

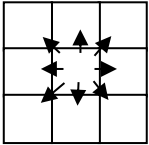
## תרגיל ריצה - חיפוש

### הבעיה

עליכם לממש מנוע חיפוש התומך במספר אלגוריתמי חיפוש לפתרון הבעיה הבאה :  
נתון לוח בגודל  $N \times N$  המדמה מפה טופוגרפית. כל משבצת מתארת אופי של תא שטח במפה. תא שטח יכול לייצג כביש עפר, אדמה רגילה, חול או צוק. נתונות גם משבצות ההתחלה, בה נמצאת מכונית שטח אוטונומית, ונקודת הסיום. מטרת המכונית היא למצוא את המסלול הזול ביותר מנקודת ההתחלה לנקודת הסיום תוך שימוש באלגוריתמים שנלמדו.

### חוקי התנועה

המכונית רשאית לנוע משבצת אחת בכל פעם לאחת משמונה המשבצות שמסביבה כמו באיור הבא :



### עלות המסלול

בתנועה לאחד מארבעת הכיוונים הבסיסיים (ימין, שמאל, למעלה, למטה) לכל משבצת יש עלות :

סימון בקובץ הקלט	תואי השטח	עלות
D	כביש עפר	1
R	אדמה רגילה	3
H	חול	5
X	צוק	-
S	נקודת התחלה	0
G	נקודת סיום	5

לתנועה באלכסון יש עלות זהה, למעט במקרה של תנועה באלכסון למשבצת עם חול. במקרה זה העלות היא 10. בנוסף, ידוע שבכל המפה יש לכל היותר 10% משבצות של כביש עפר, ולכל היותר 10% משבצות של אדמה רגילה.

### מימוש

#### קלט

התוכנית תקרא את כל הקלט שלה מקובץ יחיד - input.txt. השורה הראשונה בקובץ תקבע באיזה אלגוריתם להשתמש :

DFBnB, IDA\*, A\*, DFID, BFS, counter-clockwise. ז"א, במידה וצוין בשורה זאת clockwise, סדר הפעלת האופרטורים על קודקוד נתון יהיה ימינה, אלכסון ימינה-למטה, למטה, אלכסון שמאלה-למטה, שמאלה, אלכסון שמאלה-למעלה, למעלה, אלכסון ימינה-למעלה. במידה וצוין שהסדר הוא counter-clockwise, הסדר יהיה ימינה, אלכסון ימינה-למעלה, למעלה, אלכסון שמאלה-למעלה, שמאלה, אלכסון שמאלה-למטה, למטה, אלכסון ימינה-למטה.

בנוסף, אם מדובר ב- A\* או DFBnB, לאחר המילה clockwise או counter-clockwise יהיה רווח, ואז יהיה כתוב כיצד להעדיף קודקודים בעלי ערך זהה בפונקציית ההערכה  $f(n)$ : old-first או new-first. אם נכתבה האופציה old-first, במקרה וקיימים כמה קודקודים בעלי אותו ערך של  $f(n)$  וצריך לבחור אחד מהם (כי ערך ה-  $f(n)$  שלהם הוא הקטן ביותר בתור עדיפויות), A\* יבחר קודם את הקודקוד שנוצר הכי מוקדם. אם נבחרה האופציה new-first, A\* יבחר קודם את הקודקוד שנוצר הכי מאוחר. בצורה דומה גם ב- DFBnB, שממין את הקודקודים לפי ערכי  $f(n)$ . אם נכתבה האופציה old-first, אם יש מספר קודקודים בעלי אותו ערך הם יסודרו לפי זמן הייצור שלהם, ובסדר הפוך לזמן הייצור אם נבחרה האופציה new-first.

השורה השלישית תקבע האם להדפיס את זמן הריצה (with time) או לא (no time). השורה הרביעית תקבע האם להדפיס למסך את ה- open-list בכל שלב של ריצת אלגוריתם החיפוש (with open) או לא (no open). השורה החמישית בקובץ תכיל את גודל הלוח (N). השורה השישית תכיל את הקואורדינטות של נקודת ההתחלה ונקודת הסיום בפורמט הבא :  $(X_{start}, Y_{start}), (X_{end}, Y_{end})$ . המשבצות בלוח ימוספרו כמו שמספרים את התאים במטריצה. ז"א המשבצת השמאלית העליונה היא (1,1) והמשבצת הימנית התחתונה היא (N,N). לאחר מכן יופיע הלוח לפי המוסכמות שהובאו לעיל.

