

תרגיל בית 2 – חיפוש רב סוכני

2048

מטרת התרגיל

בתרגיל זה נממש סוכני חיפוש בסביבה מרובת סוכנים, נממש את הסוכן על פי אלגוריתם Minimax ושיפוריו עבור המשחק 2048, נתנסה במימוש היוריסטיקה עבור המשחק, ביצוע ניסויים וכתובת דו"ח.

הנחיות כלליות

- תאריך הגשה: יום רביעי, 09.6.2021, עד השעה 23:59.
- יש להגיש את המטלה בזוגות בלבד.
- יש להקליד דוחות מוקלדים בלבד.
- המתרגלים האחראיים על התרגיל: יקיר יהודה וסער הוברמן.
- ניתן לשלוח שאלות בנוגע לתרגיל לתיבת המייל הקורסית: ai.technion@gmail.com תחת הכותרת "שאלות על תרגיל בית 2". 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.
- ניתן לשאול שאלות על התרגיל בסדנה ביום רביעי בשעה 15:30.
- בקשות דחייה **מוצדקות** יש לשלוח למתרגל האחראי של הקורס **בלבד** – טל סויסה, במייל talswisa@cs.technion.ac.il.
- במהלך התרגיל ייתכן שנעלה עדכונים, תיקונים והבהרות לדף FAQ ייעודי באתר ולמסמך הנ"ל. העדכונים הינם **מחייבים**, ועליכם להתעדכן עד מועד הגשת התרגיל.
- בסיום העבודה תגישו חלק יבש הכולל מענה לשאלות ודיווח תוצאות, עליכם לכתוב תשובה בדו"ח היבש בכל סעיף שלצדו תראו את הסימן ✍️.
- העתקות תטופלנה בחומרה.

תיאור המשחק

המשחק 2048 הוא משחק של שני שחקנים, אשר יוגדרו בעבודה זו כשחקן ה-move, ושחקן ה-index. הלוח בנוי כך שבכל תור שחקן ה-index בוחר לשים ריבוע בעל אחד מהערכים {2,4} על אחת המשבצות הריקות בלוח ושחקן ה-move בוחר להזיז את כל הריבועים הקיימים בלוח לאחד מארבעת הכיוונים: ימינה, שמאלה, למעלה ולמטה. כאשר שני ריבועים בעלי ערך x זהה מתנגשים הם נעלמים ובמקומם נוצר ריבוע חדש בעל הערך 2x. המשחק נגמר כאשר שחקן ה-move לא יכול לבצע את הצעד הבא.

מטרת שחקן ה-move לקבל את הציון הגבוה ביותר, כאשר הציון מתקבל על ידי סכום ערכי הריבועים אשר יוצר שחקן ה-move בלבד, ריבועים אלו נוצרים כתוצאה מהתנגשות של זוגות ריבועים בעלי אותו

ערך. לדוגמא התנגשות של שני ריבועים בעלי הערך 4, תיצור ריבוע חדש בעל ערך 8 במקומם ותוסיף 8 נקודות לציון השחקן.

מטרתו של שחקן ה-index הוא ששחקן ה-move יקבל ערך נמוך ככל הניתן. מומלץ להתנסות במשחק בכתובת <https://play2048.co> ולקרוא בהרחבה ב- [https://en.wikipedia.org/wiki/2048_\(video_game\)](https://en.wikipedia.org/wiki/2048_(video_game)).

(*) בשונה מהמשחק המקורי, עבור כל השחקנים מלבד Expectimax, ערך הריבוע ששחקן האינדקס יניח יהיה 2 בלבד.

הרצת משחקים

לפני תחילת מימוש הסוכנים, מומלץ להתנסות במשחק. הריצו את השורות הבאות ב-terminal בתיקיה בה נמצא הקוד המצורף לתרגיל:

```
python main.py -game KeyboardGame
```

במקרה זה תוכלו לשחק את שחקן ה-move בעזרת המקשים (left), w (up), s (down), d (right) או מקשי החצים כתלות במערכת ההפעלה. כאשר שחקן ה-index הוא רנדומי וממומש במחלקה RandomIndexPlayer בקובץ submission.py.

