



## הנדסת תוכנה – תרגיל בית 1

### דגשים להגשת המטלה

1. **תאריך הגשה: יום ראשון 21.05.2023 בשעה 23:59.**
2. **הגשה בזוגות בלבד!**
3. הקוד חייב להיכתב בהתאם למוסכמות כתיבת הקוד בקורס כולל תיעוד כנדרש. קוד שלא עומד בדרישות יגרור הורדת ניקוד. ניתן למצוא את קובץ מוסכמות הקידוד באתר הקורס תחת הלשונית "קבצי עזר".
4. ההגשה מתבצעת ב-Moodle באזור המיועד על ידי אחד מהשותפים, לאחר יצירת קבוצה.
5. ניתן להגיש את התרגיל לכל היותר עד 48 שעות לאחר מועד ההגשה ללא הורדת ניקוד. לאחר 48 שעות תיבת ההגשה תיסגר ולא יהיה ניתן להגיש את התרגיל כלל.
6. פורמט הגשת התרגיל נמצא בקובץ ההנחיות ב-Moodle. **כל חריגה מפורמט זה תגרור ציון אפס.**

### מטרת התרגיל

עבודה ראשונה עם מחלקות.

### הכנות טרם תחילת התרגיל

1. פתיחת פרויקט ג'אווה חדש.
2. הורדת קבצי התרגיל והעתקת כל קבצי הקוד אל תוך תיקיית ה-src.

### הוראות כלליות

1. מומלץ להריץ את התוכנית עם מספר קליטים שונים ולחשוב על מקרי קצה אפשריים.
2. מומלץ לחזור על התרגולים וההרצאות וכן להיעזר באינטרנט.
3. יש להשתמש בגרסה 9.0.4 של ג'אווה בעת פתרון התרגיל.
4. מומלץ להשתמש ב-Git במהלך כתיבת התרגיל.

### הוראות הגשה

1. יש למלא אחר הוראות ההגשה בהתאם למסמך הדרישות "הנחיות כלליות לפתרון והגשת תרגילי הבית" אשר מופיע באתר הקורס.
2. הגשה אלקטרונית בלבד דרך אתר הקורס ב-moodle. ההגשה תכלול קובץ ה-zip בלבד בפורמט `HW1_<id1>_<id2>` כאשר `<id1>` ו-`<id2>` הם תעודות הזהות של המגשים.
3. ההגשה מתבצעת על ידי אחד מהשותפים בתיבת ההגשה. על שני השותפים להיות חברים באותה קבוצה.
4. תרגיל בית שלא יוגש על פי הוראות ההגשה – **לא ייבדק.**
5. יש להקפיד על יושרת הכנת התרגיל וההגשה.
6. יש לוודא כי הקוד מתקמפל – קוד אשר לא יעבור הידור יקבל ציון אפס.



## משחק ה-Sliding Puzzle

בתרגיל בית זה תממשו משחק בשם Sliding Puzzle ותפתרו אותו על ידי אלגוריתם מתחום הבינה המלאכותית.

משחק ה-Sliding Puzzle הינו משחק לשחקן יחיד המשוחק על גבי לוח משחק.

לוח המשחק הינו לוח מלבני המחולק למשבצות ריבועיות שוות בגודלן. בכל משבצת ישנו אריח בעל מספר ייחודי, פרט למשבצת אחת ריקה. עבור לוח בגודל  $m \times n$ , המספרים שמופיעים על גבי האריחים הינם בטווח שבין 1 ל-  $m \cdot n - 1$ , כך שכל מספר מופיע בדיוק פעם אחת.

המשחק מתחיל כאשר האריחים מסודרים בסידור התחלתי נתון על גבי הלוח. בכל צעד במשחק ניתן להזיז אריח אחד אשר נמצא בסמוך למשבצת הריקה ולהעבירו אל המשבצת הריקה. את האריחים ניתן להזיז למעלה, למטה, ימינה ושמאלה אך לא ניתן להזיזם באלכסון. המטרה במשחק הינה לסדר את האריחים על גבי הלוח לפי סדר המספרים הרשומים עליהם, כאשר האריח בעל המספר 1 נמצא בפינה השמאלית העליונה והמשבצת הימנית התחתונה הינה ריקה.

להלן דוגמה של לוח בגודל  $3 \times 3$  במצב התחלתי כלשהו, ובמצב המטרה שלו:

1	2	3
4	5	6
7	8	

מצב מטרה

7	5	4
	3	2
8	1	6

מצב התחלתי

בדוגמה זו, מן המצב ההתחלתי ישנן שלוש פעולות אפשריות: להזיז את האריח שכתוב עליו 8 למעלה, להזיז את האריח שכתוב עליו 7 למטה או להזיז את האריח שכתוב עליו 3 שמאלה. שימו לב שלכל גודל של לוח ולכל מצב בו הלוח נמצא, ישנן לכל היותר ארבע פעולות הניתנות לביצוע.

