

תרגיל בית 2 - פתרון בעיות על ידי MDP

מבוא

מטרת התרגיל היא להתנסות בפתרון אחת הגרסאות של משחק zuma בעזרת MDP עם מודל ספציפי.



המטלה הזאת אינה תלויה במטלה ה-1 וקיים שינויים למבנה המשחק.

עקרון המשחק הוא שהשחקן (הצפרדע) צריך לירות כדורים כך שמצטמצם השורה של הכדורים. השחקן מקבל צבע כדור לירות. כאשר יש 3 או יותר כדורים אחד אחרי השני בשורה מאותו הצבע אז הם נעלמים. כל צבע מתנהגת בצורה שונה שיפורט בהמשך. המטרה היא למקסם את הניקוד שלנו עד סוף המשחק.

המשחק מורכב מ:

- שורה/מערך של כדורים עליו השחקן
- המודל של המשחק (הסתברויות)
- כמות צעדים עד סוף משחק
- לצמצם

הבעיה

כקלט תקבלו את המשחק כאובייקט. המשחק מאותחל כך שמקבל את זה כקלט; זוג סדור שמורכב ממספר צעדים עד סוף משחק, רשימה של השורה של כדורים בו המשחק מאותחל בו, מילון של ערכים שמתארת את ההסתברויות והתפלגויות של המשחק וערך בוליאני בשביל לדבג. כל הסתברות פה היא שונה לפי צבעי הכדורים המדוברים. כדומה למטלה הקודמת אנחנו יכולים לנסות לכל מקום בשורה אומנם זה תלוי בהסתברות להצלחה לצבע אחרת הוא יבחר כל מקום אחר ברשימה (כולל לא להוסיף לרשימה) בהסתברות שווה. הצבע שמופיע לכל צעד היא תלוי בהתפלגות הצבעים שנתון (לדוגמה אדום 0.3 צהוב 0.5 ירוק 0.2). לצמצום יש גם הסתברות לקרות ורק כאשר הוכנס כדור לרצף שלו כלומר יכול להיות רשימה מאותחלת עם רצפים גדולים מ-3 ויכול להיות שהם לא יצטמצמו אפילו כשיש כדור שהוכנס ברצף, זה בודאות לא יצטמצם אם לא הוכנס כדור ברצף, אבל יכול להיות צמצום נגרר מצבעים שונים במיקום (דוגמה יהיה מפורט בהמשך). הניקוד לצמצום יהיה כך שיש ניקוד מינימלי של רצף מינימלי של 3 וכל כדור נוסף יכול להוסיף עוד לניקוד. בסוף המשחק אם צומצם כל השורה אז מקבלים פרס ענק, אם לא צומצם כל המשחק אז מקבלים עונש על כל כדור שנשאר לפי הצבע שלו.

זאת אומרת מטרת המשחק היא אך ורק לצמצם את השורה ולמקסם את הניקוד לפי המודל המתקבל. המשחק יכול להיגמר ב-2 אופנים:

- הצלחה: השחקן צמצם את השורה.
- כשלון: השחקן לא הצליח לצמצם את השורה תחת הכמות צעדים הנתונה.

הערה: המשחק לא נגמר ברגע שצומצם השורה רק כאשר הגענו לכמות המהלכים שהוגדר מראש.

תיאור המשימה

לתרגיל זה מצורף את הקוד שממש את המשחק ואת קובץ הבדיקה. עליכם לממש את המחלקה המממש את קונטרולר של המשחק כך שניתן יהיה לפתור את המשחק ולמקסם את הניקוד עם המודל הנתון.

קובץ ex2.py המצורף מכיל את חתימות הפונקציות שעליכם לממש. (ניתן כמובן גם לממש פונקציות נוספות):

1. פונקציית `choose_next_action`, יודע לתת את הפעולה הבאה לפי הפוליסייה שאותו הגדרתם באתחול הקונטרולר (כלומר מחזיר את הסקלר שמייצג את האינדקס לאיפה ברשימה של השורה שעליכם לצמצם את הכדור הנוכחי שניתן לזריקה).