הרצת המשחק בצורה שאינה ידנית, מבוצעת ללא הדגל -game, ובחירת הסוכנים מבוצעת על ידי player1 - שמגדיר את שחקן ה-move ו player2 - שמגדיר את שחקן ה-index. באופן הבא:

```
python main.py -player1 MiniMaxMovePlayer -player2 MiniMaxIndexPlayer
```

כאשר שמות השחקנים האפשריים מוגדרים בקובץ main.py:

```
move_players = {'GreedyMovePlayer': submission.GreedyMovePlayer,
                 'ImprovedGreedyMovePlayer': submission.ImprovedGreedyMovePlayer,
                 'MiniMaxMovePlayer': submission.MiniMaxMovePlayer,
                 'ABMovePlayer': submission.ABMovePlayer,
                 'ExpectimaxMovePlayer': submission.ExpectimaxMovePlayer,
                 'ContestMovePlayer': submission.ContestMovePlayer
                }

index_players = {'RandomIndexPlayer': submission.RandomIndexPlayer,
                 'MiniMaxIndexPlayer': submission.MiniMaxIndexPlayer,
                 'ExpectimaxIndexPlayer': submission.ExpectimaxIndexPlayer
                }
```

ניתן להגדיר את הגבלת הזמן בשניות לתור (יוסבר בהמשך) על ידי הדגל -move_time, באופן הבא:

```
python main.py -player1 MiniMaxMovePlayer -player2 MiniMaxIndexPlayer
-move_time 5
```

המשתנה move_time מגדיר את מגבלת הזמן לתור בשניות עבור כל אחד מהשחקנים. כאשר שחקני ה-move ושחקני ה-index יקבלו משתנה זה בשם time_limit בכניסה לפונקציות get_move, get_indices בהתאמה. שחקן אשר לא יחזור מהפונקציה לאחר זמן זה, יפסיד אוטומטית במשחק. (וייכשל באותו טוט)

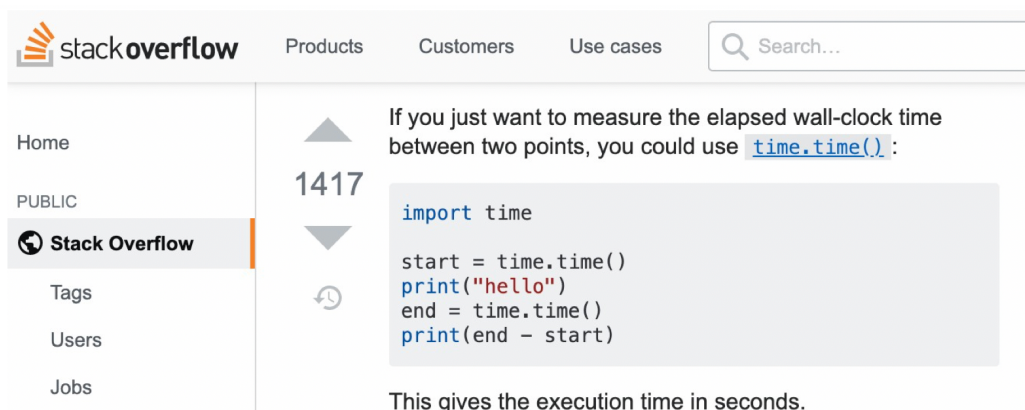
(*) ניתן להניח $move_time \geq 1$.

המלצות למימוש:

- עבור שחקני ה-move מומלץ להשתמש בפקודות הממומשות במילון commands אשר נמצא בחלק העליון של קובץ submission.py:

```
commands = {Move.UP: logic.up, Move.DOWN: logic.down,  
            Move.LEFT: logic.left, Move.RIGHT: logic.right}
```

- עבור שחקני ה-index יש לשים לב שהשמת ערך חדש במשבצת בלוח יבוצע רק למשבצת ריקה (בעלת הערך 0).
- כדי להתמודד עם מגבלת הזמן, ניתן להשתמש בספריה time:



חלק א' - מימוש סוכן משופר:

בקובץ submission.py שמצורף לתרגיל ישנו מימוש של שחקן GreedyMovePlayer, שחקן זה יבחר את הפעולה שתמקסם את ערך ה-score שלו בהסתכלות על התור הקרוב בלבד. בחלק זה תממשו סוכן מתוחכם יותר עבור השחקן move, במחלקה ImprovedGreedyMovePlayer.

