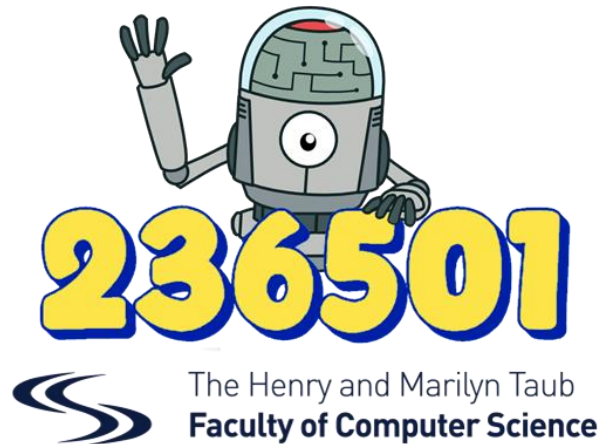


תרגיל בית 1

מרחבי חיפוש, אלגוריתמי חיפוש מיועד ולא מיועד, חיפוש לוקאלי



מטרות התרגיל:

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

הנחיות כלליות

- **תאריך הגשה:** יום רביעי, 10.7, בשעה 23:59.
- את המטלה יש להגיש **בזוגות בלבד**.
- יש להגיש מטלות מוקדמות **בלבד** בעברית או באנגלית. פתרונות בכתב יד **לא ייבדקו**.
- שאלות בנוגע לתרגיל בכל נושא שהוא יש להעלות לפורום המיועד **בפיאצה**.
- המתרגל האחראי על תרגיל: **אלון הכהן**.
 - בקשות דחיה **מוצדקות** (מילואים, אשפוז וכו') יש לשלוח למתרגל האחראי (אור רפאל בידוסה).
 - במהלך התרגיל ייתכן כי קבצי התרגיל (כולל מסמך זה) יעודכנו. הודעה על כך תפורסם בהתאם.
 - העדכונים הינם **מחייבים**, ועליכם להתעדכן עד מועד הגשת התרגיל.

- שימו לב, התרגיל מהווה כ-10% מהציון הסופי במקצוע ולכן העתקות תטופלנה בחומרה!
- ציון המטלה יורכב מהגורמים הבאים:

▪ 65% - המסמך היבש.

▪ 35% - הקוד המוגש.

- אנו יודעים שעבור חלקכם זו התנסות ראשונה בכתיבת קוד בפיתוח, והתרגיל מתוכנן בהתאם לכך.
 - שימו לב כי לא יענו שאלות בסגנון "איך מוצאים את עלות הפתרון שהוחזר?" / "איך ניגשים למפה מתוך המימוש של הפונק' ההיא?" / "באיזה שדה שמור ה...?" וכדומה.
 - 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 תינתנה דחיות על רקע זה.
 - מסמך זה כתוב בלשון זכר מטעמי נוחות בלבד, אך מתייחס לכלל המגדרים.
- כפי שצוין, אנחנו קשובים לפניית שלכם במהלך הזמן המוקצב להגשת התרגיל ומעדכנים את המסמך בהתאם. בעת הצורך, גרסאות עדכניות של המסמך יועלו לאתר הקורס, ולכל עדכון יצורף מספר גרסה כדי שתוכלו לעקוב. ייתכן שתפורסמה מספר גרסאות – אל תיבהלו מכך, השינויים בכל גרסה יכולים להיות מינוריים.
- שימו לב כי **הבהרות ועדכונים שנוספים למסמך אחרי הפרסום הראשוני יסומנו כאן בצהוב**. הנחיות **מחייבות** כלליות נוספות יופיעו תחת פוסט נעוץ בפורום הפיאה בהתאם לצורך.

הנחיות לחלק היבש

1. ככלל אצבע, בהינתן שאלה ראשית ספקו את התשובה המיידית ולאחר מכן הרחיבו והסבירו. למשל, אם השאלה היא מה סיבוכיות הזמן של אלגוריתם BFS, תשובה מספקת לדוגמה תהיה " $O(b^d)$ ", מכיוון שבמקרה הגרוע ביותר נאחסן את כל עץ החיפוש של הבעיה ב-CLOSE.
2. ישנן שאלות עם יותר מפתרון יחיד. כל פתרון נכון יקבל את מלוא הנקודות, בהנחה שהוא אכן מדויק ומנומק היטב.

