



## תחרות רובוטים

חוקרי AI באוניברסיטת סגד מארגנים תחרות תכנות רובוטים. חברתכם, שרה, החליטה להשתתף בתחרות. המטרה היא לתכנת את הפוליבוט האוליטימטיבי, מתוך הערצה לאינטליגנציה הגבוהה של גזע כלבי הרועים ההונגרי, הפולי.

פוליבוט יבחן במבוך המורכב מטבלה של  $(H + 2) \times (W + 2)$  תאים. השורות בטבלה ממוספרות מ-1 עד  $H$  מצפון לדרום והעמודות בטבלה ממוספרות מ-1 עד  $W$  ממערב למזרח. אנו נתייחס לתא שממוקם בשורה  $r$  ובעמודה  $c$  של הטבלה  $(-1 \leq c \leq W, -1 \leq r \leq H)$  כתא  $(r, c)$ .

הביטוי בתא  $(r, c)$  כאשר  $0 \leq r < H$  ו-  $0 \leq c < W$  ישנם 4 תאים סמוכים לתא  $(r, c)$ :

- התא  $(r, c - 1)$  מכונה התא **ממערב** לתא  $(r, c)$ ;
- התא  $(r + 1, c)$  מכונה התא **מדרום** לתא  $(r, c)$ ;
- התא  $(r, c + 1)$  מכונה התא **ממזרח** לתא  $(r, c)$ ;
- התא  $(r - 1, c)$  מכונה התא **מצפון** לתא  $(r, c)$ .

התא  $(r, c)$  נקרא תא **היקפי** של המבוך אם  $r = -1$  או  $r = H$  או  $c = -1$  או  $c = W$  מתקיימים. כל תא שאינו תא היקפי במבוך הוא או תא **מכשול** או תא **ריק**. בנוסף, לכל תא ריק יש **צבע**, המיוצג על ידי מספר שלם אי-שלילי בין 0 ל- $Z_{MAX}$ , כולל. תחילה, הצבע של כל תא ריק הוא 0.

לדוגמה, הביטוי במבוך עם  $H = 4$  ו-  $W = 5$ , המכיל תא מכשול אחד  $(1, 3)$ :

	-1	0	1	2	3	4	5
-1							
0		0	0	0	0	0	
1		0	0	0		0	
2		0	0	0	0	0	
3		0	0	0	0	0	
4							

תא המכשול היחיד מסומן ב-X. תאים היקפיים במבוך מוצללים. המספר הרשום בכל תא ריק מייצג את הצבע שלו.

**מסלול** באורך  $\ell$  ( $\ell > 0$ ) מהתא  $(r_0, c_0)$  לתא  $(r_\ell, c_\ell)$  הוא סדרה של תאים **ריקים** השונים זה מזה  $(r_0, c_0), (r_1, c_1), \dots, (r_\ell, c_\ell)$  בה לכל  $i$  ( $0 \leq i < \ell$ ) התאים  $(r_i, c_i)$  ו-  $(r_{i+1}, c_{i+1})$  סמוכים.

שימו לב שמסלול באורך  $\ell$  מכיל בדיוק  $\ell + 1$  תאים.

בתחרות, החוקרים מקימים מבוך שבו ישנו לפחות מסלול אחד מהתא  $(0,0)$  לתא  $(H-1, W-1)$ . שימו לב שזה גורר שמובטח שהתאים  $(0,0)$  ו- $(H-1, W-1)$  ריקים.

שרה לא יודעת אילו תאים במבוך ריקים ואילו תאים הם מכשולים.

משימתכם היא לעזור לשרה לתכנת את פוליבוט כך שהוא יהיה מסוגל למצוא מסלול קצר ביותר (כלומר, מסלול באורך מינימלי) מהתא  $(0,0)$  לתא  $(H-1, W-1)$  במבוך הלא ידוע שיוקם על ידי החוקרים. המפרט של פוליבוט והחוקים של התחרות מתוארים מטה.

שימו לב שהחלק האחרון של הסטייטמנט לבעיה זו מתאר כלי הדמיה שאתם יכולים להשתמש בו כדי לעשות ויזואליזציה לפוליבוט.