1. (3 נק') הגדירו באופן מפורש יוריסטיקה משלכם להערכת מצבי המשחק. בפרט, יש לספק נוסחה עבור כל פרמטר של המשחק אשר נלקח בחשבון לשם חישוב היוריסטיקה. היוריסטיקה צריכה להכיל לפחות 2 פרמטרים חדשים בנוסף לשחקן ה-greedy. 📝

2. (3 נק') הסבירו את המוטיבציה להגדרה זו (ולכל פרמטר המופיע בחישובים שלכם). האם אתם צופים שהיא תשפר את ביצועי השחקן ביחס ליוריסטיקה של greedy_player, הסבירו למה. 📝

3. (6 נק') ממשו שחקן במחלקה ImprovedGreedyMovePlayer בקובץ submission.py הנעזר ביוריסטיקה שהגדרתם, ללא שינויים נוספים.

4. (3 נק') בצעו 3 הרצות עבור כל אחד מהמשחקים הבאים:
א. GreedyMovePlayer נגד RandomIndexPlayer.
ב. ImprovedGreedyMovePlayer נגד RandomIndexPlayer.

צרפו והסבירו את התוצאות שקיבלתם. 📝

חלק ב' - מימוש סוכני Minimax

בחלק זה נתמקד באלגוריתם Minimax, ונממש סוכני Minimax מוגבלי משאבים.

5. (4 נק') האם האסטרטגיה של סוכן MiniMaxMovePlayer מתאימה עבור משחק עם שחקן RandomIndexPlayer? הסבירו למה, ואם לא הציעו אסטרטגיה אחרת ונמקו. (יש לכתוב תשובה קצרה, לא יותר מארבע שורות) 📝
6. (6 נק') במשחק בין השחקנים MiniMaxMovePlayer, MiniMaxIndexPlayer, בהנחה ששני השחקנים אינם מוגבלי משאבים:
א. (3 נק') מה אפשר להגיד על ערך התוצאה הסופית? האם השחקן יכול לדעת אותו בתחילת המשחק, מדוע? 📝
ב. (3 נק') במשחק בו אחד השחקנים מוגבל משאבים, האם ערך התוצאה הסופית תישאר זהה לסעיף הקודם **בהכרח**? אם כן ציינו איזה שחקן, אם לא הסבירו מדוע. 📝
7. (6 נק') סטודנט מימש שחקן Minimax עבור משחק שחמט. כאשר שיחק מול יריב ראה שישנם מצבים בו השחקן יכול לנצח בתור הבא במט (מצב סופי), אך הוא אינו מבצע מהלך זה ומבצע במקום מהלך אחר.
א. (3 נק') איך ייתכן מצב כזה? הסבירו בצורה מפורטת. 📝
ב. (3 נק') הציעו שינוי ל-Minimax שימנע מצבים כאלה. 📝
8. (3 נק') דרך נפוצה להתמודדות עם הגבלת זמן בהרצת מינימקס היא שימוש ב-iterative deepening. מנו יתרון אחד וחיסרון אחד בשיטה זו? הציעו שיפור עבודה. 📝
9. (16 נק') עליכם לממש שני שחקני מינימקס, במחלקות MiniMaxMovePlayer ו-MiniMaxIndexPlayer בקובץ submission.py.
אלו שחקנים **מוגבלי משאבים** (ניתן להשתמש בפונקציה היוריסטית של חלק א או לכתוב פונקציה יוריסטית חדשה), כאשר יוגדר כ-input לפונקציות get_move, get_indices המשתנה time_limit אשר מגדיר את מגבלת הזמן בשניות.

(*) בשביל התמודדות עם מגבלת הזמן ניתן להשתמש ב-iterative deepening, או כל גישה אחרת.

