**יישומי בינה מלאכותית**

**סמסטר חורף תש"פ**

**מטלה תכנותית 1**תאריך הגשה: 10.12.19 23:55

# הנחיות

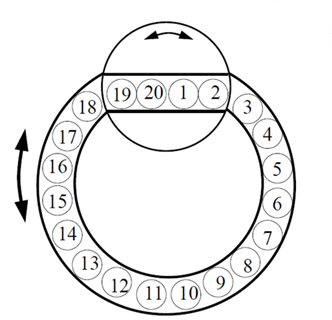
* שאלות בנושא מטלה זו יש לשאול דרך המודל, בפורום "מטלה תכנותית 1".
* יש להגיש קובץ **zip** בשם **id1\_id2.zip** ובו הקבצים הבאים **בלבד:**
  + IDs.java
  + AStarSearch.java
  + BreadthFirstSearch.java
  + PureHeuristicSearch.java
  + UniformCostSearch.java
  + TilePuzzleHeuristic.java
* ההגשה חייבת להתבצע ללא קינון בתוך תיקייה וללא הגשת הפרויקט כולו. דוגמת הגשה מצורפת למטלה.
* העבודה להגשה בזוגות בלבד אלא אם כן המגישים קיבלו אישור להגשה שאינה בזוגות.
* לפני שניגשים לממש את המטלה מומלץ לעיין רבות בהסברים וכן בקוד הקיים.
* פתרון המטלה שתגישו ייבדק מול שאר ההגשות ע"י תוכנת העתקות.

**מי שימצא כי העתיק יכשל בקורס וכן יועבר לוועדת משמעת אוניברסיטאית**.

* הפרויקט נכתב וייבדק בשפת התכנות Java, גרסה JRE 1.8.
* מסמך זה בנוי באופן הבא: הקדמה, משימות 1-5, ולאחר מכן הסבר על מחלקות המטלה.

הקדמה

במטלה זו נפתור את בעיית ה-Weighted TopSpin. מלבד מספר שינויים שיתוארו בהמשך, בעיה זו הינה זהה לבעיית ה-TopSpin הסטנדרטית שנראית כך:



בתמונה מופיע המצב הסופי היחיד של הבעיה, אליו אנו שואפים להגיע. אותו אנו נתאר בעזרת מערך המספרים הבא:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 |

המשחק מורכב משני חלקים:

• טבעת חיצונית, המכילה 20 דסקיות הממוספרות מ-1 עד 20.

• טבעת פנימית, המכילה 4 מהדסקיות של הטבעת החיצונית.

קיימות שלוש פעולות שניתן לבצע:

* LEFT – הזזת הדסקיות בטבעת החיצונית במעגל כך שהפרמוטציה נשארת אבל מסתובבת שמאלה ביחידה אחת. המערך לאחר ביצוע פעולה זו:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 |

* RIGHT - הזזת הדסקיות בטבעת החיצונית במעגל כך שהפרמוטציה נשארת אבל מסתובבת ימינה ביחידה אחת. המערך לאחר ביצוע פעולה זו:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 |

* SWAP - סיבוב הטבעת הפנימית כך שסדר הדסקיות בה יתהפך (לדוגמא: בתמונה המצורפת ניתן לסובב את הטבעת הפנימית כך שסדר הדסקיות בה יתהפך מ-19,20,1,2 ל-2,1,20,19). המערך לאחר ביצוע פעולה זו:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 19 | 20 | 1 | 2 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 |

עד כה תיארנו את הבעיה הסטנדרטית. כעת נתאר את השינויים של גרסת ה-Weighted.

* בגרסת ה-Weighted של הבעיה קיימות 10 דסקיות בלבד (במקום 20) הממוספרות מ-0 ועד 9.
* המצב הסופי של הבעיה הינו מערך הנראה כך:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

כלומר, בתא במקום נמצאת דסקית הממוספרת ב-. במצב זה, המספרים 3,4,5,6 נמצאים בטבעת הקטנה.