## פלט

במידה ונכתב בקובץ הקלט no open, כל הפלט ייכתב לקובץ output.txt. אין להדפיס במקרה כזה שום דבר על המסך. בשורה הראשונה בקובץ יש לכתוב את המסלול שנמצא ע"י האלגוריתם. מסלול הפתרון מתואר על ידי סדרת הפעולות הנדרשות למעבר על המסלול. הפעולות הן: R (ימינה), RD (אלכסון ימינה-למטה), D (למטה), LD (אלכסון שמאלה-למטה), L (שמאלה), LU (אלכסון שמאלה-למעלה), U (למעלה), RU (אלכסון ימינה-למעלה). הפעולות יופרדו ע"י מקף. לדוגמא:

R-RD-D-D-LD מתאר הליכה ימינה, ימינה-למטה, למטה, למטה ושמאלה-למטה (יש להשתמש באותיות גדולות).

בשורה השנייה יש לכתוב "Num:" ואח"כ את מספר הקודקודים שיוצרו. יש לספור גם קודקודים שלא נכנסו ל-open list, ואם קודקוד נוצר כמה פעמים יש לספור כל פעם בה הוא נוצר. בשורה השלישית יש לכתוב "Cost:" ואח"כ את עלות המסלול שנמצא. אם בקובץ הקלט נכתב שיש להדפיס גם את זמן הריצה, בשורה הרביעית יש לכתוב את הזמן שלקח לאלגוריתם למצוא את הפתרון (בשניות).

במידה ונכתב בקובץ הפלט with open, הפלט ייכתב לקובץ ה-output בדיוק כמו מקודם, אלא שבנוסף יש להדפיס **למסך** את התוכן של ה-open-list בכל איטרציה של האלגוריתם (=לפני כל הוצאה מה-open list). אני לא מגדיר פורמט מדויק (כי האופציה הזאת היא יותר בשביל debug) אבל אני מצפה לראות בצורה ברורה איך נראה כל איד מהמצבים שנמצאים ב-open-list.

בנוסף יש להגיש קובץ וורד details.docx. בתחילת הקובץ יש לכתוב את פרטי המגיש (שם ות.ז.). לאחר מכן יש לתאר במילים את הפונקציה היוריסטית בה בחרתם להשתמש ולהוכיח מדוע היא admissible ו-consistent.

## דגשים

- BFS ו- $A^*$  ימומשו עם closed list. יש להשתמש ב-hash-table גם עבור ה-open list כמו שלמדנו.
- IDA\* ו-DFBnB ימומשו עם מחסנית וללא closed-list אך עם loop-avoidance, ז"א בדיקה האם הקודקוד המפותח נמצא על הענף שעליו אנחנו עובדים או כבר במחסנית.
- DFID ימומש בצורה רקורסיבית, ללא closed-list אך עם loop-avoidance.
- אם לא נמצא מסלול יש לכתוב: "no path" בשורה הראשונה של קובץ הפלט. בשורה השנייה יש לכתוב "Num:" ואח"כ את מספר הקודקודים שיוצרו. בשורה השלישית יש לכתוב "Cost: inf". במידה ונכתב בקובץ הקלט with time, יש לכתוב בשורה הרביעית את הזמן שלקח לאלגוריתם (בשניות).
- יחס הסדר שנקבע הוא סדר יצירת הקודקודים, ולכן באלגוריתמים שמבוססים על DFS הקודקודים למעשה יצאו בסדר הפוך לסדר יצירת המצבים.
- ב-DFID האיטרציה הראשונה היא כאשר  $l=1$ , כי ברור שהמצב ההתחלתי אינו המצב הסופי.
- למרות שהמטרה שלנו היא מציאת המסלול הזול ביותר, BFS ו-DFID לא ימצאו בהכרח את המסלול הזול ביותר אלא את המסלול הקצר ביותר (=עובר בהכי פחות משבצות). עדיין בקובץ הפלט יש להחזיר את העלות של המסלול שהם מצאו (ולא את מספר הצעדים של המסלול).
- יש לממש את האלגוריתמים לפי מה שלמדנו בכיתה. בפרט, אין לבצע פעולה ומיד אחריה את הפעולה ההופכית לה (כמו תזוזה ימינה ואז תזוזה שמאלה).
- כדי לא לאבד סתם נקודות, הקפידו על פלט בדיוק לפי ההוראות: רווחים, אותיות גדולות, RU ולא UR, וכו'.