הנחיות לחלק הכולל קוד

1. אנו מעודדים אתכם לעבור על הקבצים המצורפים ולהבין כיצד הסביבה בנויה ובאילו פונקציות תוכלו להשתמש במימוש שלכם.
2. הקוד שלכם ייבדק בקפדנות על ידי טסטים. הטסטים יבדקו את הפתרונות המוחזרים על ידי האלגוריתמים שלכם אל מול המימוש שלנו על פני בעיות שונות. אנו מצפים ממכם (אלא אם צוין אחרת) להחזיר את אותם ערכים בדיוק. אנחנו נבדוק את **המסלול המוחזר**, **מספר הצמתים שפותחו** ואת **עלות הפתרון המוחזר**. שימו לב כי הטסטים יהיו מוגבלים בזמן אך תקבלו זמן גדול מאוד לכל טסט.
3. ספקו קוד ברור ונקי הכולל תיעוד המאפשר גם בדיקה ידנית.

הוראות הגשה

- עליכם להגיש קובץ יחיד בשם `AI1_<ID1>_<ID2>.zip` (כאשר `<IDX>` מתאר את ת"ז של כל מגיש/ה), המכיל:
 - קובץ בשם `dry.pdf` המכיל את התשובות לחלק היבש.
 - קובץ בשם `Algorithms.py` המכיל את המימוש שלכם לאלגוריתמי החיפוש.

מבוא ורקע

תרגיל הבית מתפרש על פני מסמך זה והמחברת המצורפת IAI_Spr24_HW1.ipynb_236501. מומלץ בחום לענות על השאלות לפי הסדר המתואר במסמך זה.

במטלה זו נעסוק בהפעלת אלגוריתמי חיפוש על מרחבי מצבים לבעיות ניווט. מומלץ לחזור על שקפי ההרצאות והתרגולים הרלוונטיים לפני תחילת העבודה על התרגיל. במהלך התרגיל תתבקשו להריץ מספר ניסויים ולדווח על תוצאותיהם. לשם כך תידרשו לבצע ניתוח של התוצאות שהתקבלו, כפי שיוסבר בהמשך המסמך.

סיפור מסגרת



איימי היא סטודנטית מצטיינת למדעי המחשב הלומדת ב-GTIIT, המכון הטכנולוגי של גואנגדונג והטכניון שבסיין. כל חייה רצתה איימי לראות חזיר בר במציאות, ומכיוון ששמעה שחזירי הבר היפים ביותר מתגוררים ברכס הכרמל, החליטה כי לקראת סיום התואר הראשון שלה היא תגיע לחיפה לסמסטר של חילופי סטודנטים. בזמן שבגתה את המערכת שלה לסמסטר, היה לאיימי ברור כי אין סיכוי שהיא מוותרת על הקורס "מבוא לבינה מלאכותית". לאחר שנחתה בשלום בישראל והתמקמה במעונות, הגיעה איימי ביום פתיחת הסמסטר לבניין טאוב שמחה ומתרגשת לקראת ההרצאה הראשונה של הקורס – אך להפתעתה הרבה, **לא ראתה אף סטודנט בבניין!** כשהלכה לשאול את מזכירות לימודי הסמכה, גילו לה כי מכיוון שבונים בניין חדש לפקולטה, כל השיעורים הועברו מבניין טאוב אל בניינים אחרים ברחבי הקמפוס... אחרי חיפושים רבים באתר הקורס, בציפורק ואפילו במודל – גילתה איימי כי ההרצאה הועברה לבניין אמרסון (אי שם בצד השני של הטכניון). אבל איך אפשר להגיע לשם, כשכל הקמפוס מלא בחפירות ובשיפוצים?