## המפרט של פוליבוט

נגדיר את המצב של התא  $(r, c)$  לכל  $0 \leq r \leq H-1$  ו- $0 \leq c \leq W-1$  להיות מספר שלם כך ש:

- אם התא  $(r, c)$  הוא תא היקפי אז המצב שלו הוא  $-2$ ;
- אם התא  $(r, c)$  הוא תא מכשול אז המצב שלו הוא  $-1$ ;
- אם התא  $(r, c)$  הוא תא ריק אז המצב שלו הוא הצבע של התא.

התוכנית של פוליבוט מורצת כרצף של מהלכים. בכל מהלך, פוליבוט מזהה את המצבים של תאים סמוכים ואז מבצע הוראה. ההוראה שהוא מבצע נקבעת על ידי המצב המזוהה. להלן תיאור מדויק יותר.

הניחו שבתחילת המהלך הנוכחי, פוליבוט נמצא בתא  $(r, c)$ , שהוא תא ריק. המהלך מבוצע כדלקמן:

1. תחילה, פוליבוט מזהה את מערך המצבים הנוכחי, כלומר, את המערך  $S = [S[0], S[1], S[2], S[3], S[4]]$ ,

המורכב מהמצב של התא  $(r, c)$  ושל כל התאים הסמוכים:

- $S[0]$  זה המצב של התא  $(r, c)$ .
- $S[1]$  זה המצב של התא ממערב.
- $S[2]$  זה המצב של התא מדרום.
- $S[3]$  זה המצב של התא ממזרח.
- $S[4]$  זה המצב של התא מצפון.

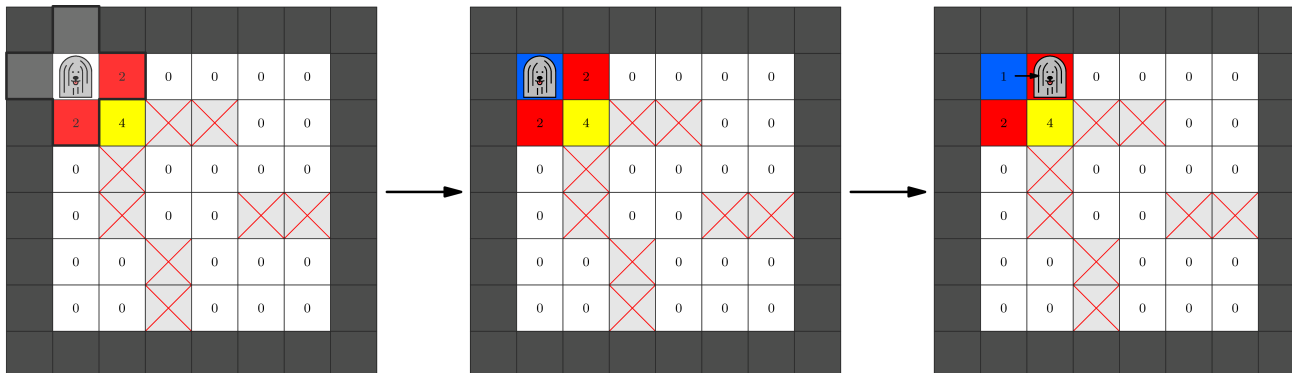
2. לאחר מכן, פוליבוט קובע את ההוראה  $(Z, A)$  שמתאימה למערך המצבים המזוהה.

3. לבסוף, פוליבוט מבצע את ההוראה הזו: הוא קובע את הצבע של התא  $(r, c)$  לצבע  $Z$  ואז הוא מבצע את

הפעולה  $A$ , שהיא אחת מבין הפעולות הבאות:

- הישאר בתא  $(r, c)$ ;
- זוז לאחד מ-4 התאים הסמוכים;
- סיים את התוכנית שלך.

