

תרגיל בית 3 – MDP

עbero על כל הנקודות לפני תחילת התרגיל.

הנקודות הכלליות:

- תאריך הגשתה: עד ליום האחרון של הסמסטר - 08/04/2024 ב-23:59 •
את המטלה יש להגיש **בצוגות בלבד**. •
יש להגיש מטלות מוקלחות בלבד, פתרונות בכתב יד לא יבדקו. •
ניתן לשלוח שאלות בנוגע לתרגיל בפייצה בלבד. •
המתרגל האחראי על תרגיל זה: **דניאל אלגריסי**. •
בקשות דחיה מוצדקות (מיילאים, אשפוז וכו') יש לשלוח למתרגל האחראי (**ספר טובל**) בלבד. •
במהלך התרגיל יתכן שנעלה עדכון, למסמך הנ"ל – תפורסם הודעה בהתאם. •
העדכנים הינם מחיבים, ועליכם להתעדכן עד מועד הגשת התרגיל. •
שים לב, העתקות תפולנה בחומרה. •
התשובות לסעיפים בהם מופיע הסימן  **ארכיכים להופיע בדוח.** •
לחلك הרטוב מספק בלבד של הקוד. •
אנחנו חשובים לפניות שלכם במהלך התרגיל ומעדכנים את המסמר זהה בהתאם. גרסאות עדכניות של
המסמך יועלו לאתר. **בבהרות** ועדכנים **শমস্পিস অধীন প্রস্তুত হারাণো যোমন কান** בצבעים. יתכן
שתפורסמנה גרסאות רבות – אל תיבהל מכך. השינויים בכל גרסה יכולים להיות קטנים.

שימוש לב שאתם משתמשים רק בספריות הפיתון המאושרות בתרגיל (מציאות בתחילת כל חלק רטוב)
לא יתקבל קוד עם ספריות נוספות

מומלץ לחזור על שקי ההרצאות והתרגילים הרלוונטיים לפני תחילת העבודה על התרגיל.

חלק א' – MDP (44 נק')

רקע

בחלק זה עוסוק בתחום החלטה מركובים, נתענין בתהילה עם **אפק אינסופי** (מדיניות סטציאנרית).

חלק א' - חלק היבש

1. בתרגול ראיינו את משואת בלמן כאשר הוגМОל ניתן עבור המצב הנוכחי בלבד, כלומר $\mathbb{R} \rightarrow S : R$, למtan:

הgomol זה נקרא "תגמול על האמתים" מכיוון שהוא תלוי בזומת שהסוכן נמצא בו.

בהתאם להגדרה זו הצגנו בתרגול את האלגוריתמים Value iteration ו-Policy Iteration למציאת המדיניות האופטימלית.

בעת, נרchieב את ההגדרה זו, לתגמול מקבל את המצב הנוכחי והמצב אליו הגיע הסוכן, בלחומר:

$\mathbb{R} \rightarrow S \times S : R$, למtan תגמול זה נקרא "תגמול תוצאות". לצורך שלמות ההגדרה, נגדיר שם לכל

$.R(s, s') = P(s'|s, a) = 0$ מתקיים - $\infty < R(s, s') < \infty$.

א. (1 נק') התאימו את הנוסחה של התוצאה של התועלת מהתרגול, עבור התוצאה של התועלת המתבקשת במקרה של "תגמול תוצאות", אין צורך לנמק.

$$U^\pi(s) = E_\pi \left[\sum_{t=0}^{\infty} \gamma^t R(S_t, S_{t+1}) \mid S_0 = s \right]$$

ב. (1 נק') כתבו מחדש את נוסחת משואת בלמן עבור המקרה של "תגמול תוצאות", אין צורך לנמק.

$$U(s) = \max_{a \in A(s)} \sum_{s'} P(s'|s, a) [R(s, s') + \gamma U(s')]$$

בסעיפים הבאים התייחסו גם ל McKersie ו- γ , והסבירו מה לדעתכם התנאים שצראים להתקיים על הסביבה mdp על מנת שתמיד נצליח למצוא את המדיניות האופטימלית.