## פתרון המשחק

ישנם מספר אלגוריתמים המסייעים במציאת פתרון למשחק – רצף של פעולות (חוקיות) אשר ביצוען אחת אחרי השנייה החל מן המצב ההתחלתי יוביל למצב מטרה. מרבית האלגוריתמים הינם אלגוריתמי חיפוש הנמצאים בשימוש בתחום הבינה המלאכותית, והם מבצעים חיפוש במרחב המצבים על ידי הרחבה של מצבים קודמים (עוד על כך מיד, וכן בקורס בינה מלאכותית אותו תיקחו בעתיד).

אלגוריתם החיפוש בו תשתמשו על מנת למצוא פתרון למשחק נקרא אלגוריתם Greedy Best First Search. על מנת לתאר את האלגוריתם, עלינו להגדיר תחילה מספר מושגים:

- מצב הינו "צילום" רגעי של הבעיה. במקרה של תרגיל זה, מצב מכיל מידע על לוח המשחק – המיקום של כל אריח.
- מצב התחלתי הינו המצב ממנו מתחילים את הבעיה.
- מרחב המצבים הוא אוסף כל המצבים הקיימים במשחק. במקרה של תרגיל זה, מרחב המצבים הינו אוסף כל הסידורים האפשריים של האריחים על גבי הלוח. שימו לב כי מרחב מצבים זה הוא (לרוב) עצום.
- קודקוד הינו עצם אשר מכיל מידע על מצב מסוים, על קודקוד האב שלו (הקודקוד אשר קדם לו ומכיל את המצב הקודם של המשחק) ועל הפעולה שבוצעה על מנת להגיע מן המצב הקודם אל המצב הנוכחי.
- ערך היוריסטי של מצב הינו מספר שלם המשערך את המרחק שנותר מן המצב הנוכחי אל מצב מטרה.



## הטכניון – מכון טכנולוגי לישראל הפקולטה למדעי הנדסה והחלטה הנדסת תוכנה אביב תשפ"ג



אלגוריתם החיפוש פועל באופן הבא:

תחילה, הוא מאתחל אוסף ריק ומכניס אליו את הקודקוד ההתחלתי – קודקוד המכיל את המצב ההתחלתי.

לאחר מכן הוא חוזר על הצעדים הבאים כל עוד האוסף אינו ריק:

1. הוצאת קודקוד אשר מכיל מצב בעל ערך היוריסטי מינימלי מבין כל הקודקודים שבאוסף.
  2. בדיקה האם המצב של הקודקוד שהוצא הינו מצב מטרה. במידה וכן, האלגוריתם מייצר מן הקודקוד את המסלול לפתרון ומסיים. ייצור הפתרון מתאפשר מאחר וכל קודקוד מכיל מידע על קודקוד האב שלו.
  3. אם המצב של הקודקוד אינו מצב מטרה, מתבצעת הרחבה לקודקוד: מעבר על כל המצבים שניתן להגיע אליהם מן המצב של הקודקוד הנוכחי, ייצור קודקוד לכל אחד מהם והכנסת הקודקודים אל האוסף.
- לבסוף, במידה ואוסף הקודקודים ריק אך לא הגענו למצב מטרה, האלגוריתם מחזיר כי אין פתרון – לא ניתן להגיע מן המצב ההתחלתי למצב מטרה.
- בנוסף לכל השלבים המתוארים, האלגוריתם מתחזק אוסף נוסף אשר שומר את כל המצבים בהם ביקר, על מנת למנוע מקרים בהם הוא יבקר פעמיים באותו מצב ובכך לחסוך בזמן הריצה ובזיכרון.

### מימוש המשחק

על מנת לממש את המשחק, תצטרכו ליצור מספר מחלקות. מחלקות אלו יתבססו אחת על השנייה, וביחד יהוו את מכלול המשחק.

### מחלקת Tile

מחלקה זו מייצגת אריח בודד. כפי שהוסבר, אריח מכיל את המספר אשר כתוב עליו. מספר זה לא יכול להשתנות לאחר יצירת האריח.

### מחלקת Board

מחלקה זו מייצגת את לוח המשחק.