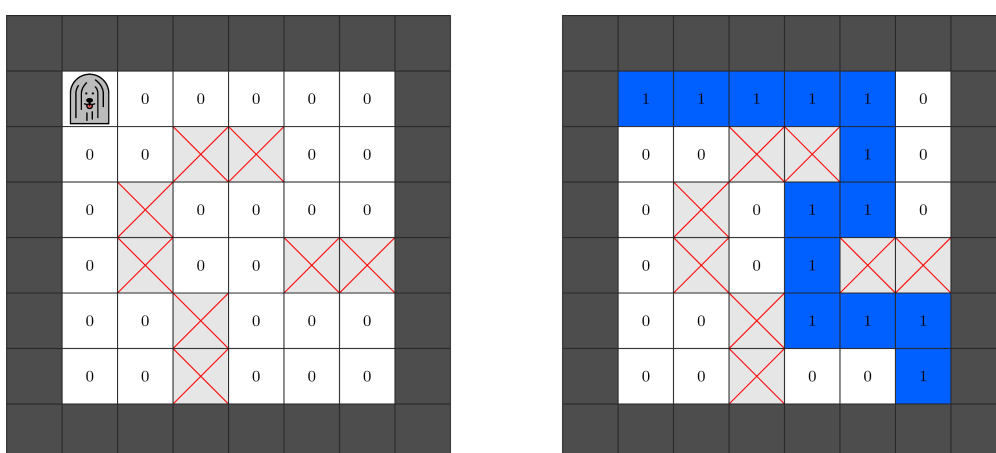
לדוגמה, הביטו בתרחיש המוצג משמאל של האיור הבא. פוליבוט נמצא כרגע בתא  $(0,0)$  שבצבע 0. פוליבוט מזהה את מערך המצבים  $S = [0, -2, 2, 2, -2]$ . יכולה להיות לפוליבוט תוכנית שבה, ברגע שהוא מזהה את מערך זה, הוא קובע את צבע התא הנוכחי להיות  $Z = 1$  ואז זז מזרחה, כפי שמוצג באמצע ומימין של האיור הבא:



## חוקי תחרות הרובוטים

- בהתחלה, פוליבוט ממוקם בתא  $(0,0)$  ומתחיל להריץ את התוכנית שלו.
- לפוליבוט אסור לזוז לתא שאינו ריק.
- התוכנית של פוליבוט חייבת להסתיים אחרי לכל היותר 500 000 מהלכים.
- אחרי שהתוכנית של פוליבוט מסתיימת, תאים ריקים במבוך צריכים להיות צבועים כך ש:
  - קיים מסלול קצר ביותר מ- $(0,0)$  ל- $(H-1, W-1)$  שעבורו הצבע של כל תא שכלול במסלול הוא 1.
  - לכל שאר התאים הריקים יש את הצבע 0.
- פוליבוט יכול לסיים את התוכנית שלו בכל תא ריק.

לדוגמה, האיור הבא מציג מבוך אפשרי עם  $H = W = 6$ . הקונפיגורציה ההתחלתית מוצגת משמאל וצביעה אחת מקובלת של התאים הריקים לאחר הסיום מוצגת מימין:



## פרטי מימוש

עליכם לממש את הפונקציה הבאה.

```
void program_pulibot()
```

- על פונקציה זו ליצור את התוכנית של פוליבוט. פונקציה זו צריכה לעבוד נכון לכל ערכי  $H$  ו- $W$  ולכל מבוך שעומד במגבלות המשימה.
- פונקציה זו נקראת פעם אחת בדיוק עבור כל טסטקייס.

פונקציה זו יכולה לבצע קריאות לפונקציה הבאה כדי ליצור את התוכנית של פוליבוט:

```
void set_instruction(int[] S, int Z, char A)
```

- $S$ : מערך באורך 5 המתאר מערך מצבים.
- $Z$ : מספר שלם אי-שלילי המייצג צבע.
- $A$ : תו יחיד המייצג פעולה של פוליבוט כדלקמן:
  - H: הישאר;
  - W: זוז מערבה;
  - S: זוז דרומה;
  - E: זוז מזרחה;
  - N: זוז צפונה;
  - T: סיים את התוכנית.

- קריאה לפונקציה זו מורה לפוליבוט שברגע שהוא מזהה את מערך המצבים  $S$  עליו לבצע את ההוראה  $(Z, A)$ .

קריאה לפונקציה זו מספר פעמים עם אותו מערך המצבים  $S$  תוביל למשוב Output isn't correct.

