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

**מטרות התרגיל**

* נתמודד עם בעיות פרקטיות ותיאורטיות של חיפוש במרחבי מצבים עצומים.
* נתרגל את הנלמד בהרצאות ובתרגולים.
* נתנסה בתכנות ב- python לפתרון בעיות פרקטיות.

**הנחיות כלליות**

* **תאריך הגשה:** יום שלישי, 01.12.2020, בשעה 23:59.
* את המטלה יש להגיש **בזוגות בלבד**.
* יש להגיש מטלות מוקלדות בלבד. פתרונות בכתב יד לא ייבדקו.
* התשובות צריכות להיות כתובות בשפה העברית.
* ניתן לשלוח שאלות בנוגע לתרגיל לתיבת המייל הקורסית: [ai.technion@gmail.com](mailto:ai.technion@gmail.com). אנו מבקשים לא לשלוח הודעות בנוגע לתרגיל לתיבות הדואר של הסגל. לפני שליחת שאלה, בדקו האם קיימת לה תשובה כבר ב- FAQ. נציין כי שאלות שנענו כבר ב- FAQ לא יענו שוב במייל.
* המתרגל האחראי על תרגיל זה: אלעד נחמיאס.
* בקשות דחיה **מוצדקות** (מילואים, אשפוז וכו') יש לשלוח למתרגל האחראי (טל סויסה) בלבד.
* במהלך התרגיל ייתכן שנעלה עדכונים, תיקונים והבהרות לדף FAQ ייעודי באתר ולמסמך הנ"ל. העדכונים הינם **מחייבים**, ועליכם להתעדכן עד מועד הגשת התרגיל.
* שימו לב, התרגיל מהווה כ- 30% מהציון הסופי במקצוע ולכן העתקות תטופלנה בחומרה.
* ציון המטלה יורכב מהגורמים הבאים:
  + **60% - המסמך היבש.** מעבר לתשובות הנכונות, אתם נבחנים גם על הצגת הנתונים והתוצאות בצורה קריאה ומסודרת במקומות בהם התבקשתם לכך. הניקוד המפורט בסעיפים של מסמך זה הינו מתוך הציון היבש בלבד.
  + **40% - הקוד המוגש.** הקוד שלכם ייבדק באופן מקיף ע״י מערכת בדיקות אוטומטיות. המערכת תבדוק את התוצאות שלכם לעומת התוצאות המתקבלות במימוש שלנו. אנו מצפים שתקבלו את אותם הערכים בדיוק. נבדוק בין היתר את המסלול המתקבל, את עלותו ואת מס׳ הפיתוחים. לכן עליכם להיצמד להוראות בתרגיל זה. הבדיקות יהיו כמובן מוגבלות בזמן ריצה. ייתנן לכם זמן סביר ביותר להרצת כל טסט. אם תעקבו אחר ההוראות במסמך זה ובקוד אין סיבה שלא תעמדו בזמנים אלו. בנוסף, יש להקפיד על הגשת קוד מסודרת בהתאם להנחיות. יש לכתוב הערות במקומות חשובים בקוד כדי שיהיה קריא וקל לבדיקה ידנית.
* אנו יודעים שעבור חלקכם זו התנסות ראשונה בכתיבת קוד בפיתון ותרגיל זה מתוכנן בהתאם לכך.
* שימו לב שלא יענו שאלות בסגנון: ״איך מוצאים את עלות הפתרון שהוחזר?״ / ״איך ניגשים למפות הכבישים מתוך המימוש של הפונק׳ ההיא?״ / ״באיזה שדה שמור ה...?״ / ״אילו שדות מצפים לקבל אובייקט מטיפוס frozenset?״ וכדומה. בכל מקום בקוד בהם אתם נדרשים להשלים את המימוש (לכתוב קוד כלשהו) השארנו לכם הערות מפורטות שמסבירות כיצד יש לעשות זאת. ברוב המקומות גם הכוונו אתכם במפורש לשמות השדות ולמתודות הרלוונטיות להם תזדקקו. בחלק מהמקומות החסרנו חלק מהפרטים בהסבר מתוך כוונה – אנחנו רוצים לעודד אתכם לעיין בקוד ולמצוא פרטים אלו בכוחות עצמכם. הכרת סביבת העבודה שסיפקנו לכם והתמצאות בה הן למעשה חלק מהתרגיל.
* בתרגילי הבית בקורס הרצת הניסויים עשויה לקחת זמן רב. לכן מומלץ מאוד להימנע מדחיית העבודה על התרגיל ו/או כתיבת הדו״ח לרגע האחרון. לא תינתנּה דחיות על רקע זה.
* מסמך זה כתוב בלשון זכר מטעמי נוחות בלבד, אך מתייחס לנשים וגברים כאחד.

