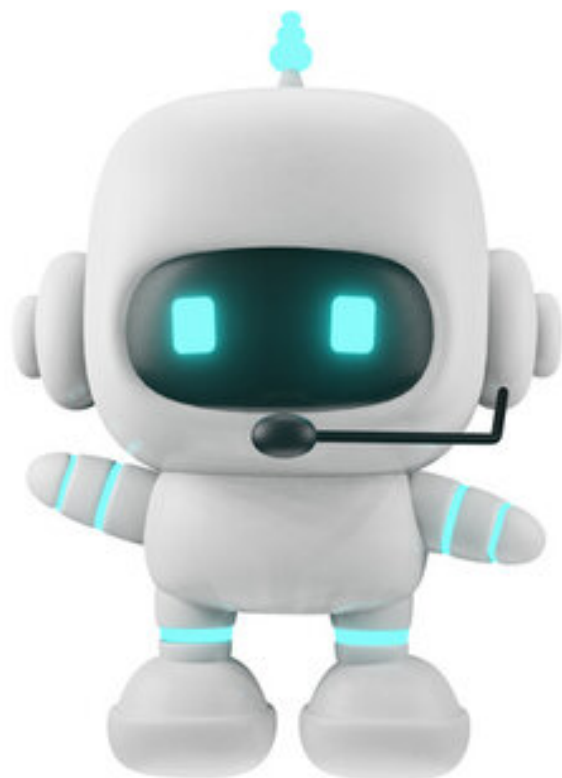


תרגיל בית 2 מבוא לבינה מלאכותית  
חיפוש רב סוכנים  
בינה מחסנים



## הקדמה ואדמיניסטרציה

### הנחיות כלליות

- תאריך הגשת התרגיל: 12/1/2025
- את המטלה יש להגיש בזוגות בלבד – בקשות להגשה ביחידים באישור המתרגל האחראי בלבד (אור רפאל בידוסה).
- יש להגיש את המטלה מוקלדת בלבד – פתרון בכתב יד לא ייבדק.
- התשובות צריכות להיות כתובות בשפה העברית או באנגלית.
- אפשר לשלוח שאלות בנוגע לתרגיל דרך הפיאצה.
- המתרגלת האחראית על התרגיל: שירה מאירי.
- בקשות דחייה מוצדקות יש לשלוח למתרגל האחראי על הקורס בלבד (אור רפאל בידוסה).
- במהלך התרגיל ייתכן שיעלו עדכונים – הודעה תפורסם בהתאם במקרה זה.
- העדכונים מחייבים וזוהי אחריותכם להתעדכן לגביהם עד מועד הגשת התרגיל. עדכונים יופיעו בטופס בצבע צהוב.
- העתקות טטופלנה בחומרה.
- ציון המטלה כולל חלק יבש וחלק רטוב. בחלק היבש נבדוק שתשובתכם נכונה, מלאה, קריאה ומסודרת. בחלק הרטוב הקוד שלכם ייבדק הן על הגבלת הזמן שתפורט בהמשך ועל אחוזי ההצלחה של האלגוריתמים שלכם לעומת אלו שאנו נממש כבדיקה.
- מומלץ להסתכל בקוד בעצמכם. שאלות בסיסיות על פייתון שלא נוגעות לתרגיל כדאי לבדוק באינטרנט לפני שאתם שואלים בפיאצה. מומלץ לקרוא את הקוד הנתון על מנת להבין את אופן פעולתו – במקרה שישנם דברים לא מובנים. (לשם כך יש הערות רבות ואף הסבר מורחב על הסביבה!)
- מומלץ לא לדחות את התרגיל לרגע האחרון מאחר שהמימוש וכתיבת הדו"ח עלולים לקחת יותר זמן מהצפוי.
- התייחסות בלשון זכר, נקבה או רבים מתייחסים כלפי כלל המגדרים.
- אין צורך להשתמש בתהליכים ובמאפיינים מורכבים של מערכות הפעלה, התרגיל בית אמור לרוץ על כל מערכת הפעלה ולא למערכות הפעלה ספציפיות. אם ישנה בעיה פרטנית, שלחו מייל למתרגלת האחראית על התרגיל.
- הספריות המותרות לשימוש הן: random, time, math. כל שימוש בספרייה נוספת יצריך אישור. שימוש לא תקני בספריות יוביל לפסילת התרגיל.
- גרסת הפייתון איתה אתם נדרשים לעבוד היא 3.10

הוראות הגשה – כל הגשה שלא בפורמט, לרבות תיקיות, תתי תיקיות או קבצים נוספים – לא

תיבדק ותזכה את ההגשה בציון 0 אוטומטי.

