדד מרחקי נסיעת מבחנות. ואכן נשמר ערך ה- הנקוב כיוון ש-

1. יבש (4 נק׳): 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. ראה בעמוד השני במסמך את ההערות המתייחסות לשאלות הוכח/הפרך (הטיפ כאן ניתן רק ככלי עזר לפיתוח האינטואיציה. יש לספק הוכחה/הפרכה פורמלית ומלאה לפי ההוראות וללא התייחסות לתוצאות ריצה כזו או אחרת).

**פתרון:**

הוכחה: אלגוריתם UCS הוא אלגוריתם שלם, והוא ימשיך לשותת בעץ (קריא: ליצור מסלולים בעץ) עד אשר ימצא פתרון בו מגיע למצב סוף. נניח שקיים פתרון במרחב , אז מסלול זה קיים בעץ החיפוש ולא ימחקו בו צמתים (כיוון שהמסלול חוקי לפי הנחה ולכן כל צומת בו אינו עולה על המרחק האופטימלי, וכיוון שמדובר ב**עץ** חיפוש, בו יש רק מסלול אחד לכל צומת מהשורש, ולכן אף מצב במסלול זה לא ימחק כיוון שניתן להגיע לכל צומת כזה רק דרך תת מסלול בP שהוא בפרט מסלול שאינו עולה על המרחק האופטימלי). לכן באיזשהו שלב האלגוריתם יגיע למסלול זה עקב סופיות עץ החיפוש (כי ראינו בסעיפים הראשונים שיש מספר סופי של מסלולים סה"כ ולכן גם מספר סופי של מצבים). יש לשים לב שאם מצב כבר קיים בopen הוא לא יושפע מיצירת אותו מצב דרך מסלול אחר שמרחקו גדול מהמרחק המותר, כי נתון שמוחקים **מיד** את המצב החדש, **לפני** שמוסיפים אותו לOPEN (ולכן לא ימחק כי אפילו לא בודקים אם הוא כבר נמצא).

1. יבש (4 נק׳): הוכח/הפרך: אם אלג׳ מחזיר פתרון אז הפתרון המוחזר בהכרח אופטימלי ע״פ **הקריטריון המשולב** שהוגדר מעלה. ראה בעמוד השני במסמך את ההערות המתייחסות לשאלות הוכח/הפרך.

**פתרון:**

הוכחה: נניח שהאלגוריתם החזיר פתרון במחיר C, אז מהגדרת האלגוריתם מתקיים כי המרחק הכולל שבו אינו עולה על (אחרת היה נמחק). יהי מסלול אופטימלי לפי הקריטריון המשולב (חוקי). כאמור בסעיף 39, עקב כך שהאלגוריתם רק מוחק צמתים ב**עץ** החיפוש, מסלול זה תמיד יהיה בעץ החיפוש ולא ימחק (נובע מההנחה שהוא אופטימלי ולכן חוקי). לכן, בכל איטרציה אלגוריתם UCS לא יפתח מסלולים שמרחק המבחנות שלהם עולה על מרחק המבחנות של הצומת (מתוך P) בעל k הגדול ביותר שנמצא בOPEN (ותמיד יש לפחות מצב אחד כזה בOPEN מתוך ההנחה שP מסלול חוקי ולכן מתחיל במצב ההתחלה, ומתוך ההנחה שעוד לא סיימנו את ריצת האלגוריתם ולכן עוד לא נמצא מצב סוף). מחיקת מצבים לא משנה אינוריאנטה זו כיוון שהיא לא משפיעה על היות איזשהו מצב מתוך P בOPEN (כולל המרחק הנלווה אליו תחת הפתרון האופטימלי P) ורק מצמצמת את OPEN (ולא מוסיפה אליו). לכן, מקבילות של אלגוריתם UCS מתברר כי הוחזר המרחק בעל מרחק הסעת מבחנות האופטימלי מבין הפתרונות החוקיים לפי מרחק כולל (כי אף פעם לא נפתח מסלולים בעלי מרחק הסעת מבחנות גדולים מהאופטימלי לפני שנפתח את P (לפי הגדרת אלגוריתם UCS) ולכן לא נפתח אותן בכלל (כי בסוף פיתוח P האלגוריתם ייעצר עקב כך שהסוף של P הוא מצב סוף לפי ההנחה שP מסלול חוקי)), ולכן אופטימלי לפי הקריטריון המשולב.