**הערות טכניות**

* גרסת python איתה אתם נדרשים לעבוד הינה 3.7. גם קבצי המקור שקיבלתם מתאימים לגרסה זו.
* לנוחיותכם, בקוד שסופק לכם הוכנסו type-annotations (ציון של טיפוסים של שדות/פרמטרים). זאת במטרה להקל עליכם בהתמצאות בקוד. אנחנו מצפים מכם להשכיל ולהשתמש ב- IDE (ממליצים על PyCharm) שיוכל לסייע לכם להתמצא בקוד ביתר קלות, יציע לכם השלמת שדות, ויזהה עבורכם שגיאות בצורה סטטית – כל אלו יחסכו לכם הרבה זמן. ה- type-annotations עוזרים ל- IDEs לעזור לכם – נצלו את זה.
* כאמור, הבדיקות האוטומטיות של הקוד שתגישו תהיינה מוגבלות בזמן פר טסט. היו סמוכים ובטוחים שמערכת הבדיקה הינה הוגנת ביותר. מימוש תקין שנצמד להוראות יעמוד במסגרת הזמנים. הסיבה למגבלת הזמן היא פשוטה – לא ניתן להריץ כל טסט אינסוף זמן – אנחנו צריכים לבדוק את כל התרגילים שלכם במסגרת זמן סבירה. בכדי לעמוד במסגרת הזמנים אתם לא מתבקשים לחשוב על אופטימיזציות כאלו או אחרות, אלא רק לעקוב באדיקות אחר ההוראות. הבינו איך משתמשים ב- iterators בפיתון ונסו להשתמש בהם בכל מקום שתוכלו (במקום ליצור רשימות איפה שאין באמת צורך בכך). אנו מכווינים אתכם לעשות כך בחלק מהסעיפים. קשה לפרט דרישת זמנים קשיחה כי לכל אחד יש מחשב בעל מפרט אחר וזה כמובן יכול להשפיע באופן ניכר על זמני הריצה. נפרט כאן הערכה כללית לזמן הריצה הצפוי של מימוש תקין במחשב אישי מודרני סביר, וזאת רק בכדי שתוכלו לקבל סדר גודל ולוודא שאתם לא חורגים מכך באופן דראסטי. אם אתם חורגים מהאמור באופן דרסטי ייתכן שיש לכם טעות במימוש – היעזרו אחד בשני כדי למצוא אותה. הריצה הארוכה ביותר אמורה לקחת כ- 3 דקות. היעזרו בהערכה גסה זו כדי לוודא/לחשוד בתקינות המימוש שלכם.
* אלא אם נכתב אחרת, אין לשנות פונקציות מוכנות שקיבלתם. בנוסף, אין לשנות את החתימה של פונקציות שהתבקשתם לממש או אחרות. בפרט, אין לשנות תוכן קבצים בהם לא נתבקשתם לבצע שינויים. אין ליצור פונקציות עזר משלכם, אנא השלימו את המימושים אך ורק במקומות המסומנים. בנוסף, אין ליצור קבצים חדשים, אלא לערוך את הקבצים שהתבקשתם במפורש בלבד. ראו הוזהרתם - חריגה מכללים אלו ככל הנראה תוביל לכישלון מיידי בבדיקות האוטומטיות. אם יש בעיה נקודתית, ניתן לשלוח מייל לתיבה הקורסית.
* אין להוסיף ו/או לשנות פקודות import בקוד. כל מה שאתם צריכים כבר מיובא במקום הרלוונטי. שימו לב שלעיתים IDEs שונים עלולים להוסיף לכם שורות import באופן אוטומטי. אחריותכם לוודא, טרם הגשת התרגיל, ששורות ה- import בקוד אותו אתם מגישים זהות לשורות בקבצים המקוריים שקיבלתם.
* אין לבצע בעצמכם טעינה של קלטים או מפות. אנחנו עשינו זאת עבורכם במקומות הנדרשים. בכל אזור בקוד בו שהתבקשתם להשלים את המימוש יש גישה לכל המבנים להם אתם זקוקים לצורך המימוש.
* לצורך ההרצות תצטרכו להתקין את החבילות הבאות של python: numpy, scipy, matplotlib, networkx. חלק מחבילות אלו מותקנות כברירת מחדל עם ההתקנה של Anaconda. את אלו שאינן מותקנות אפשר להתקין בעזרת הפקודה `pip install <package name>`.

