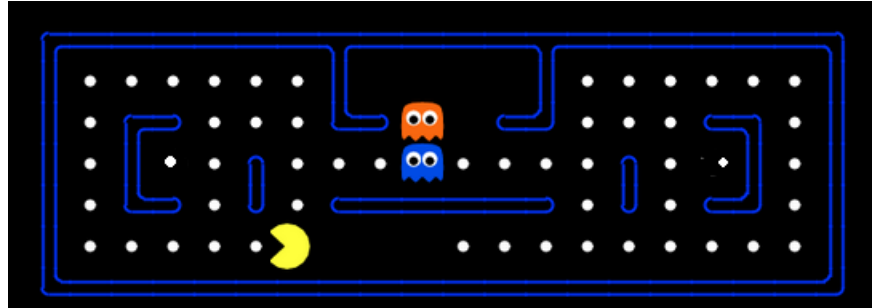


תרגיל בית 1 - פתרון בעיות על ידי חיפוש

מבוא

מטרת התרגיל היא להתנסות בפתרון אחת הגרסאות של משחק pacman בעזרת שיטות חיפוש שונות, אותן למדנו בקורס.



עקרון המשחק הוא ש-pacman (הדמות הצהובה) צריך "לאכול" את כל הגלולות (הנקודות הלבנות בתמונה למעלה), מבלי להיאכל על ידי הרוחות רפאים שיש על הלוח. הלוח המשחק מורכב מ- $N \times M$ משבצות ומכיל 5 סוגים של אובייקטים: קיר, שחקן (ה-pacman), רוחות רפאים, גלולות ורעל. **רעל משמש כמו "נפילה" מהלוח**

הבעיה

כקלט תקבלו את המצב ההתחלתי של המשחק. זוהי מטריצה (game) בגודל $N \times M$ שמייצגת את הלוח המשחק, הכניסות במטריצה מסמנות את המצב ההתחלתי של האובייקטים בו. השחקן יכול לזוז ימינה, שמאלה, למעלה ולמטה, בתנאי שבמשבצת אליה הוא זז אין קיר.

כמו שהוזכר, מטרת המשחק היא להוביל את השחקן על גבי הלוח המשחק, כך שיאכל את כל הגלולות, מבלי להיאכל על ידי אף אחת מהרוחות רפאים (לכל היותר 4 במשחק) הנעות על גבי הלוח, ומבלי לדרוח על רעל. המשחק יכול להגמר ב-2 אופנים:

- **הצלחה:** השחקן אכל את כל הגלולות
- **כשלון:** אחת המפלצות "אוכלת" את – השחקן יכול להגרם על ידי כניסה של השחקן למשבצת בה יש רוח רפאים, כניסה של רוח רפאים אל המשבצת של השחקן, או כניסה של השחקן לתא עם רעל

תיאור המשימה

לתרגיל זה מצורף קוד הממש אלגוריתמי חיפוש שלמדנו. עליכם לממש את המחלקה המייצגת משחק Pacman כך שניתן יהיה לפתור משחק נתון על ידי שימוש בקוד החיפוש המצורף.

עליכם לממש לפחות את הפונקציות הבאות (ניתן כמובן גם לממש פונקציות נוספות):

1. פונקציית actions המקבלת מצב ומחזירה tuple הכולל את כל הפעולות המותרות מאותו המצב
2. פונקציית result המקבלת מצב ופעולה ומחזירה מצב חדש הנוצר כתוצאה מהפעלת פעולה נתונה במצב נתון
3. פונקציית goal_test המקבלת מצב ומחזירה true אם המצב הנתון הוא מצב מטרה (מה שמוגדר כהצלחה בסעיף קודם), ו-false אחרת

4. פונקציית היוריסטיקה (h) המקבלת מצב ומחזירה את העלות המשוערת להגעה מהמצב הנתון למטרה.

שימו לב כי טיב ההיוריסטיקה ישפיע על הביצועים של אלגוריתמי החיפוש

5. עליכם גם לכתוב את מספרי תעודת זהות שלכם במשתנה ids מחוץ למחלקה

חובה להשתמש בקובץ ex1.py המצורף המכיל את חתימות הפונקציות שיש לממש.

ייצוג קלט

הקלט שמתאר את הבעיה מועבר לפונקציה

```
create_pacman_problem(Game)
```

כאשר:

- 99 – קיר
- 88 – שחקן נאכל ע"י רוח רפאים
- 77 – רעל
- 71 – רעל וגלולה
- 66 – שחקן (פאקמן)
- 51 – רוח רפאים **אדומה** במשבצת עם גלולה
- 50 – רוח רפאים **אדומה** במשבצת ללא גלולה
- 41 – רוח רפאים **ירוקה** במשבצת עם גלולה
- 40 – רוח רפאים **ירוקה** במשבצת ללא גלולה
- 31 – רוח רפאים **צהובה** במשבצת עם גלולה
- 30 – רוח רפאים **צהובה** במשבצת ללא גלולה
- 21 – רוח רפאים **כחולה** במשבצת עם גלולה
- 20 – רוח רפאים **כחולה** במשבצת ללא גלולה
- 11 – משבצת רגילה עם גלולה
- 10 – משבצת ריקה

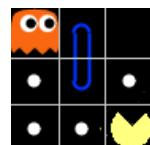
לדוגמא הקריאה:

```
create_pacman_problem (
```

```
(51, 99, 10),
```

```
(11, 99, 11),
```

```
(11, 11, 66))
```



מתארת את המשחק הבא:

3 על 3, ונמצאים בו 2 קירות, 4 משבצות אסימטריות, משבצת ריקה, רוח רפאים אחת, והשחקן. ניתן להניח את תקינות הקלט

אופן וחוקיות התזוזה

לרשותכם 4 פעולות של השחקן:

- תזוזה ימינה – מיצוגת ע"י מחרוזת "R"
- תזוזה למטה – מיצוגת ע"י מחרוזת "D"
- תזוזה שמאלה – מיצוגת ע"י מחרוזת "L"
- תזוזה למעלה – מיצוגת ע"י מחרוזת "U"

כל פועלה כזאת גורמת לפאקמן לזוז משבצת אחת בכיוון הרצוי, חוץ מתזוזה למשבצת הקיר. תזוזה כזאת מותרת, אבל לא תשנה את מיקום השחקן. חוץ מזה, אחרי זה גם רוחות הרפאים זזות. תזוזות רוחות הרפאים מתבצעת לפי החוקיות הבאה:

- כל רוח זזה משבצת אחת לאחד מ-4 הכיוונים (למעלה, למטה, ימינה או שמאלה)
 - סדר התזוזה: שחקן, אחריו רוח אדומה, אחרי זה כחולה, צהובה ולבסוף ירוקה
 - כל רוח תזוז לכיוון הכי מקרב אותה לשחקן לפי מרחק מנהטן
 - אם תזוזת הרוח ל-2 כיוונים מקרבת אותה לשחקן באותה מידה, אז סדר העדיפויות לתזוזה צריך להיות: ימינה, למטה, שמאלה ולבסוף למעלה.
 - רוח לא יכולה להישאר במקום, אלא אם כן אין לה משבצת שאליה היא יכולה לעבור.
 - רוח שדרכה על הרעל – נעלמת מהמשחק, כמו כן גם הרעל נעלם מהמשבצת הזאת. למחברי תרגיל הבית אין הבסר סביר איך רעל יכול להשפיע על רוח רפאים, אבל אלה הכללים.
 - רוח לא יכולה לעבור למשבצת שנמצאת בה רוח אחרת או קיר
- שימו לב, את תנועת הרוחות יש לחשב באופן הזה בלבד ולא על פי כל אלגוריתם אחר. פתרונות שיוחזרו עבור לוחות בהן הרוחות נעו באופן אחר מהמתואר לעיל ייכשלו בבדיקת הבודק האוטומטי.
- הערה: מצב כישלון במשחק (רוח אכלה את השחקן, או שחקן שנכנס למשבצת עם רעל) אין להמשיך לפתח את המצבים על הענף הזה

דגשים

- שימו לב שכדי לקבל את המצב מתוך קודקוד החיפוש (node), ניתן לגשת ל-node.state
- אתם יכולים לייצג מצב של הבעיה איך שאתם רוצים, אבל שימו לב שהקוד של החיפוש דורש שהמצב יהיה hashable, ולכן לא ניתן להשתמש ברשימה או במילון (או ב-tuple המכיל רשימה או מילון) על מנת ליצג מצב. ניתן לממש מחלקה המתארת מצב ויש לה פונקצית hash או להשתמש ב-tuple
- 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.

נמדדים על כמות הפעולות

בדיקת התרגיל

התרגיל המוגש ייבדק באופן אוטומטי, ולכן חשוב להקפיד על שמות מדויקים של קבצים, מחלקות ופעולות. הבדיקה האוטומטית תשתמש באלגוריתמי חיפוש שונים ($GBFS$ ו- A^*) על מנת לפתור את אוסף הקלטים בגדלים שונים. לאחר מכן הפתרון ייבדק צעד אחר צעד על מנת לוודא שהפתרון אכן תקין.

הקובץ `check.py` המצורף מריץ כל את אלגוריתמי החיפוש שנבדוק על מספר קלטים. קובץ זה לא מבצע בדיקת נכונות של הפתרון המוחזר, ולכן זוהי אחריות שלכם לוודא שהפתרונות שלכם נכונים.

שימו לב! הביצועים, וכמובן גם הציון, ינתן רק על סמך הריצות של $GBFS$ בלבד. כל שאר האלגוריתמים נמצאים בקוד בשביל התנסות בלבד.

הסבר קצר על הקוד המצורף

הקוד מורכב מחמישה קבצי פייתון:

1. `ex1.py` – קובץ שבו נעשית העבודה העיקרית שלכם, והקובץ היחיד שעליכם לשנות. אמור להכיל את ה-`class PacmanProblem` המכיל את כל הפונקציות כפי שמתואר בסעיף "תיאור משימה". הקובץ כולל את חתימות של הפונקציות שעליכם לממש. **התרגיל יבדק עם קובץ `ex1.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 לדוגמא שאפשר לפתור. זה הקובץ שעליכם להריץ לבדיקת הפתרון שלכם.
3. `search.py` – מכיל את אלגוריתמי החיפוש, ומכיל את מחלקת `node`.
4. שאר הקבצים – קבצי עזר

הגשה, בדיקת התרגיל וציונים

- הגשה בזוגות או יחידים בלבד
- מספיק שאחד מבני הזוג יגיש אין צורך ששני בני הזוג יגישו את התרגיל
- את הת.ז. של המגשים יש לרשום במשתנה `ids` בקובץ `ex1.py` (בתור `strings` בתוך רשימה קיימת. אם המגיש יחיד, אז רשימה צריכה להכיל `string` בודד)
- יש להגיש רק את הקובץ `ex1.py`. אין להגיש קבצי עזר המצורפים לתרגיל. אין להגיש קובץ בפורמט אחר (לדוגמא `zip` או `rar`)
- חלק מהציון ינתן על עמידה בדרישות הבסיס, וחלק ינתן על בסיס תחרות בין המגשים
- התחרות תתבצע על ביצועים של אלגוריתם $GBFS$ על מגוון קלטים. על האלגוריתם להוציא התכנית הקצרה ככל ניתן תוך 60 שניות