* עלות ביצוע הפעולות LEFT ו-RIGHT הינה 0 ועלות ביצוע הפעולה SWAP הינה כסכום המספרים הנמצאים בטבעת הקטנה. לדוגמא, עלות ביצוע SWAP על המערך המופיע מעלה הינה 18 (3+4+5+6).

במטלה זו עליכם לממש 4 אלגוריתמי חיפוש (BFS, UCS, PHS, A-Star) ויוריסטיקה לבעיה.

מצורפים לפרויקט מופעים שונים של הבעיה. מופעים אלו נמצאים בתיקייה instances ובה בתיקיית topSpin10.

בעת בדיקת המטלות נריץ מופעים דומים ברמת הקושי (אך לא זהים), עליכם לעמוד בזמני ריצה שונים (יוגדרו בהמשך המטלה) וכן שהאלגוריתמים יפתרו את הבעיה בצורה נכונה בהתאם לכל אחד מהאלגוריתמים.

על מנת לממש את אלגוריתמי החיפוש במטלה זו, עליכם לממש את השיטות הבאות בלבד במחלקה המתאימה:

* initLists – שיטה המאתחלת את הרשימות open ו-closed.
* getOpen – בדוק האם קודקוד נתון נמצא ב-open. במידה וכן, החזר את הקודקוד המתאים מ-open. אחרת, null. **מומלץ** מאוד ששיטה זו תמומש באופן יעיל () כדי שהאלגוריתמים ירוצו מהר.
* isOpen – החזר "אמת" אם קודקוד נמצא ב-open. "שקר" אחרת.
* isClosed - החזר "אמת" אם קודקוד נמצא ב-closed. "שקר" אחרת.
* addToOpen – הוסף קודקוד נתון ל-open.
* addToClosed – הוסף קודקוד נתון ל-closed.
* openSize – החזר את גודל ה-open.
* getBest – החזר את הקודקוד הטוב ביותר, על פי אלגוריתם החיפוש.

\* שימו לב, משימה 5 מכילה בונוס עבור הסטודנטים שיפתרו אותה בצורה הטובה ביותר.

משימה 1

הזינו את מספרי תעודת הזהות של המגישים במחלקת IDs, במשתנים id1 ו-id2.

- כעת ניתן להריץ את השיטה task1 במחלקת Main ולראות כי מודפסים מספרי תעודות זהות המגישים. שימו לב, מניסיון העבר, חלק מהסטודנטים שוכחים לבצע את משימה זו או שוכחים להגיש אותה כחלק מפתרונם למטלה ולכן ציונם נפגע (וחבל).

משימה 2

ממשו את אלגוריתם החיפוש BFS על ידי מימוש השיטות במחלקת BreadthFirstSearch, כפי שהוגדר בהקדמה.

- כעת ניתן להריץ את השיטה task2 במחלקת Main. אנו מצפים לראות 25 מופעים של הבעיה נפתרים אחד אחר השני, כאשר כמות המהלכים המבוצעים בכל פתרון הינה מינימלית. עליכם לעמוד בזמן של דקה אחת להרצת כל אחד ממופעים אלו.

משימה 3

ממשו את אלגוריתם החיפוש UCS על ידי מימוש השיטות במחלקת UniformCostSearch, כפי שהוגדר בהקדמה.

- כעת ניתן להריץ את השיטה task3 במחלקת Main. אנו מצפים לראות 25 מופעים של הבעיה נפתרים אחד אחר השני, כאשר סכום מחירי כל המהלכים המבוצעים בכל פתרון הינו מינימלי. עליכם לעמוד בזמן של דקה אחת להרצת כל אחד ממופעים אלו.

משימה 4

* ממשו יוריסטיקה עבור המשחק Weighted TopSpin במחלקת TopSpinHeuristic בשיטה getHeuristic. שיטה זו מקבלת כקלט מצב במשחק ומחזירה הערכה עבור המחיר שיעלה להגיע ממצב זה לקודקוד מטרה. ניתן לממש שיטה זו כרצונכם אך שימו לב שהערכה זו **חייבת** להיות אדמיסיבילית.
* ממשו את אלגוריתם החיפוש PHS על ידי מימוש השיטות במחלקת PureHeuristicSearch, כפי שהוגדר בהקדמה.