מתוסכלת ומבולבלת, איימי נתקלת בכס ביציאה מטאוב, ועם דמעות בעיניים שואלת איך מגיעים לשם (בלי ליפול לתוך בור חפירה בדרך, כמובן). למזלה של איימי, אתם הרי סטודנטים עם ותק בטכניון, וכמובן שאתם יודעים איפה נמצא בניין אמרסון! אתם מרגיעים את איימי ואומרים לה כי לא רק שאתם יודעים את הדרך, אלא שאחרי שכבר למדתם אלגוריתמים רבים בקורס – **גם תעזרו לה לתכנן את המסלול הטוב ביותר להגיע מבניין טאוב לבניין אמרסון.**



הגדרת סביבת העבודה עם Jupyter

לפני שתחילו לפתור את התרגיל, נבקש **שתקדישו מספר דקות** לוודא כי סביבת העבודה שלכם מוגדרת היטב, לפי ההוראות המובאות כאן. לאורך התרגיל נעבור לסירוגין בין עבודה על שאלות תיאורטיות לבין מימוש האלגוריתמים שלמדנו בכיתה עבור פתרון הבעיה אשר הוצגה במבוא. על מנת לאפשר עבודה נוחה וממוקדת במימוש האלגוריתמים עצמם, אנו מספקים לכם את המחברת **236501_IAI_Spr24_HW1.ipynb** המאפשרת הרצה נוחה של האלגוריתמים השונים וכן ויזואליזציה של פעולתם. המחברת הנ"ל היא מסוג Jupyter Notebook, המאפשרת הרצת קוד פייתון באופן שמותאם לניסויים כך שניתן לשנות שינויים מינוניים בקוד וכן להריץ מספר רב של ניסויים בזה אחר זה באופן אינטראקטיבי. הקובץ הנ"ל המשמש כמעין "מחברת" שיאפשר לכם להריץ את הניסויים שלכם לאורך התרגיל בצורה קלה. אם אתם מרגישים צורך בחיזוק וחיזוק היכולות של Jupyter, תוכלו לעבור שוב על מחברת התרגול העצמי בפייתון שבאתר הקורס. על מנת להימנע מתקלות הנובעות מהגדרת סביבת הפייתון שבה תעבדו, אנו ממליצים לעבוד עם **Google Colab**. כדי לאפשר עבודה בפלטפורמה עליכם להעלות את תוכן התיקיה של התרגיל לתוך תיקייה ב-Drive האישי שלכם, ולאחר מכן לפתוח את המחברת (קובץ ה-`ipynb`) ולפעול על פי ההוראות. צפו בבקשה **בסרטון ההדרכה הבא** על מנת למנוע אי-נעימויות בכל הקשור לתקלות הנובעות מהגדרת הסביבה. **שימו לב** כי גרסת הפייתון איתה אתם נדרשים לעבוד היא **python 3.10**. גרסאות אחרות עלולות להוביל לתקלות שנובעות מאי-תאימות במהלך ההתקנה של הספריות הנוספות הנדרשות בתרגיל שאינן נכללות בגרסת הפייתון הבסיסית, בפרט:

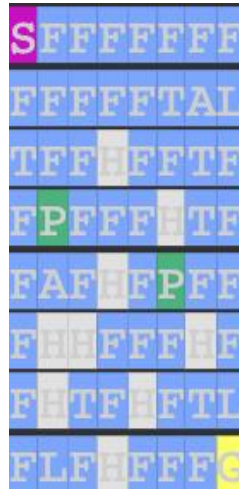
Gym
Numpy
Heapdict
Typing

- **אין לעשות** `import` לספריות שלא הוגדרו בקוד שסופק לכם ושלא ניתן להן אישור מפורש בפורום **הפיאצה**.
- התחילו לענות על חלק זה רק לאחר שהגדרתם את סביבת העבודה והבנתם כיצד להריץ את הקוד במלואו.
- מומלץ לעבור על הקוד במחברת במקביל למסמך היבש, ולענות על השאלות השונות שלב-שלב.

מרחב החיפוש

השאלות בחלק זה מתבססות על המפה "8x8" שמופיעה במחברת, אלא אם נכתב אחרת.

שאלה 1 – מבוא והיכרות עם מרחב החיפוש (9 נק') :