אין לשנות את חתימות הפונקציות כלל!

אין להשתמש בפונקציה `submit_next_action` במשחק המקורי!

ייצוג קלט

הקלט שמתאר את הבעיה

לדוגמא הקריאה:

```
game = zuma.create_zuma_game((20, [1, 2, 3, 3, 3, 4, 2, 1, 2, 3, 4, 4],
example, debug_mode))
```

```
example = {
    'chosen_action_prob': {1: 0.6, 2: 0.7, 3: 0.5, 4: 0.9},
    # 'chosen_action_prob': {1: 1, 2: 1, 3: 1, 4: 1},
    'next_color_dist': {1: 0.1, 2: 0.6, 3: 0.15, 4: 0.15},
    # 'next_color_dist': {1: 0.25, 2: 0.25, 3: 0.25, 4: 0.25},
    'color_pop_prob': {1: 0.6, 2: 0.7, 3: 0.4, 4: 0.9},
    'color_pop_reward': {'3_pop': {1: 3, 2: 1, 3: 2, 4: 2},
                        'extra_pop': {1: 1, 2: 2, 3: 3, 4: 1}},
    'color_not_finished_punishment': {1: 2, 2: 3, 3: 5, 4: 1},
    'finished_reward': 150,
    'seed': 42}
```

כאשר:

- האיבר הראשון בזוג הסדור הוא מספר הצעדים
- האיבר השני בזוג הסדור הוא רשימה מאותחלת של השורה כאשר:
 - 1 - מסמל כדור אדום
 - 2 - מסמל כדור כחול
 - 3 - מסמל כדור ירוק
 - 4 - מסמל כדור צהוב
- האיבר השלישי בזוג הסדור הוא מילון של ערכים להתפלגויות כאשר:
 - הסתברות הצלחה של כדור להיכנס למקום שבחרתם, במילון הזה המפתח הוא הצבע והערך הוא ההסתברות הצלחה של כדור מאותו הצבע - `chosen_action_prob`
 - התפלגות צבעים לקבלת זריקה, במילון הזה המפתח הוא הצבע והערך הוא ההסתברות שכדור הבא לזריקה יהיה בצבע הנ"ל, זה צריך להסתכם ל-1 - `next_color_dist`

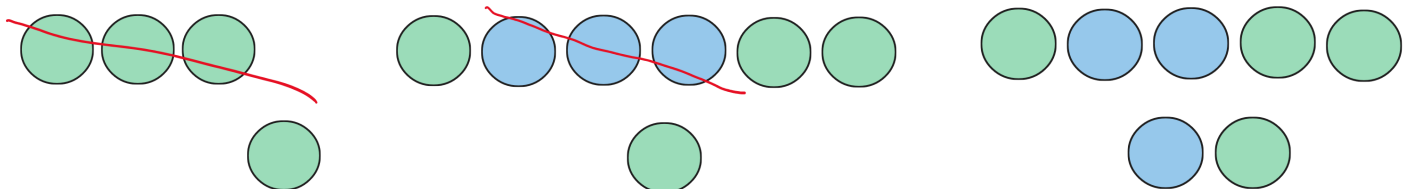
- הסתברות צמצום של רצף כדורים מצבע מסוים אחרי הכנסה, במילון הזה המפתח הוא הצבע והערך הוא ההסתברות צמצום רצף שאחרי שהוכנס כדור לרשימה מאותו הצבע ליד רצף מאותו הצבע - `color_pop_prob`
- תגמולי צמצום, במילון הזה קיים שני מילונים, במילון הראשון עם מפתח `pop_3` מתואר לכל צבע מה תגמול לצמצום מינימלי (רק 3 כדורים) כאשר המפתח זה הצבע והערך זה התגמול, במילון השני עם מפתח `extra_pop` מתואר לכל צבע את התגמול הבנוס לכל כדור נוסף ברצף מעל 3 כאשר המפתח הוא הצבע והערך הוא תגמול המתווסף לכל כדור נוסף. - `color_pop_reward`
- עונשי סוף משחק לכל כדור נשאר מצבע מסוים, במילון הזה המפתח הוא צבע והערך הוא העונש להביא על כל כדור מאותו הצבע. - `color_not_finished_punishment`
- תגמול סוף משחק ניתנת רק כאשר הסתיים המשחק עם שורה ריקה (כלומר אחרי כמות צעדים שהוגדרה השורה ריקה) זה מיוצג במספר - `finished_reward`
- מספר הנותן לשחזר תוצאות אקראיות, אם תרוצו את אותו מספר כל הפעולות האקראיות יפעלו בול אותו הדבר בהינתן אותו קלט. - `seed`
- האיבר הרביעי הוא משתנה בוליאני של להדליק הדפסת דיבאג בשביל לנוחיותכם, זה מתאר את התוצאה של ההגרלה לכל פעולה שקיימת לה אקראיות.

