

# תרגיל בית 2 – חיפוש רב סוכני



פניות לסגל בנוגע לתרגיל בית יהיו דרך הפיאצה בלבד (לא יהיה מענה דרך המייל)  
קישור לפיאצה: [piazza.com/technion.ac.il/spring2022/236501](https://piazza.com/technion.ac.il/spring2022/236501)

לפני ששואלים שאלה יש לבדוק אם היא נשאלה בעבר, שאלות שנשאלו בעבר לא יקבלו מענה  
מתרגל אחראי על התרגיל: דביר

## הנחיות כלליות

- תאריך הגשה: 02/06/22
- את המטלה יש להגיש **בזוגות בלבד** – בקשות להגשה ביחידים אשר מגיעות אל הסגל משודכות בצורה שרירותית לזוגות. עדיף למצוא בן זוג בעצמכם ולא לבנות על גישור מהסגל.
- יש להגיש את המטלה **מוקלדת בלבד** – פתרון בכתב יד לא ייבדק.
- התשובות צריכות להיות כתובות בשפה העברית, או באנגלית באישור המתרגל האחראי על הקורס (ספיר טובול).
- אפשר לשלוח שאלות בנוגע לתרגיל דרך **פיאצה בלבד**.
- המתרגל האחראי על התרגיל: דביר דוד ביטון.
- בקשות דחיה מוצדקות יש לשלוח למתרגל האחראי בלבד.
- 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 תטופלנה בחומרה
- ציון המטלה כולל חלק יבש וחלק רטוב. בחלק היבש נבדוק שתשובתכם נכונה, מלאה, קריאה ומסודרת. בחלק הרטוב הקוד שלכם ייבדק בצורה מקיפה באמצעות בדיקות אוטומטיות – אשר ישוו את המימוש שלכם למימוש שלנו. חשוב לעקוב בזהירות אחר הוראות התרגיל מאחר שסטיות מהמימוש המבוקש עלולות להוביל לכשל בטסט האוטומטי (גם אם המימוש "נכון ברובו"). במהלך הבדיקה האוטומטית יינתן זמן סביר לכל הרצה – כך שכל עוד תעקבו אחר ההוראות, המימוש שלכם יעמוד בהגבלות הזמנים ויחזיר תוצאות טובות מספיק בשביל הטסט.
- 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 ("איך אפשר לדעת אם מונית מלאה" וכדומה).
- מומלץ לא לדחות את התרגיל לרגע האחרון מאחר שהמימוש וכתובת הדו"ח עלולים לקחת יותר זמן מהצפוי.
- לשון זכר, נקבה או רבים מתייחסים כלפי כלל המינים.

## תיאור המשחק

בדומה לתרגיל הקודם, גם הפעם נתעסק בסביבה של מוניות – אולם הפעם, בטעם מעט שונה. הסביבה אינה האחת של gym שכבר ראיתם, למרות שקיים דמיון. המימוש שלה נמצא בקובץ TaxiEnv.py בקוד המצורף.

סביבת MultiTaxi כוללת לוח בגודל 4 על 4 כאשר על הלוח 2 מוניות, 2 תחנות דלק ו-2 נוסעים עם 2 יעדים אליהם הם מעוניינים להגיע, ומספר הצעדים שנותרו ליום.

המוניות מתחילות ללא כסף ועם דלק, כאשר הן ריקות. כל תזוזה (צפונה, דרומה, מזרחה או מערבה) עולה למונית יחידה אחת של דלק. מונית יכולה לאסוף לקוח (אם היא ריקה) ולהוריד לקוח (אם היא מלאה).

כאשר מונית מורידה לקוח ביעדו היא מקבלת N יחידות כסף, כאשר N הוא מרחק מנהטן בין מיקום הלקוח המקורי ויעדו של הלקוח כפול 2. לאחר שהלקוח יורד ביעדו הוא נעלם ולקוח אחר מופיע במפה במקומו (לא בהכרח באותו המיקום או עם אותו היעד).

אם מונית נמצאת בתחנת דלק היא יכולה לתדלק על ידי המרת כל יחידות הכסף שיש לה ליחידות דלק, כאשר כל יחידת כסף מומרת ביחידת דלק אחת.