לא נדרש לקרוא ל-set\_instruction עם כל מערך מצבים  $S$  אפשרי. אבל, אם פוליבוט מזהה לאחר מכן מערך מצבים עבורו עוד לא הוגדרה הוראה, תקבלו את המשוב Output isn't correct.

אחרי ש-program\_pulibot מסתיימת, הגריידר מריץ את התוכנית של פוליבוט על מבוך אחד או יותר. הרצות אלו לא נחשבות כחלק ממגבלת זמן הריצה של הפתרון שלכם. הגריידר לא אדפטיבי, כלומר, אוסף המבוכים מוגדר מראש בכל טסטקייס.

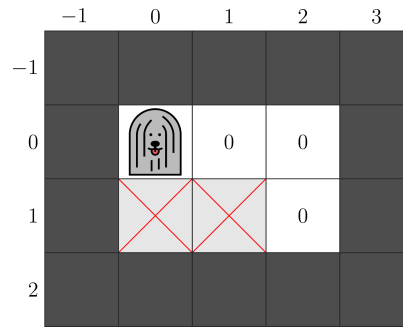
אם פוליבוט מפר את אחד מחוקי תחרות הרובוטים לפני שהוא מסיים את התוכנית שלו, תקבלו את המשוב Output isn't correct.

## דוגמה

הפונקציה program\_pulibot יכולה לבצע קריאות ל-set\_instruction כדלהלן:

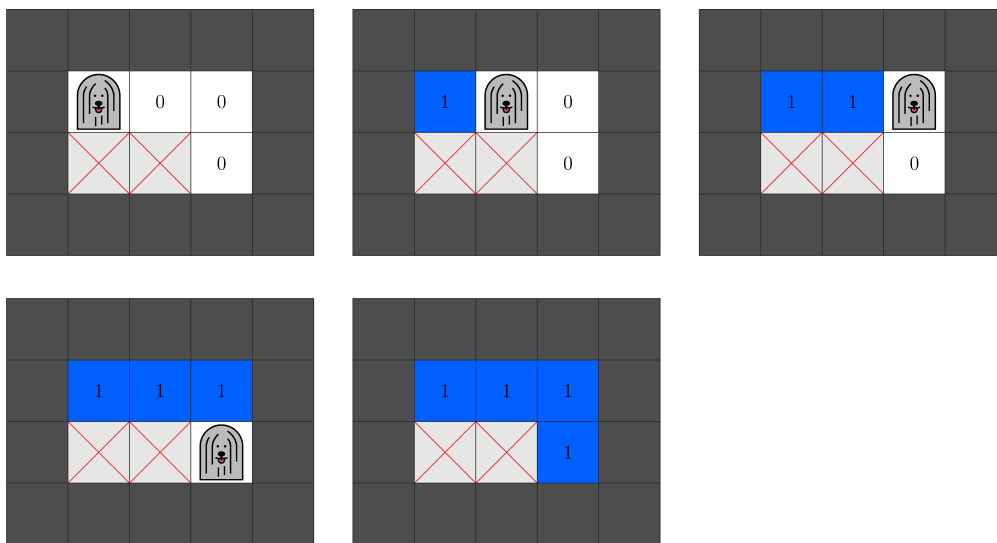
קריאה	$S$ הוראה למערך המצבים
set_instruction([0, -2, -1, 0, -2], 1, E)	קבע את הצבע ל-1 וזוז מזרחה
set_instruction([0, 1, -1, 0, -2], 1, E)	קבע את הצבע ל-1 וזוז מזרחה
set_instruction([0, 1, 0, -2, -2], 1, S)	קבע את הצבע ל-1 וזוז דרומה
set_instruction([0, -1, -2, -2, 1], 1, T)	קבע את הצבע ל-1 וסיים את התוכנית

הביטוי בתרחיש שבו  $H = 2$  ו- $W = 3$ , והמבוך כפי שמוצג באיור הבא.



עבור המבוך המסוים הזה התוכנית של פוליבוט רצה בארבעה מהלכים. מערך המצבים שפוליבוט מזהה וההוראות שהוא מבצע תואמות בדיוק לארבעת הקריאות ל-`set_instruction` שבוצעו לעיל, לפי הסדר. האחרונה מבין ההוראות הללו מסיימת את התוכנית.