1. **קוד** : עברו על המחברת עד שאתם מגיעים לחלק של DFS-G ועצרו שם (לפני שתממשו את האלגוריתם הראשון במחברת).
השאלות בחלק זה מתבססות על המפה "8x8" שמופיעה במחברת (וכן בעמוד הקודם), אלא אם נכתב אחרת.
2. תחילה נרצה להגדיר את מרחב החיפוש כפי שנלמד בתרגול. נגדיר:
 $S = X^d$ = מרחב המצבים, אשר כל מצב בו מתואר על ידי וקטור של d פיצורים.
 O = מרחב האופרטורים
 I = המצב ההתחלתי
 G = קבוצת מצבי המטרה
הגדירו את (S, O, I, G) עבור סביבת הקמפוס. (1 נק')
3. מה גודל מרחב המצבים S ? הסבירו. (1 נק')
4. מה תחזיר לנו הפונקציה Domain על האופרטור DOWN (אופרטור 0) (1 נק').
5. מה תחזיר לנו הפונקציה Succ על המצב ההתחלתי 0? (1 נק')
6. האם קיימים מעגלים במרחב החיפוש שלנו? (1 נק')
7. מה הוא מקדם הסיעוף בבעיית הניווט בקמפוס? (1 נק')
8. עבור המפה "4x4" שמופיעה במחברת, ציירו את גרף המצבים. (1 נק')
9. במקרה הגרוע ביותר, כמה פעולות ידרשו לסוכן Random Agent להגיע למצב הסופי? ובמקרה הטוב ביותר? (1 נק')
10. עבור מפה כללית בסביבת הקמפוס, בה יכולים להיות מספר מצבי מטרה (לדוגמה במקרה שבו איימי תתרחק גם אם תגיע לחומס בבית הסטודנט), האם המסלול הזול ביותר (מבחינת עלות המסלול) הוא גם המסלול שמגיע למצב המטרה הקרוב ביותר למצב ההתחלתי (במונחים של Manhattan Distance, כפי שנלמד בכיתה)? אם כן, הוכיחו. אם לא, ספקו דוגמה נגדית. (1 נק')

חיפוש לא מיועד

השאלות בחלק זה מתבססות על המפה "8x8" שמופיעה במחברת, אלא אם נכתב אחרת.
שימו לב – כאשר אתם נשאלים לגבי היווצרות ופיתוח של צמתים בעת הריצה של אלגוריתם כלשהו:

- הניחו כי צומת התחלתי נוצר אוטומטית עם תחילת הריצה של האלגוריתם.
- התייחסו לסדר היצירה והפיתוח של הצמתים תוך כדי ריצת האלגוריתם כפי שהוא מוגדר לפי הפעולות השונות במחברת-
 - Down – 0 (יפותח ראשון)
 - Right – 1
 - Up – 2
 - Left – 3 (יפותח אחרון)

שאלה 2 – Breadth First Search (BFS) (6 נק') :

- עבור בעיית הניווט הקמפוס עם מפה $N \times N$, האם האלגוריתם שלם? האם הוא קביל? (1 נק')
 - מה צריך להיות התנאי על גרף החיפוש (לא בהכרח בבעיית הניווט הקמפוס) כך שאלג' BFS (שרץ על עץ) ואלג' BFS-G (שרץ על גרף) ייצרו ויפתחו צמתים זהים באותו הסדר? (1 נק')
 - נתונה מפה בגודל $N \times N$ שלא מכילה portals. הציעו דרך להשתמש באלגוריתם BFS-G כך שיחזיר פתרון אופטימלי (עלות מינימלית) והסבירו בקצרה. (2 נק')
- **רמז:** מצאו פונקציה $T: G \rightarrow G'$ המקבלת את גרף המצבים G ויוצרת גרף חדש G' , ובעזרתה למצוא את המסלול האופטימלי בגרף G .
- נתונה מפה בגודל $N \times N$, ללא חורים, ללא Portals, המכיל $2 - N^2$ משבצות רגילות (F, T, A, L), מצב התחלתי בפינה השמאלית עליונה ומצב מטרה בפינה הימנית תחתונה. כמה צמתים יפותחו וייווצרו במהלך חיפוש BFS-G? הסבירו. (2 נק')

שאלה 3 – Depth First Search (DFS), וריאציות על DFS (23 נק') :