ג. (2 נק') נסחו את אלגוריתם Value Iteration עבור המקרה של "תגמול תוצאה".

```

function VALUE-ITERATION( $mdp, \epsilon$ ) returns a utility function
  inputs:  $mdp$ , an MDP with states  $S$ , actions  $A(s)$ , transition model  $P(s' | s, a)$ ,
           rewards  $R(s, s')$  discount  $\gamma$ 
            $\epsilon$ , the maximum error allowed in the utility of any state
  local variables:  $U, U'$ , vectors of utilities for states in  $S$ , initially zero
                      $\delta$ , the maximum change in the utility of any state in an iteration

  repeat
     $U \leftarrow U'; \delta \leftarrow 0$ 
    for each state  $s$  in  $S$  do
       $U'[s] \leftarrow \max_{a \in A(s)} \sum_{s'} P(s'|s, a) [R(s, s') + \gamma U(s')]$ 
      if  $|U'[s] - U[s]| > \delta$  then  $\delta \leftarrow |U'[s] - U[s]|$ 
    until  $\delta < \epsilon(1 - \gamma)/\gamma$ 
  return  $U$ 

```

ד. (2 נק') נסחו את אלגוריתם Policy Iteration עבור המקרה של "תגמול תוצאה".

```

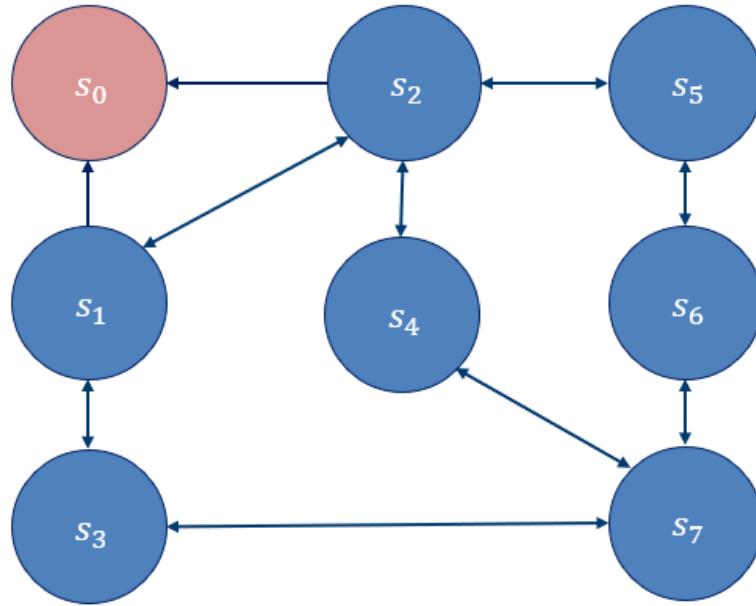
function POLICY-ITERATION( $mdp$ ) returns a policy
  inputs:  $mdp$ , an MDP with states  $S$ , actions  $A(s)$ , transition model  $P(s' | s, a)$ 
  local variables:  $U$ , a vector of utilities for states in  $S$ , initially zero
                      $\pi$ , a policy vector indexed by state, initially random

  repeat
     $U \leftarrow \text{POLICY-EVALUATION}(\pi, U, mdp)$ 
     $unchanged? \leftarrow \text{true}$ 
    for each state  $s$  in  $S$  do
      if  $\max_{a \in A(s)} \sum_{s'} P(s'|s, a) [R(s, s') + \gamma U(s')] > \sum_{s'} P(s'|s, \pi(s)) [R(s, s') + \gamma U(s')]$  then do
         $\pi[s] \leftarrow \max_{a \in A(s)} \sum_{s'} P(s'|s, a) [R(s, s') + \gamma U(s')]$ 
       $unchanged? \leftarrow \text{false}$ 
    until  $unchanged?$ 
  return  $\pi$ 

```

In policy iteration we use the expected utility function from a to calculate POLICY-EVALUATION, if $\gamma = 1$ then the Utility won't necessarily converge in both Policy and Value iteration, meaning a finite horizon is required in order to find an optimal solution for this MDP.