בתוך קובץ זיפ עם השם: HW2\_AI\_id1\_id2.zip

את הדו"ח היבש בפורמט הבא: id1\_id2.pdf

ואת הקובץ: submission.py שבו אתם ממשים את האלגוריתמים

## הקדמה

"בינה מחסנים" רוצים להפוך את המחסן לאוטונומי והם מתלבטים מי הוא הרובוט אותו הם רוצים לשכור למשימה. על הרובוטים  $R_1$   $R_2$  להתחרות אחד בשני והמנצח יתקבל לעבודה. בתרגיל זה תממשו ותחקרו אלגוריתמי משחק סכום אפס אדוורסריאליים שלמדתם בהרצאות ובתרגולים.

## תיאור המשחק

המשחק מתרחש בלוח משבצות בגודל 5X5 כאשר על הלוח: 2 רובוטים, 2 תחנות הטענה, 2 חבילות ו-2 יעדים (לכל חבילה יעד משלה ולא ניתן להביא חבילה ליעד שלא שייך לה). לכל רובוט יחידות הטענה (battery), וניקוד (credit).

מטרת כל רובוט היא לצבור יותר ניקוד מהרובוט האחר עד סוף המשחק. המשחק נגמר כאשר לשני הרובוטים נגמרת הסוללה או כאשר לכל אחד מהרובוטים נגמר מספר הצעדים שהוקצה להם (ערך מוגדר מראש, דוגמה בהמשך).

רובוט צובר ניקוד כשהוא מביא חבילה ליעדה. כאשר רובוט מביא חבילה ליעד המתאים לה - הוא מקבל N יחידות ניקוד כאשר N הוא מרחק מנהטן בין מיקום החבילה המקורי ליעדה, כפול 2. כשרובוט מביא חבילה ליעדה, הרובוט השני מאבד את אותה כמות ניקוד כמו שהרובוט הראשון הרוויח (ניתן להגיע לניקוד שלילי).

לאחר שהחבילה מגיעה אל היעד שלה היא והיעד שלה נעלמים ובמקומם מופיעים יעד וחבילה חדשים. כלומר, בכל עת על הלוח יש בדיוק שתי חבילות ושני יעדים, אחד לכל חבילה.

רובוטים מתחילים עם טעינה מלאה וללא חבילה. כל פעולת תנועה של הרובוט עולה לו יחידת סוללה אחת. רובוט יכול לבצע למעלה, למטה, ימינה, שמאלה. בנוסף, **רובוט יכול להטעין את הסוללה בתחנת הטענה כאשר הוא נמצא במשבצת של תחנת הטענה ויש לו כמות יחידות ניקוד גדולה ממש מאפס (אחרת לא ניתן לבצע טעינה)**, הוא עושה זאת ע"י המרה של כל יחידות הניקוד שלו ליחידות טעינה. רובוט יכול לאסוף ולהוריד חבילה. רובוט אוסף חבילה כאשר הוא עומד באותה משבצת כמותה ומבצע פעולת אסיפה. רובוט מוריד חבילה כאשר הוא נמצא ביעד ומבצע פעולת הורדה. (אי אפשר להוריד חבילה שנאספה במשבצת שאינה היעד ואי אפשר לבצע איסוף במשבצת שאינה עם חבילה).  
סה"כ לרובוט 7 פעולות:

move north, move south, move east, move west, pick up, drop off, charge

שימו לב כי לא ניתן להפעיל את כל הפעולות על כל המצבים במרחב החיפוש. למשל לא נוכל לזוז ימינה אם משמעות הדבר היא יציאה מהלוח, תוכלו לחקור תוצאות של פעולות ולהבין את גבולות המשחק על ידי סוכן שנקרא hardcoded שתוכלו לתת לו מסלול ספציפי ולחקור את ההתנהגות שלו.

## הסבר על הסביבה והרצת / דיבוג המשחק

בשונה מתרגיל בית 1 הפעם תעבדו עם קבצי py ולא מחברות ipynb. מוזמנים לעבוד על התרגיל בכל IDE מתאים שנוח לכם לעבודה בפייתון, אנו ממליצים על pycharm של חברת JetBrains אליה יש לכם מנוי מטעם הטכניון.