האיור הבא מראה את המבוך לפני כל אחד מארבעת המהלכים ואת הצבעים הסופיים לאחר הסיום.



אף אל פי כן, שימו לב שהתוכנית בעלת 4 הפעולות הזו עלולה לא למצוא את המסלול הקצר ביותר במבוכים חוקיים אחרים. לכן, אם תוגש, היא תקבל את המשוב `Output isn't correct`.

## מגבלות

$Z_{MAX} = 19$ . לכן, פוליבוט יכול להשתמש בצבעים 0 עד 19, כולל.

עבור כל מבוך שמשתמשים בו כדי לבדוק את פוליבוט:

- $2 \leq H, W \leq 15$
- יש לפחות מסלול אחד מהתא  $(0, 0)$  לתא  $(H - 1, W - 1)$ .

## תתי משימות

1. (6 נקודות) אין תא מכשול במבוך.
2. (10 נקודות)  $H = 2$
3. (18 נקודות) יש בדיוק מסלול אחד בין כל זוג של תאים ריקים.
4. (20 נקודות) כל מסלול קצר ביותר מהתא  $(0, 0)$  לתא  $(H - 1, W - 1)$  הוא באורך  $H + W - 2$ .

5. (46 נקודות) ללא מגבלות נוספות.

אם, בטסטיקייס כלשהו, הקריאות לפונקציה `set_instruction` או התוכנית של פוליבוט לאורך ריצתה לא מקיימות את המגבלות המתוארות בפרטי המימוש, הניקוד של הפתרון שלכם לתת משימה זו יהיה 0.

בכל תת משימה, אתם יכולים להשיג ניקוד חלקי על ידי יצירת צביעה שהיא כמעט נכונה.

פורמלית:

- הפתרון לטסטיקייס הוא **מלא** אם הצביעה הסופית של התאים הריקים מקיימת את חוקי תחרות הרובוטים.
- הפתרון לטסטיקייס הוא **חלקי** אם הצביעה הסופית נראית כדלקמן:
  - קיים מסלול קצר ביותר מ- $(0, 0)$  ל- $(H - 1, W - 1)$  עבורו הצבע של כל צומת שנמצא במסלול הוא 1.
  - אין תא ריק אחר בטבלה בצבע 1.
  - תא ריק כלשהו בטבלה הוא בצבע שונה מ-0 ומ-1.

אם הפתרון שלכם לטסטיקייס הוא לא מלא ולא חלקי, הניקוד שלכם בטסטיקייס המתאים יהיה 0.

בתתי משימות 1-4, הניקוד עבור פתרון מלא הוא 100% והניקוד עבור פתרון חלקי לטסטיקייס הוא 50% מהנקודות של תת המשימה שלו.

בתת משימה 5, הניקוד שלכם תלוי במספר הצבעים שמשתמשים בהם בתוכנית של פוליבוט. ליתר דיוק, נסמן ב- $Z^*$  את הערך המקסימלי של  $Z$  על פני כל הקריאות שנעשו ל-`set_instruction`. הניקוד של כל טסטיקייס מחושב לפי הטבלה הבאה:

תנאי	ניקוד (מלא)	ניקוד (חלקי)
$11 \leq Z^* \leq 19$	$20 + (19 - Z^*)$	$12 + (19 - Z^*)$
$Z^* = 10$	31	23
$Z^* = 9$	34	26
$Z^* = 8$	38	29
$Z^* = 7$	42	32
$Z^* \leq 6$	46	36

הניקוד עבור כל תת משימה הוא הניקוד המינימלי של טסטיקייס בתת משימה זו.

## גריידר לדוגמה

הגריידר לדוגמה קורא את הקלט בפורמט הבא:

- שורה 1:  $H \ W$
- שורה  $2 + r$ :  $0 \leq r < H$ :  $m[r][0] \ m[r][1] \ \dots \ m[r][W - 1]$

כאן,  $m$  הוא מערך של  $H$  מערכים של  $W$  מספרים שלמים, המתארים את התאים הלא היקפיים של המבוך.  $m[r][c] = 0$  אם התא  $(r, c)$  הוא תא ריק ו- $m[r][c] = 1$  אם התא  $(r, c)$  הוא תא מכשול.