במחלקה יש להגדיר בנאי המקבל מחרוזת שמייצגת את הלוח, ומאתחל את האריחים בהתאם. המחרוזת נתונה בפורמט שבו הלוח משוטח, וכל האריחים מופיעים זה לאחר זה בשורה אחת. במחרוזת, אריחים הנמצאים באותה השורה מופרדים זה מזה על ידי רווח, והשורות מופרדות זו מזו על ידי קו ניצב ('|'). בנוסף, המשבצת הריקה מיוצגת על ידי קו תחתון ('\_'). לדוגמה, הלוח שבתמונה הימנית מיוצג על ידי המחרוזת 7 5 4|\_ 3 2|8 1 6. ניתן להניח את תקינות המחרוזת.

### Enum Direction

Enum זה מייצג כיוון הזזה של אריח. כפי שהוסבר, ישנם ארבעה כיוונים אפשריים: מעלה (UP), מטה (DOWN), ימינה (RIGHT) ושמאלה (LEFT).

### מחלקת Action

מחלקה זו מייצגת פעולה בודדת אשר ניתן לבצע במשחק. בעת הגדרת המחלקה, חישבו מה מאפיין פעולה במשחק.

יש להגדיר במחלקה פעולה בשם toString שאינה מקבלת פרמטרים ומחזירה מחרוזת המייצגת את הפעולה בפורמט "Move <tile> <direction>" כאשר tile הינו המספר שכתוב על האריח אותו מזיזים ו-direction הינו הכיוון אליו מזיזים את האריח, רשום באותיות קטנות. בדוגמה הקודמת, הפעולה (האפשרית) של להזיז את האריח שכתוב עליו 7 למטה מיוצגת על ידי המחרוזת "Move 7 down".



## הטכניון – מכון טכנולוגי לישראל הפקולטה למדעי הנדסה והחלטה הנדסת תוכנה אביב תשפ"ג



### מחלקת State

מחלקה זו מייצגת מצב בודד מתוך מרחב המצבים. כפי שהוזכר, מצב מכיל את לוח המשחק.

יש להגדיר במחלקה את הפעולות הבאות:

- פעולה בוליאנית בשם isGoal אשר אינה מקבלת פרמטרים, בודקת האם המצב הנוכחי הינו מצב המטרה ומחזירה את התשובה.
- פעולה בשם actions אשר אינה מקבלת פרמטרים ומחזירה מערך של כל הפעולות אותן ניתן לבצע מן המצב הנוכחי. במערך **על הפעולות להופיע בסדר הבא**: הזזה למעלה, הזזה למטה, הזזה ימינה והזזה שמאלה. במידה ופעולה לא אפשרית אין לכלול אותה במערך, וכן אין להשאיר תאים ריקים במערך.
- פעולה בשם result אשר מקבלת פעולה (Action) ומחזירה את המצב החדש שמתקבל מביצוע הפעולה במצב הנוכחי.

### מחלקת Node

מחלקה זו מייצגת קודקוד בודד. כל קודקוד מכיל מידע על מצב, על קודקוד האב שלו ועל הפעולה שבוצעה על מנת להגיע מן המצב הקודם אל המצב הנוכחי.

יש להגדיר במחלקה את הפעולות הבאות:

- פעולה בשם expand אשר מבצעת הרחבה לקודקוד הנוכחי. הפעולה אינה מקבלת פרמטרים ומחזירה מערך של כל הקודקודים המתקבלים מן ההרחבה. אין להשאיר תאים ריקים במערך. חישבו כיצד להיעזר בפעולות המחלקה State על מנת לבצע את הרחבת הקודקוד.
- פעולה בשם heuristicValue אשר אינה מקבלת פרמטרים ומחזירה את הערך ההיוריסטי של המצב שמוכל בקודקוד.

### מחלקת Search

מחלקה זו מכילה את אלגוריתם החיפוש.

מאחר ומימוש האלגוריתם כולל בתוכו שימוש במחלקות ובפעולות שלא נלמדו בקורס (אך רובן יילמדו בהמשך) האלגוריתם מומש עבורכם.

הפעולה היחידה אותה תצטרכו לשנות במחלקה הינה פעולה בשם getRoot אשר מקבלת מחרוזת המייצגת את הלוח ההתחלתי, בונה קודקוד התחלתי מן הלוח ומחזירה אותו. שימו לב כי לקודקוד ההתחלתי אין קודקוד אב ואין לו פעולה שהביאה למצב ההתחלתי.

פרט לכך, **אין לשנות בשום דרך** את שאר הפעולות והתכונות המוגדרות במחלקה.