אופן וחוקיות התזוזה

לרשותכם $n+2$ פעולות כאשר n מייצג אורך השורה:

- יש $n+1$ פעולות של להכניס את הכדור המתקבל בין כל שני כדורים בשורה או בקצוות השורה
- יש פעולה אחת של לדלג על הכדור המתקבל לעבור לכדור הבא (נזרק ולא משמש יותר)

כל פעולה יכולה לגרום לשורה להצטמצם ויכולה לגרום לתגובת שרשרת של צמצום, כלומר אם אחרי שהכנסנו כדור ונוצר 3 או יותר ברצף מאותו הצבע נוציא את הרצף ואז נצטרך לעשות את הבדיקה שוב פעם כי יכולה להיווצר מצב שבו קיים רצף נוסף לאחר הוצאת רצף. לדוגמה מצב משחק כזה:



נכניס את הכדור הכחול ככה שיווצר רצף של 3 כדורים כחולים, הכדורים מצטמצמים ואז אחר בדיקה נוספת קיימת רצף חדש של 3 כדורים ירוקים ומצטמצמים אותם גם, לכן לאחר פעולה אחת צמצמנו את השורה. שימו לב, שהמצב ההתחלתי יכול להתחיל עם רצפים גדולים מ2.

הערה: כדורים מצטמצמים רק כאשר נזרק כדור ליד/בתוך הרצף מאותו הצבע (שהיה קיים או שנוצר מהזריקה) ואחרי צמצום הראשון בזריקה אם קיים עוד רצפים באיזור הזריקה (בודקים את אינדקס של תחילת הרצף שצומצם).

דגשים

- שימו לב שכדי לקבל מידע על המשחק יש להשתמש בפונקציות `get` ואין לכם רשות להשתמש לשנות או להשתמש בשאר הפונקציות או איברים בקובץ `zuma`, הקובץ בבדיקה לא תרוץ בDEBUG אז `show_history` לא יעבוד לכם בזמן בדיקה, ומותר לכם להשתמש ב `submit_chosen_action` רק על משחק מעותק ולא פוינטר על המשחק האמיתי (בפיתוח כל משתנה הוא אובייקט ולכן גם פוינטר אז תצטרכו לעשות `deepcopy` כדי לוודא שלא לשבש לכם תוצאות).
- הפונקציות שבתוך `game` שמותר לכם להשתמש זה `get_ball` (מחזיר את הכדור הנוכחי או מגריל כדור לפי התפלגות), `get_current_state` (מחזיר לך זוג סדור כאשר האיבר הראשון הוא השורה הנוכחית, האיבר השני הכדור הנוכחי, האיבר השלישי זה כמות הצעדים שבוצע עד כה, האיבר הרביעי זה כמות