הסביבה שאיתה תעבדו ממומשת בקובץ WarehouseEnv.py, מוזמנים לעיין בה.

בקובץ Agent.py ממומשים סוכנים אשר מהם הסוכנים שאתם תממשו יורשים, מומלץ להסתכל על הסוכנים הממומשים בקובץ בכדי להבין כיצד הם עובדים עם הסביבה. מומלץ להסתכל על הפונקציות `get_legal_operator` ו-`apply_operator` בכדי להבין את אופן ההתממשקות שלכם עם הסביבה. שימו לב שבקובץ submission יש סוכן שנקרא `hardcoded`, מטרתו לעזור לכם להבין את הסביבה. לפני שאתם שואלים לגבי כיצד הסוכן יתנהג אם יבצע פעולה בדקו בעצמכם בעזרת סוכן זה.

## הרצה

הרצת משחק נעשית ע"י שורת הפקודה הבאה שמריצים מהטרמינל:

```
python main.py greedy random -t 0.5 -s 1234 -c 200 --console_print --screen_print
```

- הארגומנט הראשון (במקרה הזה greedy) מציין את האלגוריתם שלפיו ישחק  $agent_0$
- הארגומנט השני (במקרה הזה random) מציין את האלגוריתם שלפיו ישחק  $agent_1$
- הגבלת זמן לצעד t - מקבלת ערך שמייצג את מספר השניות המקסימליות לצעד
- גרעין לקבלת ערך רנדומלי s - מקבל ערך שעוזר לחולל סביבה באופן רנדומלי, כאשר מועבר אותו ערך seed תחולל אותה סביבה
- מספר הצעדים המקסימלי עבור סוכן c -
- ערך `--console_print` דגל אופציונלי, כאשר מועבר מתבצעת הדפסה לקונסולה של המשחק שנראת כך :

```
robot 0 chose move north
[R1][P0][ ][C1][C0]
[D1][ ][ ][ ][ ]
[R0][P1][ ][ ][ ]
[ ][ ][ ][ ][ ]
[ ][ ][ ][D0][ ]
robots: [position:(0, 2) battery: 19 credit: 0 package: [None], position:(0, 0) battery: 20 credit: 0 package: [None]]
packages on street: [position:(1, 0) destination: (3, 4), position:(1, 2) destination: (0, 1), position:(0, 3) destination: (3, 3), position:(2, 3) destination: (0, 1)]
(1, 0) (3, 4) True
(1, 2) (0, 1) True
(0, 3) (3, 3) False
(2, 3) (0, 1) False
charge stations: [position:(4, 0), position:(3, 0)]
```

הפלט מכיל את מספר הסוכן ויכול להיות 0 או 1 והאופרטור בו בחר. לאחר מכן מודפס הלוח לאחר שהסוכן הפעיל את האופרטור. הלוח כולל 25 משבצות כאשר בכל אחת יכולים להיות רובוט, חבילה המחכה לרובוט או תחנת הטענה.

האותיות אשר מופיעות בלוח מסמלות:

R – Robot

C – Charge station

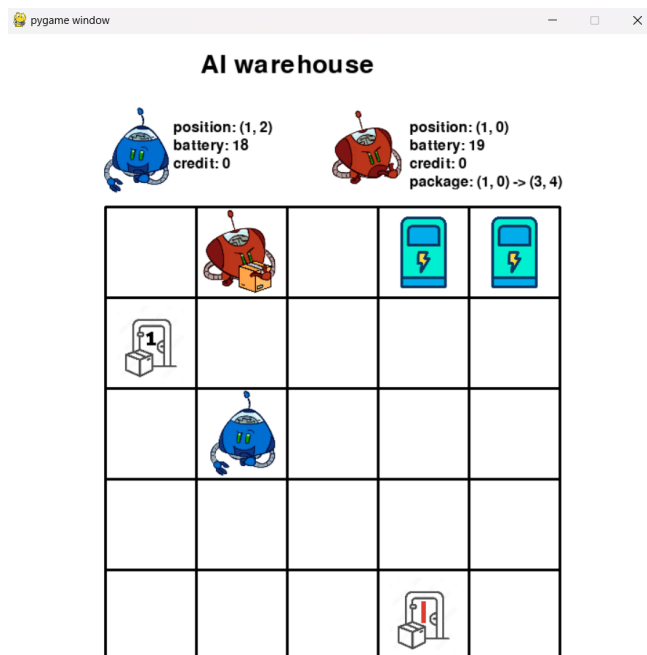
P – Package on street

D – Destination of Package on street

X – Destination of Package taken by Robot

המספר המופיע ליד כל אחת מהאותיות הוא המזהה של האובייקט אליו הוא משתייך – לדוגמא:  $R_1$  עבור הרובוט הראשון. עבור חבילה שנאספה ע"י רובוט, המספר ליד X הוא המזהה של הרובוט שאסף אותה. עבור חבילה במרחב, המספר ליד D הוא המזהה של החבילה במרחב. לעיתים מספר אובייקטים מופיעים באותו המיקום בתמונה ואז מודפס רק אחד מהם. במקרה זה, אפשר להשתמש ברשימות המפורשות שמופיעות אחרי הלוח. בנוסף שימו לב! מופיע לכם ברשימות את המקור והיעד של שתי החבילות הבאות שיופיעו על הלוח, מוזמנים ומומלץ להשתמש במידע זה בהמשך התרגיל.

- ערך --screen\_print דגל אופציונלי, כאשר מועבר יודפס לכם הנפשה של המשחק בחלון pygame שנראה כך:

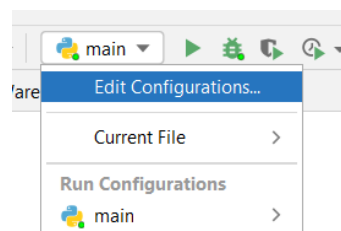


בנוסף בכדי לבדוק את הסוכנים שלכם על מספר רב של משחקים, אתם יכולים להוסיף את הדגל tournament -- וכך תקבלו הרצה רצופה של n משחקים. מוזמנים לשחק עם המשתנה של מספר המשחקים ולראות האם מקבלים התנהגויות רצויות עבור האלגוריתמים.

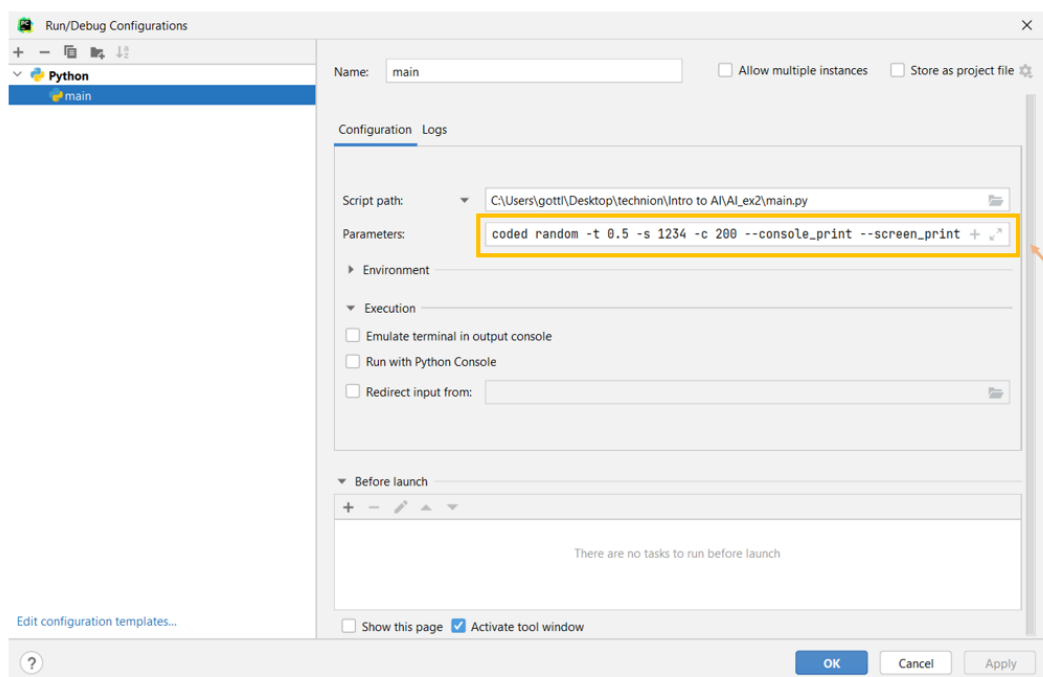
## דיבוג

בכדי לדבג את המשחק עליכם לעקוב אחר השלבים הבאים:

1.



