# תרגיל בית 1- פתרון בעיות על ידי חיפוש

## מבוא

מטרת התרגיל היא להתנסות בפתרון אחת הגרסאות של משחק סוקובן (sokoban) בעזרת שיטות חיפוש שונות, אותן למדנו בקורס.

עקרון המשחק הוא שהשחקן חייב לדחוף את כל האובייקטים (נקרא להם ארגזים) ליעדים שונים במבוך תוך שימוש במספר מינימלי של תזוזות. ניתן להתנסות במשחק דומה מאוד באופן ידני בקישור הזה.

## הבעיה

כקלט תקבלו את המצב ההתחלתי של המשחק. זוהי מטריצה (game) בגודל NxM שמייצגת את לוח המשחק, הכניסות במטריצה מסמנות את המצב ההתחלתי של האוביקטים בו. השחקן יכול לזוז ימינה, שמאלה, למעלה ולמטה, בתנאי שבמשבצת אליה הוא זז אין קיר. כמו כן, השחקן יכול לדחוף את הארגזים ביחיד עם התזוזה שלו. ניתן לדחוף רק ארגז אחד בכל פעם, ואי אפשר למשוך אותם.ך

לוח המשחק מורכב מ – NxM משבצות ומכיל 5 סוגים של אוביקטים: קיר, שחקן ,ארגזים, יעדים וקרח. על תפקיד של כל אחד נפרט בהמשך.

כמו שהוזכר, מטרת המשחק היא לשים את כל הארגזים בכל היעדים ע"י דחיפות. **אין שיוך של ארגז ספציפי ליעד ספציפי.** זאת אומרת, אפשר לשים כל ארגז בכל יעד.

כפלט של כל התוכנית כולה תקבלו תוכנית (רצף של פעולות), או פלט מיוחד שאומר שהבעיה אינה פטירה (ראה סעיף "פלט")

## תיאור המשימה

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

עליכם לממש לפחות את הפונקציות הבאות (ניתן כמובן גם למשש 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 כמטרה 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.

**<u>למען הסר ספק: קוד שעושה את החיפוש מומש על ידינו, עליכם רק למדל את העולם בצורה טובה!</u>**

## ייצוג קלט

הקלט שמתאר את הבעיה מועבר לפונקציה

create_sokoban_problem(Game)

הוא tuple של tuples שמכיל את המפה, כאשר משמעות של ערכים של תוכן התאים היא:

- 99 – קיר
- 10 – משבצת ריקה
- 15 – ארגז
- 17 – שחקן
- 20 – יעד
- 25 – יעד עם ארגז עליו
- 27 – יעד ושחקן עליו
- 30 – קרח
- 35 – קרח עם ארגז עליו
- 37 – קרח ושחקן עליו

יש להניח שבגבולות המפה יש קירות.

## 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 לשחקן לזוז משבצת אחת בכיוון הרצוי.

- תזוזה למשבצת ריקה תעביר את השחקן אליה
- תזוזה למשבצת שיש בה קיר מותרת, אבל לא תשנה את מצב המשחק.
- תוצאה של תזוזה למשבצת שיש בה ארגז תלויה במה יש במשבצת הבאה בכיוון התזוזה
  - אם משבצת הבאה ריקה או יעד שאין עליו ארגז – הארגז זז למשבצת הריקה, והשחקן זז למשבצת שהיה בא ארגז
  - אם המשבצת הבאה תפוסה ע"י ארגז או קיר – אין שינוי במצב המשחק (אפשר להניח שהשחקן חלש מדי ולא יכול להזיז 2 ארגזים ביחד)
  - אם המשבצת הבאה היא קרח ללא ארגז – אז הארגז מחליק עליו בכיוון התנועה הראשונית עד המשבצת הראשונה שהיא לא קרח (ללא ארגז). אם המשבצת הראשונה שהיא לא קרח ריקה - הארגז נעמד עליה. אחרת הארגז נעמד במשבצת האחרונה של קרח. ניתן לראות כאן דוגמא לבעיה עם קרח
- תזוזה למשבצת של קרח – גורמת לשחקן להחליק בדיוק באותה צורה כמו הארגז

## פלט

יש 3 סוגי פלט שניתן לקבל מהתוכנית –

- (-3, -3, None): יש שגיאה בקוד
- (-2, -2, None): הקוד לא מצא פתרון תוך זמן נתון
- תוכנית המורכת מפעולות: הריצה הסתיימה בהצלחה

אם אתם רוצים לקבל הודעת שגיאה תוך כדי ריצה לצורך דיבוג – בקובץ check.py תורידו את ה-try (שורות 19,21,22)

## 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.
- **תקראו את הערות בקוד, יש שם הרבה מידע על השימוש בפונקציות**

## בדיקת התרגיל

התרגיל המוגש ייבדק באופן אוטומטי, ולכן חשוב להקפיד על שמות מדויקים של קבצים, מחלקות ופעולות.
הבדיקה האוטומטית תשתמש באלגוריתמי חיפוש שונים (A* וGBFS) 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.

**<u>שימו לב!</u>** 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 של SokobanProblem 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** את התרגיל, אין צורך ששני בני הזוג יגישו.
- את הת.ז. של המגישים 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)
- 80% מהציון ינתן על עמידה בדרישות הבסיס, ו20% ינתן על בסיס תחרות בין המגישים
- התחרות תתבצע על ביצועים של אלגוריתם GBFS על מגוון קלטים. על האלגוריתם להוציא **התכנית הקצרה ככל ניתן** תוך 60 שניות

- הנוסחא לציון על התחרות היא על כל קלט $20c^*/c$ כש-$c^*$ זה אורך הפתרון הכי טוב שנמצא, c זה אורך הפתרון שלכם, והציון הסופי על התחרות הוא ממוצא של ציונים על פני קלטים בודדים.