- הצעדים שהוקצה למשחק עד סיומו), `get_current_reward` (מחזיר את התגמול שהצטבר עד כה), `get_model` (מחזיר את מודל המשחק, כלומר את ההסתברויות והתפלגויות של המשחק).
- שימו לב שאתם לא יכולים לשנות את המשחק המקורי ואת הייצוג שלו, אבל תוכלו לעשות מה שתמצאו בעתק המשחק ולייצג מצבים ופעולות באיזו צורה שבא לכם (תצטרכו כנראה להוסיף קוד כדי להתאים).
- פונקציה במשחק המעוות מותר לכם להשתמש `submit_next_action` שיבצע את כל החישובים והשלכות למשחק מקלט של מספר שהוא מתאר את האינדקס בו תרצו להכניס את הכדור הנוכחי.
- אמנם ייצוג המצב הוא בחירה שלכם אולם יש להקפיד על ייצוג הפעולות בדיוק כפי שכתוב – אותיות גדולות/קטנות, קווים תחתונים, ייצוג הפעולות נכון וכו'. הבדיקה אוטומטית, ולכן אם תטעו כאן תקבלו ציון נמוך מאד, וחבל.

בדיקת התרגיל

התרגיל המוגש ייבדק באופן אוטומטי, ולכן חשוב להקפיד על שמות מדויקים של קבצים, מחלקות ופעולות. הבדיקה תיבדק על אוסף קלטים בגדלים שונים ולכן חשוב שחישוב על מדיניות שלכם תהיה בזמן סביר. לאחר מכן הפתרון ייבדק צעד אחר צעד על מנת לוודא שהפתרון אכן תקין.

הקובץ `check.py` המצורף מריץ את המשחק עם הקלט לדוגמה ומחזיר את התגמול הסופי של המשחק והתוצאה הסופית של השורה. קובץ זה לא מבצע בדיקת נכונות של הפתרון המוחזר, ולכן זוהי אחריות שלכם לוודא שהפתרונות שלכם נכונים.

הרצת הקובץ באמצעות הפקודה:

```
python3 check.py
```

הערה: לא יהיה הרצות קוד על שרת מצידי, אם תצטרכו להריץ תעשו עצמאית דרך [הלינק](#), זה מיועד לווינדוס אבל תוכלו למצוא איך להתחבר עם לינוקס, או תריצו עצמאית. לא יהיה תוצאות השוואה כמו במטלה הקודמת, הצפי זה למקסם תגמול בזמן היעיל סביר.

הסבר קצר על הקוד המצורף

הקוד מורכב מחמישה קבצי פייתון:

1. `Ex2.py` – קובץ שבו נעשית העבודה העיקרית שלכם, והקובץ היחיד שעליכם לשנות. אמור להכיל את ה-`class` של `controller` המכיל את כל הפונקציות כפי שמתואר בסעיף "תיאור משימה". הקובץ כולל את חתימות של הפונקציות שעליכם לממש. **התרגיל יבדק עם קובץ `ex2.py` שלכם, ושאר הקבצים כפי שהם מופיעים במצבם המקורי ולכן אין טעם לשנות קבצים אחרים** (למעט הבעיות השונות שנבדוק עליהם את הקוד).
2. `check.py` – קובץ המכיל פונקציות מעטפת המנסות לפתור את הבעיה, ומכיל בעיה קטנה לדוגמה שאפשר לפתור. זה הקובץ שעליכם להריץ לבדיקת הפתרון שלכם.
3. `zuma.py` – מכיל את אלגוריתמי המשחק ואופן הפעולה שבה המשחק מתנהגת.

הגשה

- תאריך ההגשה עד ה-26.12.24
- הגשה ביחידים בלבד
- את הת.ז. של המגיש יש לרשום במשתנה `id` בקובץ `ex2.py` וכן לצרף קובץ `details.txt` המכיל את שם ות.ז של המגיש.
- שאלות על התרגיל יש לשאול בפורום שאלות ותשובות יעודי ב"למדה". שאלות יענו אחת ליומיים - שלושה במרוכז.
- הגשת התרגיל תהיה דרך מערכת ה-`submit`. יש להגיש רק את הקובץ `ex2.py` וקובץ `details.txt`. אין להגיש קבצי עזר המצורפים לתרגיל. אין להגיש קובץ בפורמט אחר (לדוגמה `zip` או `rar`). התוצאות



שהשרת מחזיר לכם לא רלוונטי,
לכן אין מה להתרגש מהציון 0 כי זה לא הציון העדכני ולא תהיה פלט לקוד.