נתון הגרף הבא:



נתונים:

- (Discount factor) $\gamma = 0.5$
 - אופק אינסופי.
 - קבוצת המצבים – מתחאים את מילוק הסוכן בגרף. $S = \{s_0, s_1, s_2, s_3, s_4, s_5, s_6, s_7\}$
 - קבוצת המצבים הסופי. $S_G = \{s_0\}$
 - קבוצת הפעולות לכל מצב (על פי הגרף), לדוגמה: $\{\uparrow, \rightarrow\}$.
 - תגמולים ("תגמול תוצאות"):
- $$\forall s \in S, s' \in S \setminus S_G: R(s, s') = -1, \quad R(s_1, s_0) = 5, \quad R(s_2, s_0) = 7$$
- מודל המעבר הוא דטרמיניסטי, כלומר כל פעולה מצליחה בהסתברות אחת.

ה. (יבש 2 נק') הרץ את האלגוריתם Value iteration שכתבת על הגרף הנתון. ומלא את הערכים

בטבלה הבאה, באשר $0 \leq s \in S: U_0(s) = 0$. (ויתכן שלא צריך למלא את כליה).

	$U_0(s_i)$	$U_1(s_i)$	$U_2(s_i)$	$U_3(s_i)$	$U_4(s_i)$	$U_5(s_i)$	$U_6(s_i)$	$U_7(s_i)$	$U_8(s_i)$
s_1	0	5	5	5	5				
s_2	0	7	7	7	7				
s_3	0	-1	1.5	1.5	1.5				
s_4	0	-1	2.5	2.5	2.5				
s_5	0	-1	2.5	2.5	2.5				
s_6	0	-1	-1.5	0.25	0.25				
s_7	0	-1	-1.5	0.25	0.25				

ו. (יבש 2 נק') הרץ את האלגוריתם Policy iteration שכתבת על הגרף הנתון. ומלא את הערכים

בטבלה הבאה, באשר המדיניות ההתחלתית π_0 מופיעות בעמודה הראשונה בטבלה. (ויתכן שלא

צריך למלא את כליה).

הניחו שבמידה ולא קיים שיפור, האלגוריתם יבחר תמיד להשאיר את הפעולה הקודמת.

	$\pi_0(s_i)$	$\pi_1(s_i)$	$\pi_2(s_i)$	$\pi_3(s_i)$	$\pi_4(s_i)$	$\pi_5(s_i)$	$\pi_6(s_i)$	$\pi_7(s_i)$	$\pi_8(s_i)$
s_1	\downarrow	\uparrow	\uparrow	\uparrow	\uparrow				
s_2	\downarrow	\leftarrow	\leftarrow	\leftarrow	\leftarrow				
s_3	\rightarrow	\rightarrow	\uparrow	\uparrow	\uparrow				
s_4	\uparrow	\uparrow	\uparrow	\uparrow	\uparrow				
s_5	\leftarrow	\leftarrow	\leftarrow	\leftarrow	\leftarrow				
s_6	\uparrow	\uparrow	\uparrow	\uparrow	\uparrow				
s_7	\uparrow	\uparrow	\uparrow	Top-Left(S_4)	Top-Left(S_4)				

ז. (יבש 2 נק') חידר על הסעיף הקודם הקודם. הפעם עם אופק סופי באשר $2 = N$ (משמעותי לב, המדיניות לא חייבת להסתיים במצב מסוים, ישנו מצבים שלא יכולים להגיע למצב מסוים עם אופק זה. ישנו Zustand עם מספר תשובות נכונות, קיבל את כלום).

	$\pi_0(s_i)$	$\pi_1(s_i)$	$\pi_2(s_i)$	$\pi_3(s_i)$	$\pi_4(s_i)$	$\pi_5(s_i)$	$\pi_6(s_i)$	$\pi_7(s_i)$	$\pi_8(s_i)$
s_1	↓	↑	↑	↑					
s_2	↓	←	←	←					
s_3	→	→	↑	↑					
s_4	↑	↑	↑	↑					
s_5	←	←	←	←					
s_6	↑	↑	↑	↑					
s_7	↑	↑	↑	↑					

ח. (1 נק') ללא תלות בשינוי של הסעיף הקודם הקודם. אם $\gamma = 0$, מה מספר המדיניות האופטימליות הקיימות? נמקו.