- קוד:** ממשו את הפונקציות הנדרשות עבור ריצתו של האלג' DFS-G בקובץ `Algorithms.py`, ע"פ ההנחיות המופיעות במחברת. (10 נק')
- עבור בעיית הניווט הקמפוס עם מפה $N \times N$, האם האלגוריתם שלם? האם הוא קביל? (1 נק')
- עבור בעיית החיפוש בקמפוס, נתונה מפה בגודל $N \times N$. האם אלגוריתם DFS (שרץ על עץ), עבור בעיית הניווט הקמפוס על מפה $N \times N$, היה מוצא פתרון כלשהו? אם כן, מה המסלול שיתקבל? אם לא, כיצד האלגוריתם היה פועל? (1 נק')
- כעת נתונה מפה בגודל $N \times N$, ללא Holes וללא Portals. המפה מכילה $2 - N^2$ משבצות "רגילות" (F/T/A/L), מצב התחלתי S בפינה השמאלית העליונה שלה ומצב מטרה S בפינה הימנית התחתונה שלה.
 - כמה צמתים יפותחו וייווצרו במהלך חיפוש DFS-G? (2 נק')
 - כמה צמתים יפותחו וייווצרו במהלך חיפוש DFS-G **backtracking**? הסבירו בתשובתכם מהו היתרון בשימוש באלג' זה על פני שימוש באלג' DFS-G מתת הסעיף הקודם. (2 נק')
- איימי רוצה למצוא מסלול בסביבת הקמפוס עם DFS-L. ידוע כי אורך המסלול הקצר ביותר לצומת מטרה הוא d אך איימי מחליטה להגביל את עצמה לחיפוש בעומק $\frac{d}{2}$ בלבד.
 - עבור מפה כללית בגודל $N \times N$, הציעו שינוי ל**בעיית החיפוש** (S, O, I, G) כך שאיימי תוכל למצוא פתרון מבלי להפר את מגבלת העומק. הסבירו למה כעת ניתן למצוא פתרון. (3 נק')
 - האם השתנה מקדם הסיעוף? אם כן, מה מקדם הסיעוף החדש b' ? רשמו את התשובה כתלות ב **b** (מקדם הסיעוף בבעיה המקורית). (1 נק')
 - מהן סיבוכיות הזמן והמקום החדשים? ענו במונחים של b, d והשוו את התשובה ל-DFS-L רגיל עם עומק d . (1 נק')
 - ספקו דוגמה לבעיה שבה DFS-L במרחב החיפוש החדש (לאחר השינויים שביצעתם בסעיף 5.1) טובה יותר מאשר DFS-L במרחב החיפוש הקודם, ודוגמה לבעיה שבה DFS-L במרחב המקורי עדיף. בתשובתכם התייחסו למספר הצמתים שפותחו. דוגמות יכולות להיות כלליות ולא בהכרח מסביבת הקמפוס. (2 נק')

שאלה 4 – Uniform Cost Search (UCS) (14 נק') :

1. **קוד:** ממשו את הפונקציות הנדרשות עבור ריצתו של האלג' UCS בקובץ `Algorithms.py`, ע"פ ההנחיות המופיעות במחברת. (10 נק')
 2. האם בבעיית החיפוש שלנו, עבור מפה $N \times N$, האלגוריתם הוא שלם? האם הוא קביל? (1 נק')
 3. עבור אילו בעיות חיפוש אלגוריתם UCS ואלגוריתם BFS-G יפעלו באותו האופן? הסבירו. (1 נק')
 4. איימי טעתה במימוש של אלגוריתם UCS ובדקה בטעות בזמן יצירת הצומת האם הוא צומת מטרה במקום בזמן הפיתוח שלו (כלומר, לאחר הוצאתו מתור העדיפויות). תנו דוגמה לגרף חיפוש שעבורו האלג' שאיימי מימשה יחזיר בכל זאת את המסלול הקל ביותר, ודוגמה לגרף חיפוש שעבורו האלג' לא יחזיר את המסלול הקל ביותר. (2 נק')
- עבור כל דוגמה, הסבירו מה המסלול והעלות ש-UCS השגוי החזיר, ומה המסלול והעלות שהאלגוריתם הנכון היה מחזיר. נדגיש שגרף החיפוש לא בהכרח צריך לייצג את בעיית הניווט הקמפוס – ניתן לתת דוגמה לגרף שמייצג בעיית חיפוש אחרת.
 - על הגרפים המוצעים להכיל קשתות מכוונות וכן את העלות של כל קשת.

חיפוש מיוחד

השאלות בחלק זה מתבססות על המפה "8x8" שמופיעה במחברת, אלא אם נכתב אחרת.