1. יבש (1.5 נק׳): ציין והסבר בקצרה יתרון צפוי של ע״פ במובנים של זמני ריצה. התייחס בתשובתך ליחסי הגדלים בין שני המרחבים (עליהם שני האלג' רצים). תשובה עד 3 שורות.

**פתרון:**

A2 עדיף על פני A1 בכך שמפתח פחות מצבים ועל כן לוקח פחות זיכרון וזמן ריצה. נשים לב ש: (מעבר ראשון לפי סעיף 1 ומעברים 2,3 לפי אי"ש ברנולי וכך שm,k חיוביים ממש). A2 רצה על לעומת A1 שרצה על ומספר פיתוחי המצבים שלהם בהתאם.

## חלק ח' – מימוש האלג׳ A\* והרצתו (1 נק׳ יבש)

1. יבש (1 נק׳): צרפו לדו"ח את התוצאות שקיבלתם בסעיף הקודם (אל תצרפו את המסלולים עצמם). האם חסכנו בפיתוחים? אם כן, בכמה? הסבירו למה בכלל ציפינו מראש ש- A\* יוכל לחסוך במס׳ הפיתוחים בתצורה שבה הרצנו אותו. לא מספיק לטעון ש- A\* גמיש יותר בבחירה של הצומת הבא לפיתוח. נסו להסביר למה בעצם אנחנו מצפים שהגמישות הזאת של A\* אכן תעזור לנו במקרה הזה לבחור מ- open צומת לפיתוח שיקדם אותנו מהר יותר למטרה. מה בעצם הוספנו לאלג' החיפוש? תשובה עד 2 שורות.

**פתרון:**

התוצאות שקיבלנו הן:

MDA(small\_MDA(5):Distance) A\* (h=MDA-MST-AirDist, w=0.500) time: 18.59 #dev: 543 |space|: 877 total\_g\_cost: 31528.65909 total\_cost: MDACost(dist= 31528.659m, money= 49.717NIS, tests-travel= 52112.429m) |path|: 8

MDA(small\_MDA(5):Distance) A\*eps (h=MDA-MST-AirDist, w=0.500) time: 6.17 #dev: 491 |space|: 821 total\_g\_cost: 31528.65909 total\_cost: MDACost(dist= 31528.659m, money= 49.717NIS, tests-travel= 52112.429m) |path|: 8

לפי התוצאות המצורפות ניתן לראות שהצלחנו לחסוך בפיתוח של 52 מצבים.

צפינו מראש ש- A\* יוכל לחסוף במספר הפיתוחים בתצורה שבה הרצנו אותו מאחר שלטובת דירוג המצבים ב- Open השתמשנו בהיוריסטיקה הקבילה (MST) ולטובת בחירת הצומת מה-Focal השתמשנו בהיוריסטיקה הלא קבילה המיודעת יותר (Sum). מאחר שכל פעם בחרנו את הצומת המתאים מ-Open כמו ש- A\* היה בוחר אבל כן שיפרנו אותו כי בחרנו כל פעם את הצומת המתאים ביותר באמצעות שימוש בהיוריסטיקה המיודעת Sum.

## חלק י' – שאלה תאורטית (12 נק' יבש)

סעיף (א) – 1 נק' יבש

כזכור, בכיתה הצגנו את אלגוריתם שהינו שלם וקביל. לאחר מכן, הצגנו את אלג׳ IDA\* שמטרתו הייתה לשפר מדד ביצועי כלשהו של אלג׳ . ציין במילה **אחת** מהו אותו מדד ביצועי עבורו אלג׳ IDA\* עדיף תמיד על פני אלג׳ A\*. הסבר (עד 2 שורות).