$\gamma = 0$ means that there is no recursive calculation of the utility, in other words the only factor left is the reward, so this limits us to look ahead with depth 1, so in our case following the rewards given above we only get a positive reward in s_1 and s_2 and the rest have equal rewards for all available moves, so in total:

$$1*1*2*2*2*2*3= 48 - \text{Total policies}$$

ט. (1 נק') ללא תלות בשינוי של הסעיף הקודם הקודם, הסבירו מה היה קורה אם

$$R(s_1, s_2) = R(s_2, s_1) = 2, \quad \gamma = 1$$

בתשובהך, התיחס גם לערכי התוצאות של כל צומת וגם לשינוי במדיניות, אין צורך לחשב.

Here the discount factor is 1, in addition to a positive reward on the path from s_1 to s_2 and vice versa, we get the same results in the first iteration but after that s_1 will want to go to s_2 given that his utility plus the reward (according to our equation) with probability and discount factor 1 will give him more than what he already has ($5+2=7$), which will lead to the utility of s_2 changing in the next iteration accordingly and for the same reasons ($2+7=9$), and so on...

The utility won't converge as explained above, but the policy will converge to the same as before with the adjustment of s_1 and s_2 pointing at each other, never including the goal state.

חלק ב' - היברот עם הקוד

חלק זה הוא רק עבור היברот הקוד, עבורי עליו במלואו וודאו כי הינכם מבינים את הקוד.

mdp.py – אתם לא צריכים לעורר כל את הקובץ הזה.

בקובץ זה מומשת הסביבה של-mdp בתחום מחלקה MDP. הבניאי מקבל:

- board - המגדיר את המצבים האפשריים במרחב ואת התגמול לכל מצב, תגמול על הצמתים בלבד.
- terminal_states – קבוצה של המצבים הסופיים (בהכרח יש לפחות מצב אחד סופי).
- transition_function – מודל המעבר בהינתן פעולה, מה ההסתברות לכל אחת מארבע הפעולות האחרות. ההסתברויות מסודרות לפי סדר הפעולות.
- gamma – המקובל ערכיהם $\gamma \in (0,1)$.
- בתרגיל זה לא נבדוק את המקרה בו $\gamma = 1$.

הערה: קבוצת הפעולות מוגדרת בבניאי והוא קבועה לכל לוח שיבחר.

لمחלקה MDP יש מספר פונקציות שימושיות לשימוש אתכם בתרגיל.

- print_rewards() – מדפסה את הלוח עם ערך התגמול בכל מצב.
- print_utility(U) – מדפסה את הלוח עם ערך התועלת U לכל מצב.
- print_policy(policy) – מדפסה את הלוח עם הפעולה שהمدיניות policy נתנה לכל מצב שהוא לא מצב סופי.
- step(state, action) – בהינתן מצב הנוכחי state ופעולה action מחזיר את המצב הבא באופן דטרמיניסטי. עברו הליכה לכיוון קיר או יצא מהלוח הפונקציה תחזיר את המצב הנוכחי state.

חלק ג' – רטוב

כל הקוד צריך להיבתב בקובץ `py.mdp_implementation.py` בקורס:
מותר להשתמש בספריות:

All the built-in packages in python, numpy, matplotlib, argparse, os, copy, typing, termcolor, random

עליכם למש את הfonקציות הבאות:

- (רטוב 6 נק'): `value_iteration(mdp, U_init, epsilon)` – בהינתן ה-mdp, וערך התועלת ההתחלתי `U`, וחסם העליון לשגיאה מהתחלה של התועלת האופטימלי `epsilon` מרים את האלגוריתם `value iteration` ומחזיר את `U` המתקבל בסוף ריצת האלגוריתם. **TODO**
- (רטוב 4 נק'): `get_policy(mdp, U)` – בהינתן ה-mdp וערך התועלת `U` (המקים את משוואת בלמן) מוחזיר את המדיניות (במידה וקיים יותר מ אחת, מוחזיר אחת מהן). **TODO**
- (רטוב 4 נק'): `policy_evaluation(mdp, policy)` – בהינתן ה-mdp, ומדיניות `policy` מוחזיר את ערכי התועלת לכל מצב. **TODO**
- (רטוב 6 נק'): `policy_iteration(mdp, policy_init)` – בהינתן ה-mdp, ומדיניות התחלתיות `policy_init`, מרים את האלגוריתם `policy iteration` ומוחזיר מדיניות אופטימלית. **TODO**

לאור העובדה שהfonקציות הבאות לא נוסחו באופן ברור בתרגיל, יצא הסבר מפורט יותר על

הדים המדרשים פה ← <https://piazza.com/class/lrurdsbmuiww0/post/336>

- (רטוב 5 נק'): `((-3)*10**(-3), get_all_policies(mdp, U, epsilon=10**(-3)))` – בהינתן ה-mdp, וערך התועלת `U` (המקים את משוואת בלמן) מדפיס\מציג את כל המדיניות המקיימות ערך זה בלבד (יש לבצע ויזואлизציה להציג כל המדיניות),

★	★	←	+1
★	■	←	-1
★	★	★	↓

הfonקציה מחזירה את מספר המדיניות (`policies`) השונות הקיימות

המקיימות את `U` (בדוגמה 4^6). **TODO**.

לדוגמא:

עליכם להדפיס את הלח (המדינהות) עבור התועלת שנמצאת בקובץ `U_for_get_all_policies`

Running the segel Utility function:

Board Utilities

0.749	0.819	0.876	1.0
0.692	WALL	0.564	-1.0
0.623	0.566	0.518	0.252

Matching policies

\rightarrow	\rightarrow	\rightarrow	$+1$
\uparrow	WALL	\leftarrow	-1
\uparrow	\leftarrow	\leftarrow	\downarrow

Number of different policies aligned with the utility above: 1

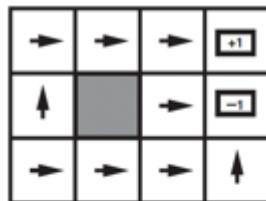
• (רטוב 5 נק'): (`get_policy_for_different_rewards(mdp, epsilon)` – בהינתן ה-mdp).

מופיע\מציג את המדיניות האופטימלית בתלות ב- R (ערכי התגמול לכל מצב שאינו סופי).

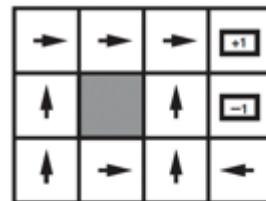
יש להציג רשימה של ערכי R שבהם יש שינויים במדיניות מהקטן גדול. **TODO**

ניתן להניח שלא יהיו שינויים במדיניות עבור ערכי R קטנים מ- 5 – גדולים מ- 5 . בנוסף, דיק של 2 ספרות אחרי הקודעה הינו מספק.

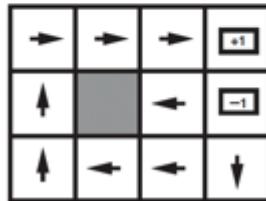
דוגמא חלקית של פתרון אפשרי:



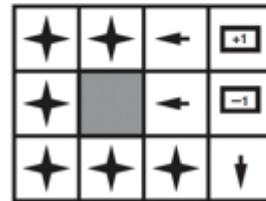
$$R(s) < -1.6284$$



$$-0.4278 < R(s) < -0.0850$$



$$-0.0221 < R(s) < 0$$



$$R(s) > 0$$

בנוסף לקוד עליו לארף להגשה היבשה את התוצאות של הפונקציות על הסביבה שניתנה בתרגיל.