### קוד נוסף אותו קיבלתם

בנוסף לאלגוריתם החיפוש, הנכם מקבלים מימוש חלקי (ומינימלי) של חלק מן המחלקות שהוזכרו קודם לכן. במחלקות אלו מוגדרות פעולות שתפקידן לסייע לאלגוריתם החיפוש ולייעל אותו. **אין לשנות את הפעולות אותן קיבלתם.** עם זאת, אתם מוזמנים לנסות ולשחק עם הפעולות ולראות את ההשפעה שלהן על אלגוריתם החיפוש ועל זמן הריצה שלו, אך יש לדאוג להחזירן לאופן בו הן ניתנו לכם.

שימו לב כי קוד זה מכיל שימוש בתכונות שעדיין לא הוגדרו במחלקות. על כן, תוכלו להיעזר בקוד הנתון בעת החשיבה על התכונות שעליכם להגדיר בכל מחלקה.



## הטכניון – מכון טכנולוגי לישראל הפקולטה למדעי הנדסה והחלטה הנדסת תוכנה אביב תשפ"ג



### ערך היוריסטי

ערך היוריסטי של מצב הינו מספר שלם ואי-שלילי המקרב את המרחק של המצב ממצב מטרה – כמה פעולות (בקירוב) נצטרך לעשות על מנת להגיע מן המצב הנוכחי למצב מטרה.

דוגמה לערך היוריסטי פשוט עבור המשחק הוא מספר האריחים שאינם נמצאים במקומם. ערך זה מהווה קירוב למרחק ממצב המטרה, שכן יש להזיז לפחות פעם אחת כל אריח שלא נמצא במקומו על מנת להגיע לפתרון.

הערך ההיוריסטי מאפשר לאלגוריתם לתעדף בין מצבים שונים, שכן בכל שלב הוא בוחר באופן חמדני קודקוד שהמצב המוכל בו הוא בעל ערך היוריסטי מינימלי מבין כל שאר הקודקודים שבאוסף. כך הערכים ההיוריסטיים מסייעים "להנחות" את האלגוריתם לכיוון מצב מטרה.

### המלצות לסדר העבודה

התחילו בלודא כי הבנתם את התיאור של המשחק ואת כל ההגדרות המופיעות בתחילת הקובץ. ההגדרות של המושגים מכילות כמעט את כל מה שנדרש על מנת להגדיר את המחלקות השונות.

לאחר מכן, עברו על הקוד אותו סיפקנו לכם, ובדקו שאתם מבינים את הפעולות שמוגדרות שם (אין חובה להבין לעומק כל שורה וכל מחלקה שנמצאת בשימוש) וכיצד מימוש אלגוריתם החיפוש תואם להסבר של האלגוריתם. מרבית קטעי הקוד מכילים תיעוד, הן ברמת הפעולות והן ברמת שורות קוד בודדות, על מנת לסייע לכם להבין את הקוד ואת מה שעליכם להגדיר במחלקות השונות. היעזרו בקוד על מנת לקבל אינדיקציה לתכונות ולפעולות אותן עליכם להגדיר בחלק מן המחלקות. לדוגמה, במחלקה Search ישנה פעולה בשם extractSolution אשר מקבלת קודקוד ומפעילה עליו פעולות שונות, כדוגמת הפעולה `getAction`. חישבו מהי התכונה שפעולה זו מרמזת לכם להוסיף למחלקה `Node`.

בנוסף, מומלץ מאוד להשאיר את הערך ההיוריסטי לסוף ולממש אותו רק לאחר בניית כל המחלקות ומימוש כל שאר הפעולות. תוכלו להחזיר בפעולה את הערך 0 ולחזור אליה לאחר שסיימתם לכתוב את כל שאר הקוד. ישנם לוחות פשוטים הניתנים לפתרון ללא ערך היוריסטי, ולכן אנו ממליצים שתוודאו שלאחר החזרת 0 מן הפעולה הנכם מצליחים לפתור את אותם הלוחות. תוכלו גם להוסיף לוחות פשוטים נוספים, כגון לוחות בגודל  $2 \times 2$  על מנת לוודא ששאר הקוד פועל באופן תקין. זכרו להחזיר את הקוד של המחלקה `Main` להיות בדיוק כפי שקיבלתם אותו וללא שינויים כלשהם. לבסוף, אחרי שווידאתם כי הקוד תקין, עברו לחשוב על אופן המימוש של הערך ההיוריסטי.