המונית המנצחת היא האחת עם מספר יחידות הכסף הגדול ביותר כאשר לא נותרו עוד צעדים ליום. מומלץ לעבור על קבצי הקוד לפני תחילת העבודה.

## הרצת המשחק

לפני תחילת מימוש הסוכנים, מומלץ להתנסות עם המשחק. הריצו את המשחק דרך טרמינל הפתוח בתיקיה עם הקוד המצורף לתרגיל באמצעות שורת הקוד:

```
python main.py greedy random -t 0.5 -s 1234 -c 1000 --print_game
```

שני הארגומנטים הראשונים הם שמות הסוכנים שירוצו. הארגומנט t הוא הגבלה לזמן של תור. s הוא seed למשחק (שולט באקראיות המשחק – איפה מתחילות המוניות וכדומה), c הוא הגבלה על מספר התורות אשר כל סוכן מבצע. print\_game מדפיס את המשחק.

בלי הארגומנט print\_game רק המנצח במשחק והכסף בסופו מודפסים. בלי t, ו-c נבחרים ערכים קבועים מראש עבור s נבחר ערך אקראי.

המשחק יריץ סוכן חמדן מול סוכן אקראי (בוחר אחד מהאופרטורים החוקיים שרירותית) אחד נגד השני.

נסתכל על אחד מהשלבים אשר מודפסים על ידי הפקודה:

```
taxi 0 chose move north
[ ][ ][ ]
[G0][ ][T0]
[ ][ ][P1][ ]
[P0][ ][T1][ ]
taxis: [position:(3, 1) fuel: 15 cash: 0 passenger: [None], position:(2, 3) fuel: 16 cash: 0 passenger: [None]]
passengers on street: [position:(0, 3) destination: (0, 1), position:(2, 2) destination: (3, 1)]
gas stations: [position:(0, 1), position:(2, 2)]
```

מודפס מספר הסוכן (יכול להיות 0 או 1) והאופרטור בו בחר. לאחר מכן מודפס הלוח לאחר שהסוכן הפעיל את האופרטור. הלוח כולל 16 משבצות כאשר בכל אחת יכולים להיות מונית, לקוח המחכה למונית או תחנת דלק.

האותיות אשר מופיעות בלוח מסמלות:

T – Taxi

G – Gas station

P – Passenger on street

D – Destination of Passenger on street

X – Destination of Passenger in taxi

המספר המופיע ליד כל אחת מהאותיות הוא המזהה של האובייקט אליו הוא משויך – T1 עבור המונית הראשונה וכדומה. עבור נוסעים במונית, המספר ליד X הוא המזהה של המונית בה הוא נמצא. עבור לקוחות ברחוב, המספר ליד D הוא המזהה של הלקוח ברחוב.

לעיתים מספר אובייקטים מופיעים באותו המיקום בתמונה ואז מודפס רק אחד מהם. במקרה זה, אפשר להשתמש ברשימות המפורשות שמופיעות אחרי הלוח.

## חלק א' – סוכן חמדן משופר (15 נקודות)

בקובץ `submission.py` שמצורף לתרגיל תמצאו מימוש של סוכן חמדן – `AgentGreedy`. הוא בוחר את המצב שממקסם את ההיוריסטיקה שלו. ההיוריסטיקה שעליה מסתמך הסוכן לא טובה במיוחד – הוא מחזיר את הפרש הכסף בינו לבין יריבו.

1. (יבש: 3 נק') הגדירו היוריסטיקה משלכם להערכת מצבי המשחק. על ההיוריסטיקה לכלול שלושה פרמטרים חדשים מלבד כסף. יש לכתוב נוסחה מפורשת עבור ההיוריסטיקה, בהינתן מיקומי האובייקטים בלוח והמאפיינים שלהם (דלק במונית, יעד הנוסע וכדומה). בחרו שמות ברורים בנוסחה שלכם.

2. (יבש: 4 נק') הסבירו את המוטיבציה לשינויים שביצעתם בהיוריסטיקה. האם לפי דעתכם סוכן חמדן המבוסס על ההיוריסטיקה שלכם ינצח את הסוכן החמדן הנתון? פרטו למה.