@@@@aa				-0.63 <= R(s) < -0.62
@@@@aaaaaa@ Policies for different rewards: @@				
@@@@aaa				
→ → → +1	→ WALL ↑ -1	↑ → ↑ ↑	↑ → ↑ ↑	
-5.0 <= R(s) < -1.48				-0.62 <= R(s) < -0.61
→ → → +1	→ WALL ↑ -1	↑ → ↑ ↑	↑ → ↑ ↑	
↑ WALL → -1	↑ → ↑ ↑	→ → → +1	→ → ↑ -1	
→ → → ↑	→ → → ↑	→ → ↑ ↑	→ → ↑ ↑	
-1.48 <= R(s) < -1.45				-0.61 <= R(s) < -0.59
→ → → +1	→ WALL ↑ -1	↑ → ↑ ↑	↑ → ↑ ↑	
↑ WALL ↑ -1	↑ → ↑ ↑	→ → → +1	→ → ↑ -1	
→ → → ↑	→ → → ↑	→ → ↑ ↑	→ → ↑ ↑	
-1.45 <= R(s) < -1.31				-0.59 <= R(s) < -0.37
→ → → +1	→ WALL ↑ -1	↑ → ↑ ↑	↑ → ↑ ↑	
↑ WALL ↑ -1	↑ → ↑ ↑	→ → → +1	→ → ↑ -1	
→ → → ↑	→ → → ↑	→ → ↑ ↑	→ → ↑ ↑	
-1.31 <= R(s) < -1.28				-0.37 <= R(s) < -0.06
→ → → +1	→ WALL ↑ -1	↑ → ↑ ↑	↑ → ↑ ↑	
↑ WALL ↑ -1	↑ → ↑ ↑	→ → → +1	→ → ↑ -1	
→ → ↑ ↑	→ → → ↑	→ → ↑ ↑	→ → ↑ ←	
-1.28 <= R(s) < -0.63				-0.06 <= R(s) < -0.04
→ → → +1	→ WALL ↑ -1	↑ → ↑ ↑	↑ → ↑ ↑	
↑ WALL ↑ -1	↑ → ↑ ↑	→ → → +1	→ → ↑ -1	
→ → ↑ ↑	→ → → ↑	→ → ↑ ↑	→ → ↑ ←	

$-0.04 \leq R(s) < -0.03$	$0.05 \leq R(s) < 0.06$
$\begin{array}{ c c c c c } \hline \rightarrow & \rightarrow & \rightarrow & +1 & \\ \hline \uparrow & WALL & \uparrow & -1 & \\ \hline \uparrow & \leftarrow & \uparrow & \downarrow \leftarrow & \\ \hline \end{array}$	$\begin{array}{ c c c c c } \hline \rightarrow & \rightarrow & \rightarrow & +1 & \\ \hline \uparrow & WALL & \leftarrow & -1 & \\ \hline \uparrow & \leftarrow & \uparrow & \downarrow & \\ \hline \end{array}$
$-0.03 \leq R(s) < 0.00$	$0.06 \leq R(s) < 0.08$
$\begin{array}{ c c c c c } \hline \rightarrow & \rightarrow & \rightarrow & +1 & \\ \hline \uparrow & WALL & \uparrow & -1 & \\ \hline \uparrow & \leftarrow & \uparrow & \downarrow & \\ \hline \end{array}$	$\begin{array}{ c c c c c } \hline \rightarrow & \rightarrow & \rightarrow & +1 & \\ \hline \uparrow & WALL & \leftarrow & -1 & \\ \hline \uparrow & \leftarrow & \uparrow \leftarrow & \downarrow & \\ \hline \end{array}$
$0.00 \leq R(s) < 0.01$	$0.08 \leq R(s) < 0.09$
$\begin{array}{ c c c c c } \hline \uparrow \downarrow \leftarrow & \uparrow \downarrow \leftarrow & \uparrow \downarrow \leftarrow & +1 & \\ \hline \uparrow \downarrow \leftarrow & WALL & \uparrow \downarrow \leftarrow & -1 & \\ \hline \uparrow \downarrow \leftarrow & \uparrow \downarrow \leftarrow & \uparrow \downarrow \leftarrow & \uparrow \downarrow \leftarrow & \\ \hline \end{array}$	$\begin{array}{ c c c c c } \hline \rightarrow & \rightarrow & \rightarrow & +1 & \\ \hline \uparrow & WALL & \leftarrow & -1 & \\ \hline \uparrow & \leftarrow & \uparrow & \downarrow & \\ \hline \end{array}$
$0.01 \leq R(s) < 0.03$	$0.09 \leq R(s) < 0.10$
$\begin{array}{ c c c c c } \hline \rightarrow & \rightarrow & \rightarrow & +1 & \\ \hline \uparrow & WALL & \leftarrow & -1 & \\ \hline \uparrow & \leftarrow & \uparrow & \downarrow & \\ \hline \end{array}$	$\begin{array}{ c c c c c } \hline \uparrow \rightarrow & \rightarrow & \rightarrow & +1 & \\ \hline \uparrow & WALL & \leftarrow & -1 & \\ \hline \uparrow \downarrow \leftarrow & \leftarrow & \uparrow \downarrow \leftarrow & \downarrow & \\ \hline \end{array}$
$0.03 \leq R(s) < 0.05$	$0.10 \leq R(s) < 0.11$
$\begin{array}{ c c c c c } \hline \rightarrow & \rightarrow & \rightarrow & +1 & \\ \hline \uparrow & WALL & \leftarrow & -1 & \\ \hline \uparrow & \leftarrow & \uparrow \leftarrow & \downarrow & \\ \hline \end{array}$	$\begin{array}{ c c c c c } \hline \uparrow \downarrow \leftarrow & \uparrow \downarrow \leftarrow & \uparrow \downarrow \leftarrow & +1 & \\ \hline \uparrow \downarrow \leftarrow & WALL & \leftarrow & -1 & \\ \hline \uparrow \downarrow \leftarrow & \uparrow \downarrow \leftarrow & \uparrow \downarrow \leftarrow & \downarrow & \\ \hline \end{array}$

