

תרגיל ריצה - חיפוש

הבעיה

עליכם לממש מנוע חיפוש התומך במספר אלגוריתמי חיפוש כדי לפתור את משחק ה- Bounded NxM-tile puzzle (וריאנט של ה- tile-puzzle שראינו בשיעור).

במשחק נתון לוח בגודל NxM המכיל NxM-1 בלוקים הממוספרים מ-1 ועד NxM-1 ובלוק ריק. חלק מהבלוקים צבועים בלבן וחלק צבועים באדום. בנוסף, על כל בלוק לבן רשום כמה הזזות ניתן לבצע אתו. הבלוקים מסודרים בסדר התחלתי נתון כלשהו, והמטרה היא למצוא את מספר הפעולות הזול ביותר מהסידור ההתחלתי למצב הסופי. במצב הסופי כל הבלוקים מסודרים מ-1 ועד NxM-1 משמאל לימין ומלמעלה למטה (בלי קשר לצבע שלהם), כאשר הבלוק הריק נמצא בפינה הימנית תחתונה. לדוגמה, אם הלוח הוא 3x4 אז המצב הסופי הוא:

1	2	3	4
5	6	7	8
9	10	11	

שימו לב שהצבע של כל בלוק וכמות ההזזות המותרות של הבלוקים הלבנים הם חלק מהגדרת המצב ההתחלתי.

הפעולות

בניגוד למשחק ה- tile-puzzle הרגיל שראינו, בו כל הזזה נחשבת צעד אחד, במשחק הזה ישנן כללים שונים ועלויות שונות התלויות בצבע הבלוק. הזזה של בלוק לבן עולה 1, וניתן להזיז אותו לבלוק הריק רק אם לא עברנו את כמות ההזזות המותרות עבורו. הזזה של בלוק אדום לבלוק הריק עולה 30, ואין הגבלה על כמות ההזזות שניתן לבצע אתו. לדוגמה, אם הלוח במצב הזה, כאשר בלוקים 7 ו-11 הם אדומים, ובלוק 6 הוא לבן עם מגבלת הזזות של 1:

1	2	3	4
5		6	8
9	10	7	11

נוכל להזיז את 6 שמאלה ואז לא נוכל להזיז אותו יותר. אח"כ נוכל להזיז את 7 למעלה ואת 11 שמאלה, כדי להגיע למצב הסופי. עלות המסלול המתואר תהיה $1+30+30=62$.

מימוש

קלט

התוכנית תקרא את כל הקלט שלה מקובץ יחיד - input.txt. השורה הראשונה בקובץ תקבע באיזה אלגוריתם להשתמש: 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). השורה הרביעית תכיל את גודל הלוח בפורמט הבא: NxM, ז"א לוח המכיל N שורות ו-M עמודות. בשורה החמישית יהיה כתוב White: ואז רשימה של מספרי הבלוקים הלבנים עם המגבלה של כל אחד מהם. לדוגמה, אם הבלוק 7 הוא לבן עם מגבלת הזזות של 2, ובלוק 2 הוא גם לבן עם מגבלת הזזות של 3, הרשימה תיראה כך: White: (2,3),(7,2). במידה ואין בלוקים לבנים השורה תכיל רק את המילה White. כל שאר הבלוקים שהם לא לבנים הם אדומים. לאחר מכן יופיע הסידור ההתחלתי של הלוח לפי שורות, כאשר יש פסיקים בין מספרי הבלוקים. הבלוק הריק יסומן כ- "_". 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 על ידי מספר הבלוק שזז וכיוון ההזזה: R (ימנית), D (למטה), L (שמאלה), U (למעלה). הפעולות יופרדו ע"י מקף. לדוגמה, המסלול המתואר קודם ייכתב בקובץ הפלט כ- 6L-7U-11L.

על מנת לקבל פלט אחד ככל שניתן, נקבע שסדר יצירת הקודקודים בעלי אב משותף יהיה לפי האופרטור שיצר אותם בסדר הבא: שמאלה, למעלה, ימינה, למטה. בנוסף, 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, או אם a ו- b נוצרו באיטרציה השנייה אבל a נוצר ע"י האופרטור למעלה, ו- b נוצר ע"י האופרטור ימינה. כך גם ב- DFBnB, שממייין 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 יכתב 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 המודל קובץ word או pdf בשם details. בקובץ עליכם לתאר במילים את הפונקציה היוריסטית בה בחרתם להשתמש ולהוכיח בצורה פורמאלית מדוע היא admissible ו- consistent.

דגשים

- A^* ימומש 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.
- עבור DFBnB החסם העליון הראשוני יהיה $\min(\text{Integer.MAX_VALUE}, (M \times N - 1)!) \cdot \text{Cost}$. ז"א, עבור מספר בלוקים קטן (עד 12, כולל), יש להשתמש בעצרת של מספר הבלוקים. מעבר לזה, יש להשתמש ב- Integer.MAX_VALUE.
- 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:" בלבד.
- ב- DFID האיטרציה הראשונה היא כאשר $l=1$, כי ברור שהמצב ההתחלתי אינו המצב הסופי.
- למרות שהמטרה שלנו היא מציאת המסלול הזול ביותר, DFID לא ימצא בהכרח את המסלול הזול ביותר אלא את המסלול הקצר ביותר (=עם הכי פחות פעולות הזזה). עדיין בקובץ הפלט יש להחזיר את העלות של המסלול שהוא מצא (ולא את מספר הצעדים של המסלול).
- יש לממש את האלגוריתמים לפי מה שלמדנו בכיתה. בפרט, אין לבצע פעולה ומיד אחריה את הפעולה ההפוכה (כמו הזזת בלוק ימינה ואז החזרתו שמאלה).
- כדי לא לאבד סתם נקודות, הקפידו על פלט בדיוק לפי ההוראות: רווחים, אותיות גדולות, 4x5 ולא 4X5, 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 כמובן אותה הפונקציה), ונכונות ההוכחה שהפונקציה היא admissible ו- consistent.
- קוד מתועד וקריא (שמות משתנים ופונקציות משמעותיים), אבל אין לכתוב הערות בעברית.
- הגשה בזמן.

פרטי ההגשה

- ההגשה ביחידים בלבד. תתבצע בדיקת העתקות.
- 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 javac *.java ואז java Ex1. התוכנית תיצור את הקובץ output.txt באותה ספרייה והוא צריך להיות זהה לקובץ output.txt שניתן כדוגמה.
- הגשה קבצי המקור 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) לתיקית הספאם. תבדקו שם וסמנו אותו כ-"not spam". בנוסף, המערכת לא מצליחה לשלוח מייל ל-walla. הגשת הקובץ details נעשית דרך המודל, באותו תאריך של הגשת קבצי המקור במערכת ה-submit.
- תאריך הגשת התרגיל- 19.02.24
- 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!