3. (רטוב: 8 נק') ממשו את `AgentGreedyImproved` במחלקה `submission.py` בקובץ `submission.py` את ההיוריסטיקה שלכם.

מומלץ להסתכל בקובץ `Agent.py` על מנת להבין איך יש לממש את המחלקה. מומלץ לקרוא גם את הקוד בקובץ `TaxiEnv.py`, מאחר שתצטרכו לגשת לשדות ולמתודות של המחלקה.

צרפו את התוצאות הסופיות עבור 3 הרצות של כל אחד מהמשחקים הבאים:

א. חמדן נגד החמדן המשופר

ב. חמדן נגד אקראי

ג. חמדן משופר נגד אקראי

## חלק ב' – סוכן Minimax (20 נקודות)

1. (יבש: 5 נק') חבר בקורס מימש סוכן Minimax בתרגיל בית זה (ספוילר) ומבקש את עזרתכם באיתור באגים בקוד שלו. שמתם לב שלעיתים הסוכן יכול לנצח בתור באמצעות צעד מסוים, אולם הוא בוחר בצעד אחר. האם יש באג בקוד של חברכם? הסבירו בפירוט מה הוביל להתנהגות של הסוכן, וכמו כן הציעו שינוי שימנע את ההתנהגות הזו.

2. (יבש: 5 נק') דרך להתמודד עם הגבלת זמן ריצה על מינימקס היא באמצעות `iterative deepening`. השימוש ב-`iterative deepening` עם Minimax דומה לאחד אשר ראיתם עם DFS. מדובר בהרצת האלגוריתם כמה פעמים כאשר בכל חזרה הגבלת העומק גדלה – עד אשר נגמר הזמן שהוקצב לריצה. התוצאה מהאיטרציה האחרונה של האלגוריתם (כאשר החסם על העומק היה מקסימלי) היא האחת שמוחזרת.

שיטה אחרת היא באמצעות גיזום של ענפים (אלפא בטא). הציגו שני יתרונות עבור כל אחת מהשיטות. הציעו שינויים שאפשר לבצע לשיטות אשר יוכלו לעזור, באופן כללי או עבור סוכנים בסביבת `MultiTaxi`.

3. (רטוב: 10 נק') עליכם לממש את המחלקה `AgentMinimax` בקובץ `submission.py`. אתם יכולים לדרוס את ההיוריסטיקה. שימו לב שהסוכן מוגבל משאבים, כאשר המשתנה `time_limit` מגביל את מספר השניות שהסוכן יכול לרוץ לפני שיחזיר תשובה. אפשר להשתמש בכל שיטה על מנת להתמודד עם מגבלת הזמן (חוץ מגיזום).

## חלק ג' – סוכן Alpha-Beta (20 נקודות)

1. (רטוב: 10 נק') ממשו שחקן אלפא-בטא מוגבל משאבים במחלקה AgentAlphaBeta.
2. (יבש: 5 נק') נניח שבסביבה היו  $k$  מוניות במקום 2. אילו שינויים יהיה צריך לעשות במימוש של הסוכן? מה יקרה לזמן הריצה של האלגוריתם ככל שנגדיל את  $k$ ? פרטו.
3. (יבש: 5 נק') האם הסוכן שמימשתם בחלק ג' יתנהג שונה מהסוכן שמימשתם בחלק ב' מבחינת זמן ריצה ובחירת מהלכים? הסבירו.