(**) הדגש בתרגיל זה הוא על מימוש נכון של אלגוריתם Minimax אשר מתמודד עם מגבלת הזמן, לכן מימוש נכון אשר משתמש בפונקציות יוריסטיות פשוטות יקבל את מלוא הנקודות.

(***) ניתן לבחון את נכונות המימוש על ידי הרצת מספר משחקים בין השחקנים

MiniMaxIndexPlayer, MiniMaxMovePlayer.


חלק ג' – מימוש סוכני $\alpha - \beta$

בחלק זה נתמקד באלגוריתם $\alpha - \beta$, ונממש סוכן $\alpha - \beta$ מוגבל משאבים.

10. (8 נק') סטודנט בקורס היה מעוניין לנצח בטורניר. בשביל ששחקן האלפא -בטא שלו ירוץ מהר יותר, החליט כי עליו לבצע גיזום יותר משמעותי זאת תוך התפשרות על משפט ההבטחה של האלגוריתם, כך שערך המתקבל יהיה לכל היותר שונה באפסילון מהערך המקורי. האלגוריתם מקיים את התנאי הבא לכל מצב במשחק:

$$Alg(state) \leq MiniMax(state) + \epsilon$$

א. (4 נק') שנו שורה אחת בלבד מהקוד של אלפא-ביתא כדי לקבל את ההתנהגות המבוקשת.

ב. (4 נק') תנו דוגמה למקרה בו האלגוריתם מחזיר ערך אחר מערך המינימקס. 


11. (8 נק') ממשו שחקן $\alpha - \beta$ מוגבל משאבים במחלקה ABMovePlayer בקובץ submission.py.

(*) הדגש בתרגיל זה הוא על מימוש נכון של אלגוריתם $\alpha - \beta$ אשר מתמודד עם מגבלת הזמן, לכן מימוש נכון אשר משתמש בפונקציות יוריסטיות פשוטות יקבל את מלוא הנקודות.

12. (8 נק') עבור זמני התורות 1,2,4 בשניות ויוריסטיקה זהה בצעו משחק מול MiniMaxIndexPlayer, פעם אחת עם שחקן MiniMaxMovePlayer ופעם אחת עם שחקן ABMovePlayer. והציגו:

א. (גרף עומק ממוצע כפונקציה של time_limit. (אפשר כגרף יחיד או גרף עבור כל אחד משחקני ה-move בנפרד).

ב. גרפי תוצאת משחק כפונקציה של time_limit. (אפשר כגרף יחיד או גרף עבור כל אחד משחקני ה-move בנפרד).

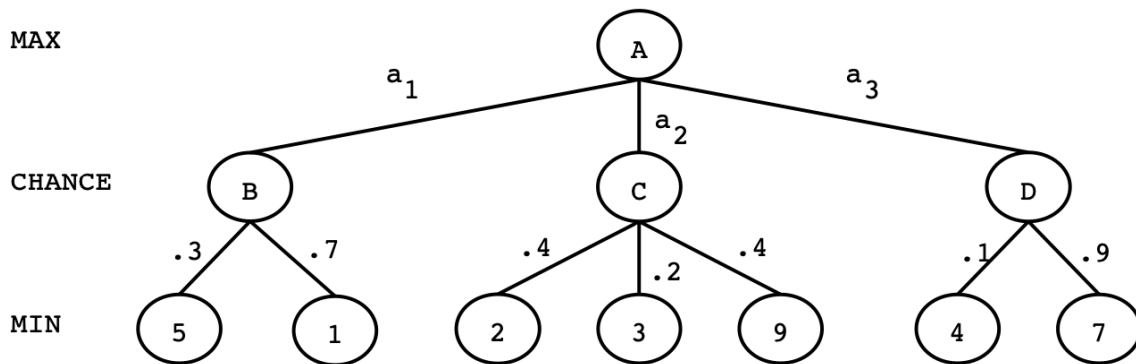
הסבירו את התוצאות. 

חלק ד' – מימוש סוכני Expectimax

בחלק זה נתמקד באלגוריתם Expectimax, ונממש סוכני Expectimax מוגבלי משאבים.