שאלה 5 – יוריסטיקות (8 נק') :

1. תהיינה שתי יוריסטיקות קבילות h_1, h_2 .
 1. הוכיחו/הפריכו: היוריסטיקה $h = h_1 + h_2$ קבילה. (1 נק')
 2. הוכיחו/הפריכו: היוריסטיקה $h = \frac{h_1 + h_2}{2}$ קבילה. (1 נק')
2. תהיינה שתי יוריסטיקות עקביות h_1, h_2 .
 1. הוכיחו/הפריכו: היוריסטיקה $h = h_1 + h_2$ עקבית. (1 נק')
 2. הוכיחו/הפריכו: היוריסטיקה $h = \frac{h_1 + h_2}{2}$ עקבית. (1 נק')
3. נגדיר יוריסטיקה חדשה עבור בעיית הניווט בקמפוס:
$$h_{CAMPUS}(s) = \min\{\min\{h_{Manhattan}(s, g) \mid g \in G\}, C_{portal}\}$$
כאשר הביטוי $h_{Manhattan}(s, g)$ מתאר את מרחק מנהטן מהמצב הנוכחי למצב סופי, והביטוי C_{portal} מתאר את מחיר השימוש ב-Portal (לדוגמה, 100 בבעיית הניווט במקפוס). שימו לב כי היוריסטיקה מחשבת את מרחקי מנהטן מהמצב הנוכחי למצב סופי על פני כל צמתי היעד.
4. האם היוריסטיקה h_{CAMPUS} קבילה עבור כל מפה בבעיית הניווט בקמפוס? אם כן, הסבירו בקצרה. אם לא, ספקו דוגמה נגדית. (2 נק')
5. האם היוריסטיקה h_{CAMPUS} עקבית עבור כל מפה בבעיית הניווט בקמפוס? אם כן, הסבירו בקצרה. אם לא, ספקו דוגמה נגדית. (2 נק')

שאלה 6 – Greedy Best First Search (3 נק') :

1. האם האלג' Greedy Best First Search, על מפה כללית עבור בעיית הניווט בקמפוס בגודל $N \times N$, הוא שלם? האם הוא קביל? (1 נק')
2. תנו יתרון וחיסרון של האלג' Greedy Best first Search ביחס ל-Beam Search. בתשובה התייחסו להגדרות השלמות והעקביות ולסיבוכיות הזמן והזיכרון. תוכלו להתייחס לבעיית חיפוש כללית, ולא ספציפית עבור בעיית הניווט בקמפוס. (2 נק')

שאלה 7 – A^* , וריאציות על A^* (21 נק') :

1. **קוד:** ממשו את הפונקציות הנדרשות עבור ריצתו של האלג' $W-A^*$ בקובץ `Algorithms.py`, ע"פ ההנחיות המופיעות במחברת. השתמשו ביוריסטיקה h_{CAMPUS} . (10 נק')
2. בכיתה הגדרנו את פונקציית ההערכה עבור הצומת הבא לפיתוח במהלך ריצת האלג' A^* באופן הבא:
$$f(v) = h(v) + g(v)$$
איימי טוענת כי אפשר להשתמש בפונקציה $f'(v) = \frac{h(v) + g(v)}{2}$ במקום ב- $f(v)$, ותוצאת הריצה של האלג' תהיה שקולה.

הסבירו בקצרה מדוע איימי צודקת (בתשובתכם התייחסו לסדר פיתוח הצמתים, המסלול המוחזר ועלות המסלול המוחזר מהאלג' בעת

השימוש בפונקציה $f'(v)$). (2 נק')

3. **קוד**: ממשו את הפונקציות הנדרשות עבור ריצתו של האלג' A^* בקובץ `Algorithms.py`, ע"פ ההנחיות המופיעות במחברת. גם עבור אלג' זה,

השתמשו ביוריסטיקה h_{CAMPUS} . (5 נק')

▪ **רמז**: חשבו כיצד תוכלו לממש את האלג' A^* ע"י שימוש במימוש שהצעתם עבור $W-A^*$.

4. תנו יתרון וחיסרון של האלג' $ID-A^*$ ביחס ל- A^* . באילו מקרים הייתם מעדיפים להשתמש בכל אחד מהם? (2 נק')

5. תנו יתרון וחיסרון של האלג' $A^*-\epsilon$ ביחס ל- A^* . באילו מקרים הייתם מעדיפים להשתמש בכל אחד מהם? (2 נק')

ניתוח התוצאות

שאלה 8 – Benchmarking (4 נק') :

בשאלה זאת נשווה בין אלגוריתמי החיפוש השונים שמימשתם על בעיות שונות.