**הוראות עבור שאלות הוכח/הפרך על מרחב MDA בתרגיל**

* הנח שכל הנקודות במפת הכבישים הן נקודות ב- והמרחק בין זוג נק׳ הוא המרחק האוקלידי.
* הנח כי כביש במפת הכבישים הוא בהכרח דו-כיווני. הכביש הינו קו ישר במישור.
* הפרכות:
  + הפרכה אפשרית אך ורק בעזרת פירוט של **דוגמא נגדית**. תשובות הפרכה ללא מתן דוגמא נגדית קונקרטית ושלמה לא יזוכו בנקודות. בפרט, לא יתקבלו תשובות שינסו לתאר רעיון איך אפשר לבנות דוגמא נגדית או להפריך את הטענה באמצעות שימוש בטענות.
  + דוגמא נגדית היא למעשה **הגדרה מלאה של מפת כבישים + מרחב MDA** (כולל למשל מספר מטושים בכל מעבדה, מספר דיירים בכל דירה, קיבולת האמבולנס, מספר מטושים התחלתי באמבולנס וכו') - ללא נתונים אלו לא נוכל לבדוק לכם את התשובה. אין צורך לספק איור של מרחב MDA כולל האופרטורים שלו. יש לספק איור של מפת הכבישים תוך ציון הפרטים של כל נקודה בה (האם זו דירה/מעבדה/נק' התחלה ואת כל הנתונים שלה). לשם הפשטות, אפשר לבנות מפת כבישים בה כל הנקודות הן נקודות רלוונטיות למרחב MDA עם כבישים ישירים בניהם.
  + אם ישנם קבועים נוספים בשאלה (כמו אפסילון) – אל תשכחו לספק אותם. זכרו שבעת הבדיקה לא נוכל לנסות לנחש נתונים חסרים כדי להשתכנע שהדוגמא שלכם פועלת. תפקידכם הוא לשכנע אותנו בכך.
  + בנוסף, יש לצרף לדוגמא הנגדית **טבלת מעקב** אחר ריצת האלג' המדובר. זאת הדרך היחידה שתשכנע אותנו (ואתכם) שהדוגמא שלכם אכן מהווה סתירה לטענה. לא נוכל לבדוק את הדוגמא שלכם ללא המעקב. בטבלת המעקב מופיעים שלבי הריצה של האלג' ובכל אחד מפורטים מבני הנתונים של האלג' (open, close) וכל הצמתים המופיעים בהם כולל ערכי f, g, h שלהם. בטבלה יצוין מיהו הצומת הנשלף מ- open בתחילת כל שלב בריצה. הטבלה תהיה קריאה ברורה ומסודרת ויהיה קל לעקוב אחרי ריצת האלג' הרלוונטי על הדוגמא שלכם מתוך התבוננות בה.
  + שימו לב להבדל בין נקודה על מפת הכבישים לבין מצב במרחב החיפוש עליו האלגוריתם רץ. עבור נקודה מסוימת במפה יכולים להיות מספר מצבים שונים במרחב החיפוש שבהם המיקום הנוכחי הוא אותה הנקודה. הקפידו ליצור הבדל בטבלת המעקב בין **נקודות שמייצגות מצבים שונים**. זה יעזור לכם ולנו להשתכנע שהמעקב תקין.
  + מפת הכבישים צריכה להיות קשירה ובעיית ה- MDA המתוארת צריכה להיות פתירה עבורה אלא אם נאמר אחרת.
  + על המרחב להיות קטן ככל הניתן (מספר **מינימלי** של נקודות במפת הכבישים). טיפ: קודם כל נסו למצוא דוגמא שעובדת ולאחר מכן צמצמו אותה. שימו לב: לא יענו שאלות בסגנון "מה הגודל של הדוגמא המינימלית שקיימת?" / "האם הדוגמא שלי מספיק קטנה או שכדאי לי לחשוב עוד איך לצמצם אותה?"
  + שימו לב: יש להראות שאכן ניתן **למקם את הנקודות במישור** באופן שתיארתם תוך שמירה על המרחקים שציינתם על גבי הקשתות (קיום אי-שוויון המשולש). אם זה לא טריוויאלי ציינו את הקואורדינטות של הנקודות (x,y) במישור.
* הוכחות:
  + הוכחה צריכה להיות פורמלית ומסודרת ואורכה יהיה לכל היותר 7 שורות.
  + הסבר רעיוני/אינטואיטיבי לא יזוכה בנקודות כלל.

אנחנו קשובים לפניות שלכם במהלך התרגיל ומעדכנים את המסמך הזה בהתאם. גרסאות עדכניות של המסמך יועלו לאתר. הבהרות ועדכונים שנוספים אחרי הפרסום הראשוני יסומנו כאן בצהוב. בנוסף, לכל עדכון יהיה מספר גרסה כדי שתוכלו לעקוב. ייתכן שתפורסמנה גרסאות רבות – אל תיבהלו מכך. השינויים בכל גרסה יכולים להיות קטנים.