$0.11 \leq R(s) \leq 5.00$
 $\begin{array}{|c|c|c|c|c|} \hline \uparrow \downarrow \leftarrow & \uparrow \downarrow \leftarrow & \leftarrow & +1 & | \\ \hline \uparrow \downarrow \leftarrow & WALL & \leftarrow & -1 & | \\ \hline \uparrow \downarrow \leftarrow & \uparrow \downarrow \leftarrow & \uparrow \downarrow \leftarrow & \downarrow & | \\ \hline \end{array}$
 $[-1.48, -1.45, -1.31, -1.28, -0.63, -0.62, -0.61, -0.59, -0.37, -0.06, -0.04, -0.03, 0.0, 0.01, 0.03, 0.05, 0.06, 0.08, 0.09, 0.1, 0.11]$
 Done!

עבור מצבים סופיים וקירות (WALL), הערך שצורך לחזור בתאים אלו עבור טבלאות המדיניות הוא `None`. כל ערך אחר לא יתקבל בתשובה.

main.py – דוגמת ריצה לשימוש בכל הפונקציות.

בתחילת הקובץ אנו טוענים את הסביבה משלושה קבצים:
board, terminal_states, transition_function
ויצרים מופע של הסביבה (mdp).

- שימושו לב, שברגע הקוד ב-main לא יוכל לחש מכיוון שאתם צריכים להשלים את הפונקציות הרלוונטיות ב-`mdp_implementation.py`.
- בנוסף, על מנת לראות את הלוח עם הצבעים עליהם להריץ את הקוד ב-IDE לדוגמה PyCharm.

הוראות הגשה

- ✓ הגשת התרגיל תבוצע אלקטרונית בזוגות בלבד.
- ✓ הקוד שלכם יבדק (גם) באופן אוטומטי ולכן יש להקפיד על הפורמט המבוקש. הגשה שלא עומדת בפורמט לא תיבדק (ציון 0).
- ✓ המצאת נתונים לצורך בניית הגרפים אסורה ומהויה עבירה ממשמעת.
- ✓ הקפידו על קוד קרייא ומתועד. התשובות בדוח צירכות להופיע לפי הסדר.
- ✓ יש להגיש קובץ压缩 יחיד בשם `zip.zip<id1><id2><id3>.AI` (ללא סוגרים משולשים) שמכיל:
 - קובץ בשם `MDP_HW3_AI.MDP.PDF` המכיל את תשובהיכם לשאלות היבשות.
 - קבצי הקוד שנדרשתם למש בתרגיל ואף קובץ אחר:
`mdp_implementation.py` -

אין להכיל תיקיות בקובץ ההגשה, הגשה שלא עומדת בפורמט לא תיבדק.