2.



מתחילים לכתוב!

### חלק א' - ImprovedGreedy (18 נק')

1. (יבש: 4 נק') כפי שלמדתם, אנו מגדירים בעיית חיפוש במרחב המצבים עבור ריבוי סוכנים בתור  $\langle S, A, f, c, s_0, R \rangle$ . פרמלו את המשחק הנתון לכם בצורה זו (מספר מילים עבור כל אחד מהפרמטרים ומה הוא מייצג במשחק שלנו).
2. (יבש: 4 נק') הגדירו יוריסטיקה משלכם להערכת מצבי המשחק. עליכם לתעד אותה **בנוסחה מפורשת** ועלייה לכלול לפחות שלושה מאפיינים של הסביבה. **בחרו שמות ברורים בנוסחה שלכם**.
3. (רטוב: 10 נק') ממשו בקובץ submission.py את הפונקציה `smart_heuristic` שבה משתמש הסוכן AgentGreedyImproved **לפי היוריסטיקה שתיארתם בסעיף הקודם**.

## חלק ב' - RB-Minimax (22 נק')

1. (יבש: 3 נק') אנו רוצים להשתמש באלגוריתם מינימקס מוגבל משאבים.  
מה היתרונות והחסרונות של שימוש ב**יוריסטיקה קלה לחישוב** (ניתן לחשב אותה מהר) לעומת היוריסטיקה קשה לחישוב (אורכת זמן חישוב רב) בהינתן שהיוריסטיקה הקשה לחישוב יותר מיודעת מהקלה לחישוב?
  2. (יבש: 4 נק') חברתכם לקורס דנה מימשה סוכן minimax, היא שמה לב כי לעיתים הסוכן יכול לנצח תוך צעד אחד בלבד אך הוא בוחר בצעד אחר. האם **בהכרח** יש לה באג באלגוריתם? אם אין באג הסבירו מה באלגוריתם גורם להתנהגות שכזו. אם יש באג מה הוא יכול להיות?
  3. (יבש: 2 נק') הצצתם לסעיף הבא וראיתם כי עליכם לממש מינימקס מוגבל משאבים אך המשאב שמוגבל הוא אינו עומק החיפוש אלא זמן הריצה של האלגוריתם. הסבירו במילים כיצד תתמודדו עם הגבלת הזמן במקום עומק. מהו השם הכללי לאלגוריתמים אשר יכולים לשפר את ביצועיהם ככל שיש להם יותר זמן? ציינו אלגוריתם נוסף שכזה אשר עליו למדנו בקורס.
  4. (רטוב: 10 נק') עליכם לממש את המחלקה AgentMinimax בקובץ submission.py. שימו לב! הסוכן מוגבל משאבים, כאשר המשתנה time\_limit מגביל את מספר השניות שהסוכן יכול לרוץ לפני שיחזיר תשובה. (הגבלת הזמן עלייה אתם נבדקים הינה שנייה כלומר 1-t).
- \*אתם רשאים להיעזר בספריה func\_timeout אך אין זו דרישה
5. (יבש: 3 נק') נניח שבסביבה היו K שחקנים במקום 2. אילו שינויים יהיה צריך לעשות במימוש סוכן Minimax? כתבו פסאודו קוד בדומה לזה שראינו בתרגול.
    - a. בהינתן שכל סוכן רוצה לנצח ולא אכפת לו מתועלת של סוכנים אחרים.
    - b. בהנחה והדבר היחיד שכל סוכן רוצה הוא לפגוע בתועלת שלכם.
    - c. בהנחה שכל סוכן רוצה שהסוכן שאחריו יקבל תועלת מקסימלית.



### חלק ג' - Alpha-Beta (13 נק')

1. (רטוב: 10 נק') ממשו שחקן אלפא - בטא מוגבל משאבים במחלקה AgentAlphaBeta בקובץ submission.py, כך שיתבצע גיזום כפי שנלמד בהרצאות ובתרגולים.
2. (יבש: 3 נק') האם הסוכן שמימשתם בחלק זה יתנהג שונה מהסוכן שמימשתם בחלק ב' **מבחינת זמן ריצה ובחירת מהלכים**? הסבירו.