13. (7 נק') בגיזום $\alpha - \beta$ גוזמים בצמתי min ובצמתי max. האם באלגוריתם Expectimax ניתן לגזום באותו אופן **בדיוק**? אם כן הסבר במילים למה, אחרת תן דוגמה נגדית 🖋️

14. (5 נק') עבור עץ ה-Expectimax הבא:



א. (2 נק') חשב את ערכי ה-Expectimax בצמתיים B, C, D. (כאשר רשום x. הכוונה להסתברות 0.x) 🖋️

ב. (1 נק') ערכה של איזה צומת יועבר לצומת A? 🖋️

- i. צומת B
- ii. צומת C
- iii. צומת D

ג. (2 נק') אם ערכי ה-Utility בעלים יוכפלו בקבוע c, האם תשתנה תשובת הסעיף הקודם? 🖋️

15. (14 נק') עליכם לממש שני שחקני Expectimax **מוגבלי משאבים**, במחלקות ExpectimaxMovePlayer ו-ExpectimaxIndexPlayer בקובץ submission.py. כאשר ישנה צומת רנדומית אשר מגדירה את ערך המשבצת בהסתברויות הבאות:
 $P(2) = 0.9, P(4) = 0.1$. השחקן ExpectimaxIndexPlayer יקבל את ערך המשבצת כפרמטר value לפונקציה get_indices.

(*) הדגש בתרגיל זה הוא על מימוש נכון של אלגוריתם Expectimax אשר מתמודד עם מגבלת הזמן, לכן מימוש נכון אשר משתמש בפונקציות יוריסטיות פשוטות יקבל את מלוא הנקודות.
 (**) ניתן לבחון את נכונות המימוש על ידי הרצת מספר משחקים בין השחקנים ExpectimaxMovePlayer, ExpectimaxIndexPlayer.

חלק ה' – תחרות

כל זוג נדרש לממש שחקן יחיד מסוג `move` שייצג אותו בתחרות בה ישתתפו כל הזוגות. הסטודנטים ששחקניהם יצטיינו בתחרות יזכו בבונוס לציון הסופי של הקורס: 5 נקודות לזוג המנצח, 3 נקודות למקום השני, ושתי נקודות למקום השלישי. את שחקן התחרות שלכם יש לממש בקובץ `submission.py`, במחלקה בשם `ContestMovePlayer`. אתם יכולים להגיש את השחקן שמימשתם בחלק ג' או לממש שחקן חדש, עבור שחקן זה גם כן תהיה מגבלת זמן לתור כמו בחלקים קודמים. ניתן לבחון את ביצועי השחקן מול היריבים שמימשתם או שסופק לכם. במידה והשחקן שתגישו לא ירוץ, יורדו לכם נקודות בתרגיל. ההשתתפות בתחרות היא חובה.

חוקים:

- אסור לשחקן להשתמש ברשת האינטרנט או ברשת מקומית כלשהי.
- אסור לשחקן להשתמש בכל סוג של קוד מקבילי.
- כל שחקן שינסה לרמות או לשבש את קוד היריב או המערכת ייפסל, ויורדו למגישים נקודות בתרגיל.
- חל איסור להשתמש במידע שעובד מראש ונשמר בקובץ.
- אסור להשתמש בחבילות/ספריות שאינן מובנות בפייתון, מלבד `networkx`, `collections` והחבילות אשר מבוצע בהם `import` בקוד שסופק לכם.
- כל חישוב לשם קביעת הצעד הבא של הסוכן צריך להתבצע אך ורק בקריאה לפונקציה `.get_move`.

הוראות הגשה

יש להגיש את כל הקבצים (קוד ודו"ח) בקובץ `zip` יחיד ששמו `<id2>_<id1>_AI2` כאשר במקום `<id1>` ו-`<id2>` יש לרשום את מספרי תעודת הזהות של המגישים. מלבד לדו"ח, קובץ ה-`zip` צריך לכלול קובץ קוד יחיד בשם `submission.py`.

בהצלחה!