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

1. יבש (1 נק'):

ראשית נסדר את הדירות – K! , לאחר מכן יש m אפשרויות לכל מעבדה בין דירות + אפשרות ללא מעבדה- ונשארו m אפשרויות למעבדה האחרונה, סה"כ:

1. יבש בונוס (3 נק'):

דומה לסידור הקודם, אך הפעם בין כל שתי דירות נחלק לשני מקרים- מקרה ראשון, לא עברנו במעבדה ולכך יש אפשרות אחת. מקרה שני, עברנו במעבדה: כעת נבחר את מספר המעבדות הנוספות שנעבור בהן ונסדר את המעבדות שנבחרו, כלומר:

סה"כ:

1. יבש (2 נק׳):

|  |  |  |  |
| --- | --- | --- | --- |
| Estimated calculation time | *#possiblePaths* |  |  |
| ~18.48 [secs] | ~22.04\*10^6 | 2 | 7 |
| ~3.85 [mins] | ~24.77\*10^7 | 3 | 7 |
| ~2.26 [hours] | ~79.27\*10^8 | 3 | 8 |
| ~19.56 [hours] | ~63.00\*10^9 | 4 | 8 |
| ~3.69 [days] | ~28.54\*10^10 | 3 | 9 |
| ~5.33 [months] | ~11.42\*10^12 | 3 | 10 |
| ~21.05 [years] | ~50.23\*10^13 | 3 | 11 |
| ~1.08 [thousand years] | ~24.11\*10^15 | 3 | 12 |
| ~22.41 [thousand years] | ~46.78\*10^16 | 4 | 12 |
| ~1.55 [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׳):   
   במקרה המקסימלי, אנחנו נמצאים במצב ההתחלתי ונוכל להגיע לכל דירה או לכל מעבדה  
   כלומר   
   במקרה המינימלי, אנחנו במעבדה, ואם אנחנו במצב מטרה, אז הדרגה היא 0.
2. יבש (1 נק׳):   
   לא ייתכנו- כי כדי שיווצר מעגל, נצטרך לבקר באותה דירה פעמיים או באותה מעבדה פעמיים:  
   לא נבקר באותה דירה, כי מהגדרת האופרטור נוכל לבקר בדירה רק אם ואחרי ביקור אחד, di מתווסף לTaken.

ביקור לשם לקיחת מטושים יקרה פעם אחת במעבדה. בביקור חוזר, הגענו עם Transferred מסוים, ובהכרח המספר ישתנה בהגעה השנייה למעבדה ולכן זהו לא אותו מצב אף פעם.

1. יבש (1 נק׳):   
   מספר המצבים הוא אינסופי כיוון שיש אינסוף אפשרויות למספר המטושים (N טבעי). לא כל המצבים ישיגים, לדוגמא המצב בו מספר המטושים גדול יותר ממספר המטושים ההתחלתי + מספר המטושים הקיים בכל במעבדות.
2. יבש (1 נק׳):   
   ייתכנו בורות ישיגים, לדוגמא המצב בו מספר המטושים הכולל נמוך ממספר הדיירים הכולל (כלומר נתקע בשלב מסויים ללא מטושים אך עם דירות שלא ביקרנו בהן)
3. יבש (1 נק׳):

האורך המינימלי של מסלול: נניח שנתחיל עם מספר מטושים המספיק לבקר בכל הדירות, נשאר לבקר במעבדה האחרונה, לכן סה"כ: K+1

האורך המקסימלי: נניח שנתחיל בביקור בכל המעבדות ללקיחת מטושים ולאחר מכן נבקר בכל הדירות כשאחרי כל דירה נבקר במעבדה, לכן סה"כ: M + 2\*K

1. יבש (1 נק׳):

כש-

## חלק ד' – מתחילים לתכנת

### תרגילים

1. רטוב

## חלק ה' – אלגוריתם A\* (2 נק׳ יבש)

### תרגילים

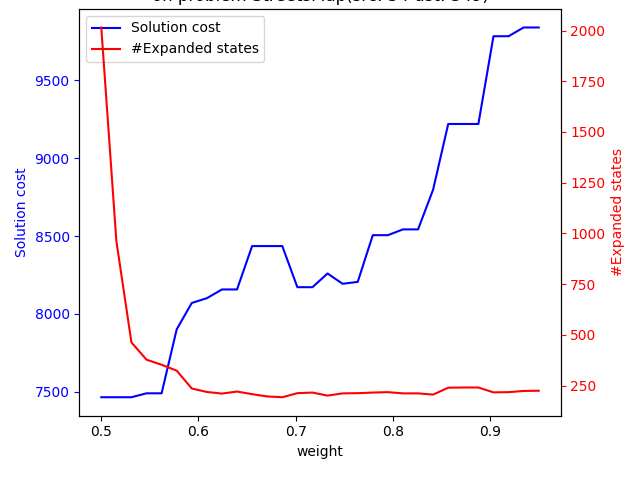
1. רטוב
2. רטוב
3. רטוב
4. יבש (1 נק')

מספר מצבי פיתוח ( nullhuristic) : 17354

מספר מצבי פיתוח ( AirDistHeuristic ) : 2015

החיסכון: 88 אחוז

1. רטוב
2. יבש (1 נק׳):



**הסבר**: הגרף מתאר את מספר הצמתים שפותחו<אדום> ועלות הפתרון <כחול> (של ריצת A\* עם AirDistHeuristic) כתלות בערכי w שונים בתחום [0.5,0.95].

