

תרגיל ריצה - חיפוש

הבעיה

עליכם לממש מנוע חיפוש התומך במספר אלגוריתמי חיפוש כדי לפתור את המשחק הבא. נתון לוח של 3×3 תאים, המכיל 6 גולות: 2 אדומות, 2 כחולות ו-2 ירוקות. הגולות מונחות על הלוח בסדר התחלתי נתון כלשהו, והמטרה היא למצוא את מספר הפעולות הזול ביותר מהמצב ההתחלתי למצב הסופי, שנתון אף הוא.

הפעולות

ניתן להזיז כל גולה לתא שנמצא בסמוך אליה במאוזן או במאונך, אך לא באלכסון. הלוח מעגלי, ולכן אם מזיזים גולה מקצה הלוח היא מגיעה לצד השני. בנוסף, ניתן להזיז גולה לתא רק בתנאי שהוא ריק (=שאין בו כרגע גולה), ושהוא לא צבוע בשחור. עלות הזזה של גולה כחולה היא 1, גולה ירוקה 3, וגולה אדומה 10.

לדוגמה, אם הלוח במצב הזה:

●	●	
●	●	
●	●	

נוכל להזיז גולה כחולה אחת ימינה, ואז למעלה, גולה ירוקה אחת שמאלה, וגולה ירוקה אחת למעלה כדי להגיע למצב הזה:

●	●	●
●	●	
		●

עלות המסלול המתואר תהיה $1+1+3+3=11$.

אם הלוח במצב הזה:

●	●	
●	●	■
●	●	

לא נוכל להזיז גולה כחולה, עד שנוזיז אחת מהגולות האדומות או הכחולות.

מימוש

קלט

התוכנית תקרא את כל הקלט שלה מקובץ יחיד - input.txt. השורה הראשונה בקובץ תקבע באיזה אלגוריתם להשתמש: BFS, DFID, A*, IDA*, או DFBnB. השורה השנייה בקובץ תקבע האם להדפיס את זמן הריצה (with time) או לא (no time). השורה השלישית תקבע האם להדפיס למסך את ה- open list בכל שלב של ריצת אלגוריתם החיפוש (with open) או לא (no open). לאחר מכן יופיע הסידור ההתחלתי של הלוח לפי שורות, כאשר יש פסיקים בין התאים. תא שמכיל גולה אדומה יסומן כ-R, גולה כחולה יסומן כ-B, וגולה ירוקה יסומן כ-G. תאים ריקים יסומנו כ- "_", ותאים שחורים יסומנו כ- X. לאחר מכן תופיע שורה בה יהיה כתוב "Goal state:" ולאחריה יופיע הסידור הסופי של הלוח אליו צריך להגיע (באותו פרמט של הסידור ההתחלתי). ניתן להניח שקובץ הקלט תקין.

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 סדרת הפעולות שנמצאה ע"י האלגוריתם. 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 הפתרון שנמצא. אם בקובץ הקלט נכתב שיש להדפיס גם את זמן הריצה, בשורה הרביעית יש לכתוב את הזמן שלקח לאלגוריתם למצוא את הפתרון (בשניות).

הפעולות יסומנו ע"י המיקום של הגולה שזוהה, הצבע שלה, ולאיה תא היא זוהה. לשם כך נמספר את התאים, כמו שממספרים את התאים במטריצה. ז"א התא השמאלי העליון הוא 1,1 והתא הימני התחתון הוא 3,3. יש לכתוב נקודתיים בין כל מאפיין של הפעולה, והפעולות עצמם יופרדו ע"י מקף כפול.

לדוגמה, המסלול המתואר קודם ייכתב בקובץ הפלט כ-

(2,2):B:(2,3)--(2,3):B:(1,3)--(3,1):G:(3,3)--(3,2):G:(2,2)

עבור A^* , נחיל יחס סדר על קודקודים בעלי ערך זהה בפונקציית ההערכה $f(n)$, לפי זמן הייצור שלהם. זאת אומרת, נניח שיש שני קודקודים a, b בעלי אותו ערך של $f(n)$, וכרגע A^* צריך לבחור אחד מהם (כי ערך ה- $f(n)$ שלהם הוא הקטן ביותר בתור העדיפויות). במקרה כזה A^* יבחר את a אם הוא נוצר באיטרציה השנייה ו- b נוצר באיטרציה הרביעית. כך גם ב- $DFBnB$ שממין את הקודקודים לפי ערכי $f(n)$, אם יש מספר קודקודים בעלי אותו ערך הם יסודרו לפי זמן הייצור שלהם.

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 הלוח בכל אחד מהמצבים שנמצאים ב- open-list.

בנוסף, יש להגיש דרך ה- Moodle קובץ וורד או pdf בשם details. בתחילת הקובץ יש לכתוב את פרטי המגיש (שם ות.ז.). לאחר מכן יש לתאר במילים את הפונקציה היוריסטית בה בחרתם להשתמש ולהוכיח בצורה פורמאלית 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.
- עליכם לקבוע בעצמכם מהו החסם העליון הראשוני עבור $DFBnB$.
- אין לספור את היצירה של קודקוד ההתחלה.
- 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". אם בקובץ הקלט נכתב שיש להדפיס גם את זמן הריצה, בשורה הרביעית יש לכתוב את הזמן שלקח לאלגוריתם להחליט שאין פתרון (בשניות).
- 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 (=עם הכי פחות פעולות הזזה). 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 הזזת גולה ימינה ואז החזרתה שמאלה).
- כדי לא לאבד סתם נקודות, הקפידו על פלט בדיוק לפי ההוראות: רווחים, אותיות גדולות, וכו'.
- יש לתעד את הקוד ולתת שמות משתנים ופונקציות משמעותיים, אבל אין לכתוב הערות בעברית.

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 כמובן אותה הפונקציה), ונכונות ההוכחה שהפונקציה היא admissible ו-consistent.
- 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 Java 23. 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 `javac *.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.
- הגשת הקובץ details נעשית דרך ה-Moodle, באותו תאריך של הגשת קבצי המקור במערכת ה-submit.
- תאריך הגשת התרגיל- 24.12.24
- 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).

בהצלחה!