## חלק ד' - Expectimax (15 נק')

1. (יבש: 5 נק') עבור משחקים הסתברותיים כמו שש־בש, בהם יש מגבלת משאבים, משתמשים באלגוריתם RB-Expectimax. הניחו כי ידוע שהפונקציה היוריסטית  $h$  המשומשת באלגוריתם זה מקיימת  $-1 \leq h(s) \leq 1$   $\forall s$ . איך ניתן לבצע גיזום לאלגוריתם זה? תארו בצורה מפורטת את תנאי הגזימה, והסבירו את הרעיון מאחוריו.
2. (רטוב: 10 נק') הסוכן של minimax ו-alpha-beta מניח שהסוכן משחק את המהלך הגרוע ביותר עבורו אך דבר זה לא תמיד נכון. אפשר להתחשב באפשרות שהיריב יבחר בפעולה שאיננה הפעולה המזיקה ביותר (עבורנו) באמצעות סוכן Expectimax.  
גילינו מידע סודי על הרובוט המתחרה, הוא בוחר בין כל הפעולות בצורה יוניפורמית (בצורה אחידה) למעט תזוזה ימינה ולאסיפת חבילה (כאשר פעולות אלו אפשריות) אשר להן הוא מקצה הסתברות גדולה פי 2 משאר הפעולות.  
ממשו אלגוריתם Expectimax המשתמש במידע הסודי שקיבלתם.

## חלק ה' - משחק עם פקטור סיעוף גדול (8 נק')

1. (יבש: 4 נק') להלן שינויים אפשריים ששוקלים "בינה מחסנים" לעשות במשחק בכדי לבחון יכולות נוספות של הרובוטים. עבור כל שינוי ציינו מה ההשפעה שלו על מקדם הסיעוף וחשבו את מקדם הסיעוף החדש המתקבל.

a. הגדלת לוח המשחק להיות 8X8 והוספת מחסומים בסביבה. (מחסומים משמע משבצות שהסוכן לא יכול לעבור בהן).

b. הוספת היכולת של רובוט בכל תור לבחור משבצת על הלוח ולהניח עלייה בלוק, משמע בכל תור יכול הרובוט לנוע למעלה, למטה, ימינה, שמאלה, לאסוף חבילה, להוריד חבילה, להטעין, ולהניח בלוק על הלוח, בלוק יכול להיות מונח על כל משבצת ריקה. (כולל ריקה מחבילות, עמדות טעינה, רובוטים אחרים ובלוקים אחרים).

## 2. (יבש: 4 נק') בהנחה ומימשו את השינוי השני עבור הסביבה (סעיף b1)

- a. האם יש אלגוריתם מהסעיפים הקודמים שנוכל להשתמש בו שזמן הריצה שלו סביר? (סביר משמע לא גדול מהותית מהזמן שלוקח לו להחזיר צעד עבור המשחק בלי השינוי).
- b. הציעו אלגוריתם שונה מאלו שמימשתם בסעיפים הקודמים, שנלמד בקורס, שירוך בזמן סביר. הסבירו מדוע בחרתם בו ולמה הוא טוב להתמודדות עם האתגר שנוצר משינוי הסביבה.

## חלק ו' - יבש - שאלה פתוחה - MCTS (12 נק')

בשאלה זו נתמקד בבעיית ה-Multi-Armed Bandit עם שתי זרועות שנלמדה בכיתה.

**תזכורת:** בבעיה זו אתן עומדות בפני שתי מכונות מזל.

בכל תור אתן יכולות למשוך באחת משתי הזרועות  $a_1, a_2$  ולקבל תגמול  $r$  המקיים  $r \in [0, 1]$ .  
ההתפלגות של התגמול עבור כל אחת לא ידועה ומטרתכן היא למקסם את התגמול המצטבר.

נזכיר כי עבור משיכה בזרוע  $a_i$ :

$$UCB_1(a_i) = \frac{U(a_i)}{N(a_i)} + C \cdot \sqrt{\frac{\ln(N_{all})}{N(a_i)}}$$

כאשר:

- $U(a_i)$  – סכום התגמולים שהתקבלו עד כה ממשיכת  $a_i$ .
- $N(a_i)$  – מספר הפעמים שנמשכה הזרוע  $a_i$ .
- $C \geq 0$  – קבוע ממשי אי שלילי.
- $N_{all}$  – מספר הפעמים שנמשכו כלל הידיות.