**פתרון:**

המדד הביצועי עבורו אלג' IDA\* עדיף תמיד על פני אלג' A\* הוא **זכרון**.

מאחר והאלג' מבצע העמקה הדרגתית כי ניתנת לו מגבלת f\_limit, ולכן מבוצע חיפוש בעץ כאשר צריכת הזיכרון היא לינארית באורך המסלול בניגוד ל-A\* שצריכת הזכרון שלו היא בפרופורציה ל-Open U Closedסעיף (ב) – 5 נק' יבש

1. (1 נק' יבש) באיזה מדד ביצועי אלג׳ IDA\* עלול להיות משמעותית פחות טוב מאשר אלג׳ במקרים רבים? תשובה עד 2 מילים.
2. (2 נק' יבש) למה מדד זה נפגע ב- IDA\* (לעומת )? תשובה עד שורה אחת.
3. (2 נק' יבש) האם מדד זה נפגע באותו האופן כמו שהוא נפגע ב- ID-DFS לעומת BFS? אם כן, למה? אם לא, מה ההבדל? תשובה עד 3 שורות.

**פתרון:**

1. סיבוכיות זמן
2. IDA\* לעומת A\* מפתח פעמים רבות את אותם הצמתים (שפיתח בעבר) מאחר וחוסך בזכרון
3. מדד זה לא נפגע באותו האופן כמו שהוא נפגע ב-ID-DFS לעומת BFS מאחר שאמנם ב-ID-DFS אנחנו גם מפתחים את אותם הצמתים שפותחו בעבר, אבל פעולה זו לא יקרה באותה מידה מאחר שבעץ רוב הצמתים הם בתחתית, והצמתים שבשכבות העליונות הם אלה שמפתחים מספר פעמים.
4. אופציה 2 : מדד זה נפגע באותו האופן כמו שהוא נפגע ב-ID-DFS לעומת BFS מאחר שב-ID-DFS אנחנו גם מפתחים את אותם הצמתים שפותחו בעבר (מכניזם דומה), אבל הפגיעה קשה יותר כי כעת לא כל הקשתות בעלות משקל 1 ולכן יהיה הרבה יותר איטרציות (אחת לכל עלות שקיימת בגרף החיפוש) במקום איטרציה אחת לכל עומק כמו שהיה ב-BFS.

סעיף (ג) – 6 נק' יבש

אלג׳ דומה ל- IDA\* (הרגיל), עם השינויים הבאים:  
(א) משנים את ערך ה- f-limit ההתחלתי להיות f-limit := .  
(ב) משנים את כלל העדכון של f-limit באופן הבא:

ניסוח אלטרנטיבי שקול:

כאשר הינו כלל העדכון של f-limit באלג׳ IDA\* המקורי. כלומר זהו הערך שאלג׳ IDA\* המקורי היה בוחר בתור ערך ה- f-limit הבא בתום האיטרציה האחרונה שבוצעה ע״י .

שימו לב, יש לספק ביטויים מתמטיים סגורים התלויים בקבועים המוגדרים בשאלה בלבד. בפרט, אין להגדיר קבועים אחרים שאינם מופיעים בגוף השאלה.

1. (3 נק') כמה איטרציות לכל היותר יבצע על ? הסבר (לכל היותר 3 שורות).
2. (3 נק') ספק חסם עליון הדוק עבור . הסבר (לכל היותר 3 שורות).

**פתרון:**

1. לפי הנתון החישוב של ה-Limit הבא הוא לפחות , כדי למצוא את מס' האיטרציות צריך לחשב את ההפרש המינימלי האפשר בין ה-Limit ההתחלתי לסופי. נבדיל בין שני תרחישים הראשון הוא בו ה-Limit הבא הוא ואז מתבצעות איטרציות, לבין התרחיש בו נבחר ואז מתקיימות . לסיכום מספר האיטרציות הוא:
2. נשים לב שכאשר הגענו לאיטרציה לפני הסיום כעת ה-Limit הבא שיבחר הוא: , כלומר הקפיצה ב-Limit החדש היא לפחות ב- ולכן נקבל שחסם עליון הדוק עבור הוא :