### הנחיות לפתרון

- בעת פתרון התרגיל ניתן להגדיר מחלקות נוספות.
- יש לממש את כל המחלקות והפעולות שהוגדרו לאורך המסמך ולבצע בהן שימוש כחלק מהפתרון. ניתן (ואף מומלץ) להגדיר פעולות עזר נוספות.
- בכל מחלקה יש לכלול את התכונות המתאימות לה, ולספק בעבורן פעולות `get` ו-`set` **במידת הצורך בלבד**.
- על המחלקות להכיל בנאים. אין חובה שבנאי יקבל כפרמטרים את כל התכונות המוגדרות במחלקה (ואף זה אינו רצוי בעבור חלק מן המחלקות).
- יש לתעד את כל הפעולות והמחלקות אותן אתם מגדירים בעזרת שימוש ב-`JavaDoc` בהתאם לקובץ מוסכמות התיעוד אשר מופיע באתר הקורס. בנוסף, יש לתעד שורות קוד שעשויות להיות קשות להבנה.
- בעת פתרון התרגיל ניתן ואף מומלץ להגדיר קבועים ולא להשתמש במספרי קסם.
- שימו לב כי לא כל לוח ניתן לפתרון. ישנם מצבים התחלתיים אשר לא ניתן להגיע מהם למצב המטרה.



## הטכניון – מכון טכנולוגי לישראל הפקולטה למדעי הנדסה והחלטות הנדסת תוכנה אביב תשפ"ג



- יש ליצור כל מחלקה (וכן את ה-enum) בקובץ נפרד בתיקית ה-src ולהגדירה כפומבית.
- במהלך פתרון התרגיל, יש להקפיד על שמות משמעותיים למשתנים, לפעולות ולמחלקות.
- במהלך פתרון התרגיל, יש להקפיד על הרשאות הגישה השונות בהתאם לעקרונות שנלמדו בקורס.
- תרגיל זה עוסק בעבודה עם מספר מחלקות ועיקרו הוא קשר ההכלה. יש לפתור את התרגיל ללא שימוש בהורשה.
- אין להיעזר באף פעולה שלא אתם כתבתם, גם לא בפעולות שאינן דורשות ייבוא, פרט לפעולות של המחלקה Integer והמחלקה String. בנוסף, אין לייבא ספריות ואין להיעזר בפעולות של הספריות אשר יובאו לצורך הרצת אלגוריתם החיפוש.

### בדיקת התרגיל

בדיקת התרגיל תכלול בדיקה של הפעולות השונות אשר מפורטות לאורך המסמך, וכן בדיקה של חוקיות הפתרונות אותם ימצא אלגוריתם החיפוש.

הציון יקבע לפי הקוד אותו כתבתם, תקינותן של הפעולות השונות, תקינות הפתרונות המתקבלים וחלק תחרותי.

### חלק תחרותי

בתרגיל זה ישנו חלק תחרותי – אתם תתחרו מול ההגשות של שאר הסטודנטים בקורס על פתרון של מגוון לוחות, חלקם פשוטים וחלקם מאתגרים.

התחרות מתבססת על מציאת פתרון קצר ככל שניתן ללוחות נתונים. כל הרצה של אלגוריתם החיפוש תוגבל ל-60 שניות, והמטרה שלכם הינה למצוא פתרון (חוקי) קצר ככל שניתן בזמן שהוקצב.

משקלו של החלק התחרותי הינו 20 אחוז ממשקל התרגיל ומשקל שאר החלקים הוא 90 אחוז. הציון המקסימלי על התרגיל הוא 110.

### הרצת התוכנית

במחלקה Main קיימים מספר קטעי קוד אשר משמשים להרצת הקוד, וכן מוגדרת פעולה ראשית. **אין לשנות כלל** את תוכן המחלקה.

הפעולה הראשית מדפיסה הודעות בהתאם לפתרונות שמצאתם ללוחות השונים:

- במידה והצלחתם לפתור לוח מסוים יודפס אורך הפתרון ולאחר מכן הפתרון עצמו.
- במידה ומצאתם כי ללוח מסוים לא קיים פתרון יודפס כי הלוח אינו פתיר.
- במידה והתקבלה שגיאת זיכרון במהלך חיפוש מסוים (ייתכן, שכן מרחב החיפוש גדול מאוד) תודפס הודעה שנגמר הזיכרון.
- במידה ולא הגעתם לפתרון ונגמר הזמן המוקצה לבעיה תודפס הודעה שנגמר הזמן.

נוסף לכך, לכל לוח יודפס מספר המצבים שהורחבו במהלך ריצת אלגוריתם החיפוש, ולבסוף יודפס כמה לוחות בסך הכל הצלחתם לפתור מתוך כלל הלוחות הנתונים.

שימו לב כי לתרגיל זה לא מצורף קובץ פלט, ועליכם לבדוק באופן עצמאי את התקינות של כלל הפעולות ושל הפלטים המתקבלים מהרצת התוכנית.

בהצלחה