נספח MDP

דוגמת הרצה (שימוש לב שරצה זו השתמשה במודל הסתברותי שונה משלכם)

יצירת הסביבה:

```
mdp = MDP(board=board_env,  
           terminal_states=terminal_states_env,  
           transition_function=transition_function_env,  
           gamma=1.0)
```

הדפסת הלוח עם התגמולים לכל מצב:

```
print('ooooooooooooooooooooo')  
print("oooooo The board and rewards oooooo")  
print('oooooooooooooooooooo')  
print(mdp.print_rewards())
```

פלט:

```
ooooooooooooooooooooo  
oooooo The board and rewards oooooo  
oooooooooooooooooooo  
| -0.04 | -0.04 | -0.04 | +1 |  
| -0.04 | WALL | -0.04 | -1 |  
| -0.04 | -0.04 | -0.04 | -0.04 |
```

:Value iteration

```
print('oooooooooooooooooooo')  
print("oooooo Value iteration oooooo")  
print('oooooooooooooooooooo')  
  
U = [[0, 0, 0, 0],  
      [0, 0, 0, 0],  
      [0, 0, 0, 0]]  
print("\nInitial utility:")  
mdp.print_utility(U)  
print("\nFinal utility:")  
U_new = value_iteration(mdp, U)  
mdp.print_utility(U_new)  
print("\nFinal policy:")  
policy = get_policy(mdp, U_new)  
mdp.print_policy(policy)
```

פלט:

```
oooooooooooooooooooo  
oooooo Value iteration oooooo  
oooooooooooooooooooo  
  
Initial utility:  
| 0.0 | 0.0 | 0.0 | 0.0 |  
| 0.0 | WALL | 0.0 | 0.0 |  
| 0.0 | 0.0 | 0.0 | 0.0 |  
  
Final utility:  
| 0.812 | 0.868 | 0.918 | 1.0 |  
| 0.762 | WALL | 0.66 | -1.0 |  
| 0.705 | 0.655 | 0.611 | 0.388 |  
  
Final policy:  
| RIGHT | RIGHT | RIGHT | +1 |  
| UP | WALL | UP | -1 |  
| UP | LEFT | LEFT | LEFT |
```

:Policy iteration

```
print('@@@@@@@@@@@ Policy iteration @@@@')  
print('@@@@@@@ Policy iteration @@@@')  
print('@@@@@@@ Policy iteration @@@@')  
  
print("\nPolicy evaluation:")  
U_eval = policy_evaluation(mdp, policy)  
mdp.print_utility(U_eval)  
  
policy = [['UP', 'UP', 'UP', 0],  
          ['UP', 'WALL', 'UP', 0],  
          ['UP', 'UP', 'UP', 'UP']]  
print("\nInitial policy:")  
mdp.print_policy(policy)  
print("\nFinal policy:")  
policy_new = policy_iteration(mdp, policy)  
mdp.print_policy(policy_new)  
  
print("Done!")
```

פָלְטָן

```
@@@@@@@ Policy iteration @@@@  
@@@@@@@ Policy iteration @@@@  
@@@@@@@ Policy iteration @@@@  
  
Policy evaluation:  
| 0.812 | 0.868 | 0.918 | 1.0 |  
| 0.762 | WALL | 0.66 | -1.0 |  
| 0.705 | 0.655 | 0.611 | 0.388 |  
  
Initial policy:  
| UP | UP | UP | +1 |  
| UP | WALL | UP | -1 |  
| UP | UP | UP | UP |  
  
Final policy:  
| RIGHT | RIGHT | RIGHT | +1 |  
| UP | WALL | UP | -1 |  
| UP | LEFT | LEFT | LEFT |  
  
Done!
```