**ערך W**: נרצה ערכי w עבורם נפתח כמה שפחות מצבים ונשלם כמה שפחות על הפתרון, מהגרף ניתן לראות כי נקודה זו מתקבלת באזור שבין 0.58~ לבין 0.52~, כלומר זה האזור העדיף בשבילנו.  
נשים לב שהכלל אכן בא לידי ביטוי בגרף באופן כללי אך יש מקטעים בהם הוא אינו נכון, למשל בקטע 0.68~ עד 0.71~ שם יש ירידה בעלות הפתרון אך עלייה בערכי w.

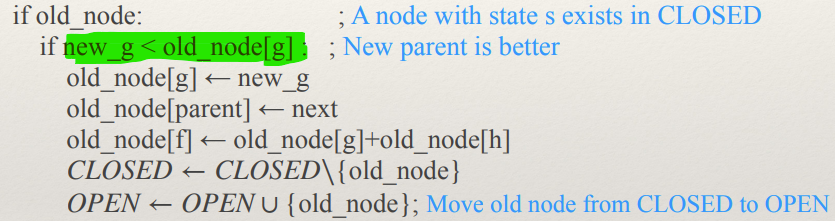
## חלק ו׳ – מימוש בעיית מד״א (15 נק׳ יבש)

1. רטוב
2. רטוב
3. שאלה יבש (2 נק'):

במרחב "העל" יהיו הרבה יותר מצבים אפשריים, מסלולים יתנפחו והcache יהיה מנופח בהתאמה. כיוון שאנחנו בודקים עלויות בcache בפתרון בעיית המרחק, cache גדול יותר אומר יותר זמן לחפש בו, ודבר זה יפגע לנו ביעילות זמן ריצה של הפתרון.

1. שאלה יבש (3 נק׳):   
   (i)   
     
   (ii)

שורה זו לא מספיקה -היא מונעת שינוי של שדות שניגשים אליהם by value אבל לא מונעת שינוי של שדות שניגשים אליהם by reference . אל הסטים ניגשים by reference ולכן הם מוגדרים להיות FrozenSet, וזהו הדבר הנוסף שמבטיח שלא יהיה ניתן לשנות בטעות את האובייקט.  
(iii)

כן יכול להווצר מצב כזה. במקרה בו בעת פיתוח צומת נגלה מצב שכבר פיתחנו אשר ערך הפונקציית g שלו "זולה" יותר, ואז נכניס את הצומת לתור ה OPEN שוב למרות שכבר פיתחנו את המצב הזה , רק שהפעם נעדכן את ערכי g ו f לערכים החדשים שגילינו . זה קורה פה:  


(iv)

הצורך בהגדרת MDAState כ- frozen נובע מזה שבעת פיתוח צמתים עוקבים, כאשר סורקים צומת מסוים לא נרצה לאפשר למצב העוקב שמפותח לשנות את הערכים של מצב האבא שלו, מכוון שכל צומת עוקב הוא חלק ממסלול שונה בסריקה. שינויים כאלה עלולים לגרום להגדרת מצבים שגויים ולשגיאות במהלך החיפוש. לדוגמא במתודה cost\_with\_state\_expand של המחלקה MDAProblem :

המתודה מחזירה **צמתים** **עקבים** עבור צומת מסוים עם שימוש ב yield. כאשר מעצבעים פיתוח של עוקב הנוצר מביקור במעבדה L כלשהי ( הפעלת אופרטור ביקור) ניתן לשנות ישירות את הרשימה visited\_labs באופן הבא : להוסיף את המעבדה L לרשימה visited\_lbas ואז לבצע השמה של המעבדה L ל visited\_labs במצב החדש שפיתחנו.

state\_to\_expand.visited\_labs.append(L)

new\_state.visited\_lab=L

כתוצאה מכך גם האבא וגם המצב העוקב חולקים מצביע לאותה רשימה visited\_labs בעוד שהרשימה לא נכונה עבור האבא. בפעם הבאה שנפתח עוקב לאותו מצב השגיאה תגרום לפיתוח מצבים לא נכונים