1. (יבש: 4 נק') עבור סעיף זה בלבד, זרוע  $a_1$  נמשכה 5 פעמים וקיבלה תגמולים בסך 3.5 (במצטבר) וזרוע  $a_2$  נמשכה 2 פעמים וקיבלה תגמולים בסך 1.4 (במצטבר). תחת ערך  $C = 1$ , בצעו את השלבים הבאים:

א. חשבו את ערכי  $UCB(a_1), UCB(a_2)$ .

ב. קבעו איזה זרוע האלגוריתם יבחר למשוך בסיבוב הבא.

2. (יבש: 4 נק') עבור סעיף זה בלבד, בשלב האתחול בוצעו 8 משיכות ב- $a_1$  ו-2 משיכות ב- $a_2$ . ממוצע התגמולים שהתקבל ממשיכת  $a_1$  היה 1 בעוד שממוצע התגמולים ממשיכת  $a_2$  היה 0.5. מצאי חסם תחתון ל- $C$  כך שהפעולה הבאה שתיבחר לפי כלל החלטה תהיה  $a_2$ . פרטי חישוביך.

3. (יבש: 4 נק') עבור סעיף זה בלבד, משיכה ב- $a_1$  מניבה תמיד תגמול של 1 בעוד שמשיכה ב- $a_2$  מניבה תמיד תגמול של 0. בשלב האתחול כל זרוע נמשכה פעם אחת בלבד. כעת נמשיך למשוך בידיות לנצח על פי כלל החלטה. האם בהכרח כלל החלטה יבחר מתישהו ב- $a_2$ ? תחילה סמני את ההיגד הנכון ביותר:

1. כן, עבור כל ערך  $C \geq 0$  מתקיים כי מתישהו  $a_2$  ייבחר.
2. כן, עבור כל ערך  $C > 0$  מתקיים כי מתישהו  $a_2$  ייבחר, אולם לא עבור  $C = 0$ .
3. לכל ערך  $C \geq 0$  מתקיים כי  $a_2$  לעולם לא ייבחר.
4. קיים ערך  $C > 0$  עבורו  $a_2$  לעולם לא ייבחר אך גם קיים ערך שכזה עבורו  $a_2$  מתישהו כן ייבחר.

כעת נמקי בפירוט את התשובה אותה סימנת, לא יינתן ניקוד לשאלה ללא נימוק מתאים.

## שאלה פתוחה – אלגוריתם מונטה קרלו למשחקי אינפורמציה חלקית (12 נק')

שאלה זו עוסקת באלגוריתם "מונטה קרלו למשחקי אינפורמציה חלקית" המופיע בסוף תרגול 6, ולא באלגוריתם MCTS המופיע בהרצאה.

לאורך השאלה הניחו כי כל אחת מההשלמות החוקיות למצב החלקי סבירה באותה מידה, קרי המצב האמיתי מתפלג יוניפורמית מעל  $S_{complete}$ .

1. (2 נק') הסבירו בקצרה מה הבעיה שאלגוריתם מונטה קרלו מנסה לפתור וכיצד הוא מתמודד עם בעיה זו.

2. (3 נק') האלגוריתם מקבל כפרמטר את  $K$ , מספר המצבים הנדגמים מעל  $S_{complete}$ . הסבירו את השיקולים בבחירת ערך הפרמטר. מה יקרה אם נבחר קטן מדי או גדול מדי? התייחסו בתשובתכם למשאבי החישוב ולטיב מידול העולם.

3. (4 נק') דנה טוענת כי עבור ערך  $K$  גדול השווה ל-  $|S_{complete}|$  ניתן למדל את אלגוריתם מונטה קרלו באמצעות צמתים הסתברותיים ואלגוריתם Expectimax. בהינתן משחק אינפורמציה חלקית וערך  $K$  שכזה הציעו מידול המשתמש בצמתי Expectimax אשר יהיה שקול להרצת מונטה קרלו. הסבירו את שקילות האלגוריתמים. בתשובתכם ניתן להזניח גיזומים.

4. (3 נק') בהתבסס על תשובתכם מסעיף 2, הציעו גרסת Anytime לאלגוריתם וכתבו פסאודו-קוד לגרסה זו. הסבירו כיצד הפתרון שלכם מתיישב עם עקרון אלגוריתמי ה-Anytime כפי שנלמדו בכיתה.