1. **קוד**: הריצו את החלק הרלוונטי במחברת וודאו כי אתם רואים את קובץ ה-`csv` שנוצר (תוכלו לפתוח אותו עם Excel או כל תוכנה אחרת מתאימה).

2. **הסבירו את התוצאות**. האם הן תואמות לציפיות שלכם? האם התוצאות היו משתנות עם יוריסטיקה מיוחדת יותר? נתחו והסבירו את

התוצאות במונחים של מספר פיתוחים, המסלול המוחזר ועלות הפתרון (מומלץ לצרף גם ניתוח ויזואלי כלשהו, כגון גרף או טבלה).

שימו לב שבסעיף זה אין תשובה נכונה או לא נכונה, אך אתם נדרשים לספק הסבר מפורט ומבוסס. (4 נק')

חיפוש לוקאלי

שאלה 9 – Local Search (12 נק') :

עכשיו כשאתם ואיימי כבר חברים, היא מציעה לעזור לכם ללמוד סינית! אתם מסכימים, והיא שולחת לכם מסמך ובו כל מיני משפטים בסינית שתוכלו לתרגל. במשך שבוע אתם מנסים לקרוא את המסמך אך ללא הצלחה (מה לעשות שזה מרגיש לכם כמו סינית), עד שאתם מבקשים את עזרתה של איימי. כשהיא פותחת את המסמך במחשב שלכם, היא רואה שבגלל שהמחשב שלכם בעברית – כל המילים במסמך התבלבלו... למשל, אם הופיע במסמך המקורי המשפט: "我的名字叫艾米" ("קוראים לי איימי"), במסמך שבמחשב שלכם יופיעו המילים בסדר אחר, למשל: "名字我的艾米" (לי קוראים איימי).

הערות:

- הניחו כי במסמך המקורי כל מילה מופיעה פעם אחת בלבד.
- המילים מתערבבות בכל המסמך ולא רק בתוך כל משפט. הפרמוטציות של המילים במסמך הינן רנדומליות לחלוטין (ולא קורות לפי איזה סדר קבוע כלשהו).

מכיוון שאתם עדיין לא יודעים סינית, אתם לא מצליחים לסדר את המילים בסדר הנכון לבדכם. למזלכם, איימי נותנת לכם פונקציית ערך (utility function) המקבלת את המסמך ומחזירה את מספר המילים שנמצאות במיקומן המקורי במסמך (ערכים בין 0 ל- n), וכן אופרטור הלוך שתי מילים (לאו דווקא רצופות) ומחליף בין מיקומיהן. אחרי כל מה שלמדתם בהרצאות ובתרגולים של "מבוא לבינה מלאכותית", אתם רוצים להפעיל אלגוריתם חיפוש לוקאלי אשר משתמש בפונקציית הערך ובאופרטורים הנ"ל שהפתרון שלו יהיה הסידור הנכון של המילים במסמך.

1. כיצד יש להגדיר את המצבים במרחב החיפוש? (2 נק')
2. מהו מספר המצבים במרחב החיפוש? (1 נק')
3. אתם יודעים כי איימי קיבלה 100 בתרגיל בית 1 של "מבוא לבינה מלאכותית", ולכן מבקשים את עזרתה גם בהפעלת האלגוריתם. היא מציעה לכם להשתמש ב-SAHC (Steepest Ascent Hill Climbing), על מנת למצוא פתרון. האם האלגוריתם Steepest Ascent Hill Climbing בהכרח ימצא פתרון? (2 נק')
4. איימי סיפרה לכם שאלון המתרגל יודע גם הוא קצת סינית. בשעת הקבלה אתם נעזרים בו בפתרון הבעיה, והוא מייעץ לכם להשתמש ב-SAHC with sidesways steps.
 1. האם האלגוריתם SAHC with sidesways steps בהכרח ימצא פתרון? (2 נק')
 2. אלון טוען כי עבור המקרים בהם SAHC ו-SAHC with sideways steps מוצאים פתרון, עדיף להשתמש ב-SAHC with sideways steps כי ייתכן שהוא ימצא פתרון בפחות צעדים. האם אלון צודק? (3 נק')
5. אחרי דיון סוער בין אלון לאיימי, הגעתם לפשרה ומחליטים להשתמש באלגוריתם Stochastic Hill Climbing. האם האלגוריתם Stochastic Hill Climbing בהכרח ימצא פתרון? (2 נק')