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

1. רטוב
2. רטוב
3. יבש (4 נק׳): הוכח/הפרך:   
   היוריסטיקה קבילה. הוכחה:

ע"פ הגדרה h יוריסטיקה קבילה אם מתקיים:

נסמן h'= MDAMaxAirDistHeuristic . h' נותנת לכל צומת ערך שמציין את המרחק האווירי המקסימלי בין 2 נקודות בהן המסלול צריך לעבור (שעדיין לא עבר בהן). כלומר הערך מציין את המרחק המקסימלי בין 2 דירות שעדיין לא ביקרו בהן. נבחן את המקרה בו נשארה דירה אחת בלבד שעדיין לא ביקרו בה. מתקיים: h'(s)≤〖h'〗^\* (s) . מאחר והמרחק האווירי הוא יוריסטיקה קבילה ( הוכח בהרצאה). לפי א"ש המשולש כל הוספה של דירה תאריך את המסלול ולכן מתקיים לכל s :

h'(s)≤h^('\*) (s)

ולכן קיבלנו כי h' קבילה ע"פ הגדרה.

1. רטוב
2. רטוב
3. יבש (4 נק׳): הוכח/הפרך:

לא קבילה. דוגמא נגדית: עבור k=3 ומיקומי הדירות הם : והמיקום ההתחלתי של האמבולנס הוא (0,0) נקבל:

H(s)=sum((0,0)=>(1,0)=>(3,0)=>(-2,0))=1+2+5=8

אבל קיים מסלול קצר יותר:

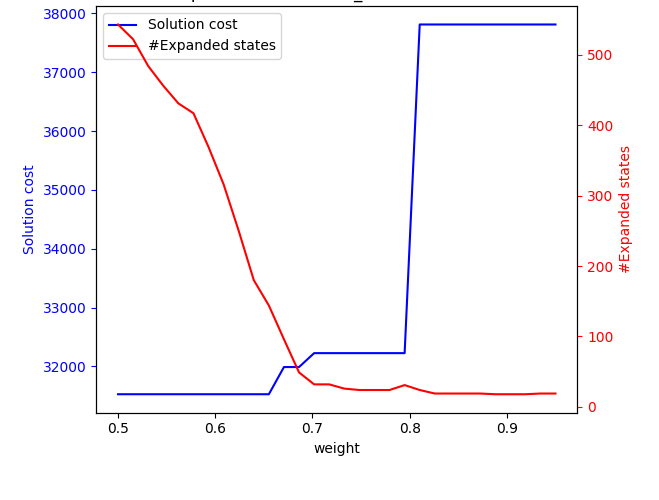
H(s\*)=sum((0,0)=>(-2,0)=>(1,0)=>(3,0))=2+3+2=7

כלומר מצאנו מסלול s\* עבורו *. ולכן זו לא יוריסטיקה קבילה.*

1. רטוב
2. רטוב
3. יבש (4 נק׳): הוכח/הפרך:   
   יוריסטיקה קבילה:

ע"פ הגדרת הפונקציה הערך של מצב s הוא סכום קשתות ה MST של גרף G , כאשר הצמתים בו מוגדרים להיות דירות שעדיין לא בוקרו בהם ממצב S. משקל כל קשת הוא המרחק האווירי בין שני צמתים. על מנת להגיע ממצב s למצב מטרה צריך לעבור בכל הצמתים בגרף G. , כלומר ניתן להסתכל על הבעיה של מסלול בין דירות כמסלול של עץ פורש בגרף G. נתון שהערך היוריסטי של s הוא ערך העץ הפורש המינימאלי של גרף זה , בהכרח כל מסלול אחר בין הדירות יהיה **לפחות**  באותו המחיר. מחיר של כל קשת מוגדר לפי המרחק האווירי בין שתי דירות שהיא מחברת, זאת אומרת ש**מקבילות יוריסטיקת המרחק האווירי**  ומכך שמדובר ב **עץ פורש** נקבל

ע"פ הגדרה קיבלנו ש MDAMSTAirDistHeuristic קבילה.

1. רטוב + יבש (1 נק׳):  
   

הסבר:

ערכים כדאיים:

## 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).

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

1. רטוב + יבש (0.5 נק' יבש):

מס' נשווה את שתי סוגי הבעיות שבדקנו, mda small ו mda moderate, עם monetary cost לעומת distance cost:  
MDA(small\_MDA(5):Distance): total\_cost: MDACost(dist= 31528.659m, money= 49.717NIS, tests-travel= 52112.429m)  
MDA(moderate\_MDA(8):Distance): total\_cost: MDACost(dist= 43034.794m, money= 95.847NIS, tests-travel= 176505.013m)  
MDA(small\_MDA(5):Monetary): total\_cost: MDACost(dist= 31923.809m, money= 42.050NIS, tests-travel= 53317.118m)  
MDA(moderate\_MDA(8):Monetary): total\_cost: MDACost(dist= 54951.037m, money= 77.201NIS, tests-travel= 172922.318m)  
**SMALL**: כשנלקח distance cost, קיבלנו מחיר: NIS49.717  
לעומת בmonetray cost, שקיבלנו: 42  
**MODERATE**: במקרה עם distance cost קיבלנו מחיר: NIS95.847  
עם monetary: 77