- כעת ניתן להריץ את השיטה task4 במחלקת Main. אנו מצפים לראות 25 מופעים של הבעיה נפתרים אחד אחר השני. עליכם לעמוד בזמן של 10 שניות להרצת כל אחד ממופעים אלו.

משימה 5

ממשו את אלגוריתם החיפוש AStar על ידי מימוש השיטות במחלקת AStarSearch, כפי שהוגדר בהקדמה.

- כעת ניתן להריץ את השיטה task5 במחלקת Main. אנו מצפים לראות 25 מופעים של הבעיה נפתרים אחד אחר השני, כאשר סכום מחירי כל המהלכים המבוצעים בכל פתרון הינו מינימלי. עליכם לעמוד בזמן של 30 שניות להרצת כל אחד ממופעים אלו.

* **בונוס** - זוג הסטודנטים אשר עבורם זמן ריצת השיטה task5 יהיה הקצר ביותר יזכו ב-2 נקודות בונוס לציון הסופי בקורס (כל אחד).

מחלקות הפרויקט

ASearch – מחלקת חיפוש אבסטרקטי. ממנה יורשים כלל החיפושים שעליכם לממש בפרויקט ובה ממומש כבר החיפוש האבסטרקטי כפי שנלמד בכיתה.

ASearchNode – מחלקת קודקוד חיפוש אבסטרקטי. מחלקה זו מחזיקה במצביע למצב הקודקוד הקודם בעץ וכן המצב הנוכחי בבעיה. ממחלקה זו יורשת מחלקת BlindSearchNode, המשמשת עבור חיפושים מסוג Blind Search.

AStarSearch – מחלקת אלגוריתם החיפוש AStar, בה עליכם לממש את השיטות המוגדרות בהקדמה. מחלקה זו יורשת ממחלקת החיפוש האבסטרקטי ASearch.

BlindSearchNode – מחלקת קודקוד Blind Search. מחלקה זו יורשת ממחלקת ASearchNode ומחזיקה בערך ה-g של הקודקוד וכן בעומק הקודקוד בעץ החיפוש.

BreadthFirstSearch – מחלקת אלגוריתם החיפוש BFS, בה עליכם לממש את השיטות המוגדרות בהקדמה. מחלקה זו יורשת ממחלקת החיפוש האבסטרקטי ASearch.

HeuristicSearchNode – מחלקת קודקוד Heuristic Search. מחלקה זו יורשת ממחלקת BlindSearchNode ומחזיקה בערך ה-h של הקודקוד וכן בשיטת חישוב היוריסטיקה TopSpinPuzzleHeuristic.

IDs – מחלקה זו משמשת לקבלת מספרי תעודת הזהות של מגישי המטלה.

Main – המחלקה הראשית של הפרויקט. ממנה קוראים למשימות המוגדרות במטלה.

PureHeuristicSearch – מחלקת אלגוריתם החיפוש PHS, בה עליכם לממש את השיטות המוגדרות בהקדמה. מחלקה זו יורשת ממחלקת החיפוש האבסטרקטי ASearch.

TopSpinPuzzle – מחלקת המשחק Weighted TopSpin. מחלקה זו מגדירה את הבעיה שעלינו לפתור.

TopSpinPuzzleHeuristic – מחלקת היוריסטיקה עבור המשחק Weighted TopSpin. במחלקה זו תצטרכו לממש את היוריסטיקה המוגדרת המטלה.

TopSpinPuzzleMove – מחלקת "צעד" עבור המשחק Weighted TopSpin. מחלקה זו מתארת אופרטור במשחק.

TopSpinPuzzleState – מחלקת "מצב" עבור המשחק Weighted TopSpin. מחלקה זו מתארת מצב במשחק.

UniformCostSearch - מחלקת אלגוריתם החיפוש UCS, בה עליכם לממש את השיטות המוגדרות בהקדמה. מחלקה זו יורשת ממחלקת החיפוש האבסטרקטי ASearch.

בהצלחה!