הגרייידר לדוגמה קורא תחילה ל-`program_pulibot()`. אם הגרייידר לדוגמה מזהה הפרה של הפרוטוקול, הגרייידר לדוגמה מדפיס `Protocol Violation: <MSG>` ומסיים את ריצתו, כאשר `<MSG>` זו אחת מהודעות השגיאה הבאות:

- `Invalid array`:  $-2 \leq S[i] \leq Z_{MAX}$  לא מתקיים עבור  $i$  כלשהו, או שהאורך של  $S$  הוא לא 5.
- `Invalid color`:  $0 \leq Z \leq Z_{MAX}$  לא מתקיים.
- `Invalid action`: התו  $A$  הוא לא אחד מבין `H, W, S, E, N` או `T`.
- `Same state array`: `set_instruction` נקראת עם אותו המערך  $S$  לפחות פעמיים.

אחרת, כש-`program_pulibot` מסתיימת, הגרייידר לדוגמה מריץ את התוכנית של פוליבוט במבוך שמתואר בקלט.

הגרייידר לדוגמה מייצר שני פלטים.

תחילה, הגרייידר לדוגמה רושם לוג של הפעולות של פוליבוט במבוך לקובץ `robot.bin` בתיקיית העבודה. קובץ זה מהווה את הקלט לכלי הויזואליזציה המתואר בחלק הבא.

שנית, אם התוכנית של פוליבוט לא מסתיימת בהצלחה, הגרייידר לדוגמה מדפיס את אחת מהודעות השגיאה הבאות:

- `Unexpected state`: פוליבוט זיהה מערך מצבים ש-`set_instruction` לא נקראה איתו.
- `Invalid move`: ביצוע פעולה הוביל להגעה של פוליבוט לתא לא ריק.
- `Too many steps`: פוליבוט ביצע 500 000 מהלכים מבלי לסיים את ריצת התוכנית שלו.

אחרת, יהי  $e[r][c]$  מצב התא  $(r, c)$  אחרי שהתוכנית של פוליבוט מסתיימת. הגרייידר לדוגמה מדפיס  $H$  שורות בפורמט הבא:

- שורה  $r + 1$  ( $0 \leq r < H$ ):  $e[r][0] \ e[r][1] \ \dots \ e[r][W - 1]$

## כלי המחשה

חבילת הקבצים המצורפת למשימה זו מכילה קובץ בשם `display.py`. בעת ההרצה, סקריפט פייתון זה מציג את הפעולות של פוליבוט במבוך המתואר על ידי הקלט לגרייידר לדוגמה. לצורך כך, הקובץ הבינארי `robot.bin` חייב להמצא בתיקיית העבודה.

על מנת להריץ את הסקריפט, הריצו את הפקודה הבאה.

```
python3 display.py
```

ממשק גרפי פשוט יופיע. הפיצ'רים העיקריים הם כדלקמן:

- אתם יכולים לראות את הסטטוס של המבוך השלם. המיקום הנוכחי של פוליבוט מודגש על ידי מלבן.

- אתם יכולים לדפדף בין המהלכים של פוליבוט על ידי לחיצה על כפתורי החיצים או לחיצה על ה-hotkeys שלהם. אתם גם יכולים לקפוץ למהלך ספציפי.
- המהלך הבא בתוכנית של פוליבוט מוצג בתחתית. מוצגים מערך המצבים הנוכחי וההוראה שתבוצע. לאחר הצעד הסופי, מוצגת או את אחת מהודעות השגיאה של הגריידר, או Terminated אם התוכנית מסתיימת בהצלחה.
- לכל מספר שמייצג צבע, אתם יכולים להקצות צבע רקע חזותי, וכן טקסט לתצוגה. הטקסט לתצוגה הוא מחרוזת קצרה שתופיע בכל תא בצבע הזה. אתם יכולים להקצות צבעי רקע וטקסטים לתצוגה באחת מהדרכים הבאות:
  - קבעו אותם בחלון הדיאלוג אחרי לחיצה על כפתור ה-Colors.
  - ערכו את התוכן של הקובץ colors.txt.
- כדי לטעון מחדש את robot.bin, השתמשו בכפתור ה-Reload. זה שימושי אם התוכן של robot.bin השתנה.