1. רטוב
2. יבש (2 נק׳):

היוריסטיקה קבילה עבור testtravel cost:  
ניקח מצב S כלשהו. נניח h\*(s) = 0. אזי אין יותר דירות לבקר ולכן גם h(s) = 0.  
אחרת, h\*(s) > 0, נותרו דירות לביקור. לכל דירה k נסמן מרחק למעבדה הקרובה ביותר k\_closet\_lab ומרחק שהאמבולנס נסע בפועל בפתרון האופטימלי למעבדה k\_real\_lab.  
כיוון שלכל דירה, k\_closet\_lab < k\_real\_lab, אז גם h(s) = sum(k\_closet\_lab) < sum(k\_real\_lab) = h\*(s)  
והדבר נכון לכל S, לכן היוריסטיקה אופטימית תמיד, כלומר קבילה.

1. רטוב + יבש (0.5 נק' יבש):

הפלט שיצא:  
MDA(moderate\_MDA(8):TestsTravelDistance):  
 total\_cost: MDACost(dist= 93226.428m, money= 59.969NIS, tests-travel= 131265.153m)  
ניתן לראות שהמרחק שהבדיקות עברו בפתרון זה: tests-travel= 131265.153m   
קטן מהמרחק שהן עברו בפתרונות הקודמים: tests-travel= 176505.013m

### שילוב בין 2 מדדים

1. יבש (3 נק׳): הוכח/הפרך:

הוכחה:

נניח שקיים פתרון במרחב המקורי, ונסמן את מסלול הפתרון האופטימלי - q= .

מסלול זה הוא מצב מקבל במרחב המסלולים, נסמן את המסלול המשוייך אליו   
ב- .p = כיוון שעלות המצב הכי יקר היא מתקבל שלכל צומת-

לכן עבור העלות שקיבלנו כל צומת יקבל ערך קטן מאינסוף ויוכלו להפתח אם נרוץ מספיק זמן. בסופו של דבר גם יפתח מצב מטרה, ולכן אם קיים פתרון במרחב המקורי האלגוריתם בהכרח יחזיר פתרון.

1. יבש (3 נק׳): הוכח/הפרך:   
   הוכחה:

מהגדרת , יש מסלולים (לפחות אחד, נסמנו p) שכל צמתיו מקיימים את התנאי בגרף   
ש לכל אפסילון. מהגדרת פונ' המחיר, המחיר של מסלולים אלו נמוך משל כאלה שאינם מקיימים תנאי זה, ובפרט לא ייתכן שנפתח מצב מטרה שאינו מקיים תנאי זה (בגלל מינימליות UCS). יתר על כן, מצב מטרה שנפתח הוא המצב המינימלי מכל אלו המקיימים את התנאי (מאותה סיבה) ולכן גם בעל ערך המינימלי.  
סה"כ מצב מטרה שנפתח עם האלגוריתם יהיה המינימלי תחת שתי פונ' העלות ולכן המסלול שיתקבל אופטימלי ע''פ הקריטריון המשולב

1. רטוב + יבש (0.5 נק׳):

הפלט שהתקבל:  
MDA(moderate\_MDA(8):TestsTravelDistance):

total\_cost: MDACost(dist= 65686.522m, money= 99.486NIS, tests-travel= 132209.981m)

לעומת סעיפים קודמים (מסודר בסדר עולה של dist):

|  |  |  |
| --- | --- | --- |
|  | Test-travel | dist |
| A\* , testtravel cost | 131265.153m | 43034.794m |
| A2 | 132209.981m | 65686.522m |
| A\*, dist cost | 176505.013m | 93226.428m |

ניתן לראות ש 2A אכן מאזן בין המדדים(נמצא בין שניהם), אך מוותר לשם כך על הdist האופטימלי.

חישוב הערך הנל: (65686.522/43034.794) -1 = 0.53 ~ 0.6 , ערך ה-epsilon אכן נשמר!

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).