## חלק ד' – סוכן Expectimax (15 נקודות)

1. (יבש: 5 נק') אותו החבר מחלק ב' גילה לכם שנית מבעוד מועד שהוא מימש סוכן Expectimax בתרגיל הבית. שנית, הוא חושש מבאגים במימוש שלו. שמתם לב שהוא השתמש באותו המימוש בדיוק כמו בסוכן האלפא-בטא שלו. בהנחה שהמימוש של החבר עבור סוכן אלפא-בטא נכון, האם מדובר בבאג? אם כן, הצג דוגמה בה נקבל תוצאות שונות ללא הגיזום. אם לא, הסבר מדוע.
  2. (רטוב: 10 נק') הסוכן של  $\alpha$ - $\beta$  ו-minimax מניח שהסוכן היריב יבחר באופרטור שיוביל לתוצאה האופטימלית בתורו, אולם זה לא תמיד מתרחש. לדוגמה, כאשר אנו מתחרים עם סוכן חמדן, סביר להניח שהוא לא יבחר בפעולה האופטימלית מדי פעם.  
אפשר להתחשב באפשרות שהיריב יבחר בפעולה שאינה אופטימלית בתורו באמצעות סוכן Expectimax.
- נבחר את המשקולות של כל אופרטור בצורה שתתחשב בכך שכאשר היריב יכול לאסוף נוסע, להוריד נוסע או לתדלק – ישנה הסתברות גדולה יותר שהוא יבצע פעולה זו מאשר שיזוז למקום אחר (מאחר שרק במקרים מסוימים ניתן לבצע פעולות "מיוחדות" אלו).
- ממש שחקן ExpectiMax מוגבל משאבים, כאשר ישנה צומת רנדומית אשר מגדירה את ההסתברות לאופרטורים של היריב בצעד נתון בצורה הבאה – ההסתברות של תזוזת זהה, וההסתברות של כל אחד משאר האופרטורים גדולה פי 2.
- כלומר, נניח שהיריב יכול לנוע לכיוון מזרח או צפון וכמו כן יכול לאסוף נוסע או לתדלק בתור מסוים. נסמן את ההסתברות לתזוזה עם  $x$  ואת ההסתברות לפעולה שהיא לא תזוזה עם  $y=2x$ . אזי נקבל:

$$x+x+2x+2x=1$$

$$x=1/6$$

$$y=1/3$$

## שאלה פתוחה (30 נקודות)

נהגת המונית שמחה גורה שמעה על תרגיל הבית שקיבלתם לאחר שהסיעה סטודנט מהקורס לתחנת הרכבת (הרכבל בדיוק התקלקל), והחליטה לממש אלגוריתם מינימקס לא מוגבל משאבים שבוחר בצעד המוביל לערך המינימקס המקסימלי (אפשר להניח שיש ערך מקסימלי יחודי).

א. שמחה מתחרה בנהגי מוניות והחליטה שפונקציית התועלת שלה תהיה "פונקציית סכום אפס". הביאו דוגמה לפונקציה אפשרית כזו.

שמחה התבלבלה ובטעות במקום להחזיר את ערך פונקציית התועלת היא מחזירה את הלוגריתם בבסיס 2 שלה!

לדוגמה, במקום להחזיר 4 מוחזר  $\log_2(4) = 2$ .

ב. האם בעקבות השינוי המהלכים שתבצע שמחה יהיו שונים? אם לא, הסבירו למה. אם כן, הביאו דוגמה שעבורה שמחה תבצע צעד שונה.

ג. נהג המונית מוטי לוכים גם ניסה לממש את אלגוריתם מינימקס, במקום ערך פונקציית התועלת, מחזירים את ערך פונקציית ערך התועלת בחזקת שבע. חזרו על סעיף א' עם האלגוריתם של מוטי.

ד. נהג מונית שלישי, סמי הרצל, ניסה לעזור לשמחה ומוטי והציע להם להשתמש באלגוריתם Expectimax במקום. חזרו על סעיף א' וסעיף ב' עם השינוי שהציע סמי.

ה. נהגת מונית רביעית, תמי תאן, מימשה פונקציית עזר  $f: S \rightarrow R$  המקיימת:

$$|MM(s) - f(s)| \leq 2$$

$$f(s) = MM(s) \text{ if } s \text{ is a leaf}$$

כאשר בהינתן  $s \in S$  מצב,  $MM(s)$  הוא ערך המינימקס לא מוגבל משאבים שלו.

בכל אחד מהמקרים הבאים, ציינו האם תמי יכולה להשתמש בפונקציה  $f$  על מנת להקטין את מספר הצמתים שיפותחו על ידי האלגוריתם בלי לפגוע בנכונות שלו. בכל סעיף, אם תשובתכם חיובית הוסיפו את הקוד המתאים כך שיתבצע מירב הגיזומים האפשריים.

1. אלגוריתם אלפא-בטא ללא הגבלת משאבים

2. אלגוריתם אלפא-בטא מוגבל משאבים (שימו לב שפונקציית העזר לא משתנות מהסעיף הקודם).