## אופן הניקוד

- קוד נכון, שמממש את האלגוריתמים כמו שנלמדו בכיתה, ומחזיר את התוצאה המבוקשת על כל הקלטים החוקיים.
- איכות הפונקציה היוריסטית בה בחרתם להשתמש ב- $A^*$ , IDA\* ו-DFBnB (זו כמוכן אותה הפונקציה), ונכונות ההוכחה שהפונקציה היא admissible ו-consistent.
- קוד מתועד וקריא (שמות משתנים ופונקציות משמעותיים).
- הגשה בזמן.

## פרטי ההגשה

- ההגשה לפי כל כללי הגשת המטלות, כפי שמתואר בנוהל הגשת המטלות (נמצא במודל).
- ניתן לכתוב את התוכנית ב-Java בלבד, והיא צריכה להתקמפל ולרוץ בגרסת 1.8. שם המחלקה בה נמצאת פונקציית ה-main יהיה Ex1. יש להשתמש ב-default-package בלבד (ללא תתי תיקיות). חובה להגיש את קבצי המקור.
- אין לממש GUI.
- עליכם להניח שקובץ ה-input.txt (שאתם מקבלים כקלט) נמצא באותה ספרייה בה נמצאת התוכנית, ולכן אין לקרוא את המיקום שלו כארגומנט או לציין ספרייה ספציפית בקוד שאתם מגישים (במידה וכן, ירדו על כך נקודות).
- קובץ ה-output.txt (שאתם מוציאים כפלט) צריך להיכתב באותה ספרייה בה נמצאת התוכנית, ולכן אין לקרוא את המיקום שלו כארגומנט או לציין ספרייה ספציפית בקוד שאתם מגישים (במידה וכן, ירדו על כך נקודות). יש להניח שהקובץ לא קיים, עליכם ליצור אותו בספרייה בו התוכנית רצה, ולכתוב לתוכו את הפלט.
- יינתן קלט ופלט לדוגמה. ודאו שתוכנתכם עובדת אותו כמו שצריך, אך זהו לא הקלט היחיד אותו תיבדק התוכנית.
- התוכנית תיבדק דרך ה-command line ולא ב-eclipse. לכן, כדי לוודא שהתוכנית שלכם עובדת עליכם להעתיק את קבצי המקור ואת הקובץ input.txt שניתן כדוגמה לאחת הספריות במחשב, לפתוח command line ולהריץ `java *.java` ואז `Ex1.java`. התוכנית תיצור את הקובץ output.txt באותה ספרייה והוא צריך להיות זהה לקובץ output.txt שניתן כדוגמה.
- ההגשה נעשית דרך מערכת הגשות submit בכתובת <http://submit.org.il/ariel>. שם המשתמש שלכם הוא שם המשתמש שהוקצה לכם באי-מייל שלכם באריאל, כמו שכתוב במכלול. לדוגמה, אם הדואר שלכם במכלול הוא `israel.israeli@msmail.ariel.ac.il`, אז שם המשתמש שלכם הוא: `israel.israeli`. הסיסמה היא 4 הספרות האחרונות של תעודת הזהות.
- ברגע שתגישו את המטלה תקבלו למייל הפרטי שלכם שרשום במכלול (לא המייל באריאל) את תוצאות ההגשה- האם התרגיל התקמפל, ואם כן מה היה הפלט שלכם בהשוואה למה הפלט הרצוי. תוכלו כמובן להגיש שוב ולדרוס את ההגשה הקודמת. מכיוון שכתובת המערכת לא מעודכנת ב-DNS של אריאל, המייל מהמערכת יגיע (ככל הנראה) לתיקיית הספאם. תבדקו שם וסמנו אותו כ-"not spam". בנוסף, המערכת לא מצליחה לשלוח מייל ל-walla.
- תאריך הגשת התרגיל- 17.05.23
- שימו לב: אין לחכות לרגע האחרון כדי להגיש! לקראת סוף הזמן ההגשה השרת עמוס, ולכן ה-feedback מתקבל בשלב מאוחר יותר (אחרי שתאריך ההגשה כבר עובר). בנוסף, כל הגשה דורסת את ההגשה הקודמת, ולכן אם חיכיתם לרגע האחרון ומתוך לחץ הגשתם בטעות גרסה לא מעודכנת, זו הגרסה שתיבדק.

**בהצלחה!**