הפרכה:

דוגמא נגדית-  
ניקח גרף עם הצמתים הבאים: i=(0,0), d1=(1,0), d1=(1,0), d1=(1,0), d1=(1,0),

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |
| 6 | 5 | 4 | D1 | i | i | Next |
|  |  |  | {d2,l1,l2} | {d1,d2} | {i} | Open |
|  |  |  | {I,d1} | {i} | {} | Close |
|  |  |  |  |  |  |  |

1. יבש (4 נק׳):

הוכחה:

לפי אופן פעולת האלגוריתם, הוא מחפש פתרון לפי פונ' העלות , לכן אם מוחזר פתרון הוא יהיה אופטימלי ביחס לפונ' זו(מאופטימליות UCS). בנוסף, הפתרון מקיים שלכל צומת העלות שלו קטנה מ,(אחרת היה נמחק ולא מפותח),ומאופטימליות נקבל כי הפתרון המוחזר אופטימלי ע"פ הקריטריון המשולב.

1. יבש (1.5 נק׳):

פשוט רץ על מרחב הרבה יותר גדול מ(מרחב המסלולים של ), דבר המייקר את זמן הריצה.

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

1. רטוב
2. רטוב
3. 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.

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

1. רטוב
2. רטוב

## 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).

סעיף (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.

בהמשך השאלה נבחן וריאציה לאלג׳ IDA\* שמטרתה להתמודד עם הבעיה עליה נשאלתם בסעיף ב׳, תוך הקרבה של איכות הפתרון.

נתונים:

* נתון עבור מרחב חיפוש בעל מקדם סיעוף חסום (כרגיל).
* נסמן ב- את אוסף הקשתות בגרף המצבים של .
* נתונה עבור פונק׳ עלות והיוריסטיקה **קבילה**.
* נתון כי עלות פתרון אופטימלי במרחב הינו .
* נתון קבוע עבורו מתקיים .

הגדרות נוספות:

נגדיר את הקבוצה הבאה: . בנוסף נגדיר פונק׳ באופן הבא:

*ניסוח שקול אחר:*

לדוגמא, עבור מקבלים את הקב׳ . להלן תרשים המתאר את הפונק׳ בקטע :

*A picture containing text

Description automatically generated*

בהינתן אלג׳ חיפוש ומרחב , נגדיר את להיות עלות הפתרון שמוחזר ע״י האלג׳ בריצתו על המרחב . כמו כן, נגדיר: .

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).

## חלק י' – הגשת המטלה

* **יש לכתוב קוד ברור:**
  + קטעי קוד מסובכים או לא קריאים יש לתעד עם הערות.
  + לתת שמות משמעותיים למשתנים.
* **הדו"ח:**
  + יש לכתוב בדו"ח את תעודות הזהות של **שני** המגישים.
  + הדו"ח צריך להיות מוקלד במחשב ולא בכתב יד. הדו״ח צריך להיות מוגש בפורמט PDF (לא נקבל דוחות שהוגשו בפורמט וורד או אחרים).
  + יש לשמור על סדר וקריאות גם בתוך הדו"ח.
  + אלא אם נכתב אחרת, תשובות ללא נימוק לא יתקבלו.
  + יש לענות על השאלות לפי הסדר ומספרי הסעיפים שלהם.
* **ההגשה:**
* יש להעלות לאתר קובץ zip בשם AI1\_123456789\_987654321.zip (עם תעודות הזהות שלכם במקום המספרים).
* בתוך ה- zip צריכים להיות זה לצד זה:
  + הדו"ח הסופי בפורמט PDF בשם: AI1\_123456789\_987654321.pdf.
  + תיקיית הקוד ai\_hw1 שקיבלתם בתחילת המטלה, עם כל השינויים הנדרשים.

**נא לא להכניס ל- zip את התיקייה db שבתיקייה שקיבלתם – אנא מחקו אותה משם.**

שימו לב: הקוד שלכם ייבדק ע״י מערכת בדיקות אוטומטיות תחת מגבלות זמני ריצה. במידה וחלק מהבדיקות יכשלו (או לא יעצרו תוך זמן סביר), הניקוד עבורן יורד באופן אוטומטי. לא תינתן הזדמנות להגשות חוזרות. אנא דאגו לעקוב בהדיקות אחר הוראות ההגשה. שימו לב כי במהלך חלק מהבדיקות ייתכן שחלק מהקבצים שלכם יוחלפו במימושים שלנו. אם עקבתם אחר כל הדגשים שפורטו במסמך זה -עניין זה לא אמור להוות בעיה.

לא תתאפשרנה הגשות חוזרות, גם לא בגלל טעות טכנית קטנה ככל שתהיה. אחריותכם לוודא טרם ההגשה שהתרגיל רץ בסביבה שהגדרנו ושהקוד עומד בכל הדרישות שפירטנו.

אנא עברו בשנית על ההערות הטכניון שפורסמו בתחילת מסמך זה. וודאו שאתם עומדים בהם.

שימו לב: **העתקות תטופלנה בחומרה.** אנא הימנעו מאי-נעימויות.

מקווים שתהנו מהתרגיל!

בהצלחה!