**שאלה 1**

**סעיף א**

המפרט נבחר על פי התבוננות באוסף הפעולות האפשריות שמופיעות בנספח בסוף התרגיל. הפעולות הדרושות על פי המסמך הן:

CreateGraph graphName: יכולת זאת ממומשת על ידי הקונסטראקטור הדיפולטי. התאמת השם לגרף ממומשת ב-TestDriver עצמו.

AddNode graphName nodeName: אנו ממשים יכולת זאת על ידי מתודת

AddEdge graphName parentNode childNode

ListNodes graphName

ListChildren graphName parentNode

**שאלה 2**

**סעיף א**

תיאור מילולי כללי של אופן פעולת האלגוריתם:

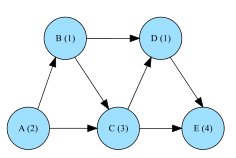
האלגוריתם עובר על הצמתים לפי מרחקם מהצמתים ב-starts כך שבכל שלב מטפלים בצומת הכי קרוב לאחד הצמתים שעדיין לא טופל. ברגע שהצומת המטופל הינו אחד מצמתי המטרה, המסלול אליו מוחזר.

האלגוריתם משתמש ב-3 מבני נתונים, finished, active paths. Paths מכיל מיפוים מצמתים, למסלול הכי קצר לצומת מצומת התחלה כלשהו. active תור עדיפויות שמכיל צמתים פוטנציאלים לטיפול. finished הוא סט הצמתים שסיימנו לטפל בהם.

כל צומת מטופל לא יותר מפעם אחת. בכל שלב, מוציעים את הצומת עם המרחק הקצר ביותר מ-active ומוסיפים את הילדים שלו שעדיין לא טופלו ולא נמצאים ב-active, ל-active ול-paths עם המרחק והמסלול המתקבלים עלידי הוספת הצומת למרחק ולמסלול של צומת האב שגילה אותם בהתאמה.

האלגוריתם מסתיים כאשר צומת שמטפלים בו נמצא ב-goals או כאשר active ריק (במקרה זה, אין מסלול מאף צומת ב-active לצומת ב-goals)

תיאור הדוגמה:



|  |  |  |  |
| --- | --- | --- | --- |
| finished | active | paths | שלב |
| {} | {A(2)} | {A:[A]} | 0(הכוונה לאתחול) |
| {A} | {C(4),B(3)} | {A:[A], B:[A,B], C:[A,C]} | 1 |
| {A,B} | {C(5),D(4)} | {A:[A], B:[A,B], C:[A,C], D:{A,B,D}} | 2 |
| {A,B,D} | {E(8),C(5)} | {A:[A], B:[A,B], C:[A,C], D:{A,B,D}, E{A,B,D,E}} | 3 |
| {A,B,D,C} | {E(8)} | {A:[A], B:[A,B], C:[A,C], D:{A,B,D}, E{A,B,D,E}} | 4 |
| {A, B, D, C} | {} | {A:[A], B:[A,B], C:[A,C], D:{A,B,D}, E{A,B,D,E}} | 5 |

האלגוריתם מאתחל את active להכיל את A, paths שמכיל מיפוי מ-A למסלול שמכיל את הצומת A בלבד ו-finished ריק.

**בשלב הראשון:** A נשלף מ- activeוהבנים שלו(B, C) נוספים לתור עם הערכים שרשומים בסוגריים בעמודת ה-active בשלב 1. המסלולים שנוצרים על ידי הוספת C ו B לסוף המסלול ב-path שמתאים ל-A ממופים ב-paths ל-B ו C בהתאמה. A נכנס ל- finished.

**בשלב השני:** B נשלף מ-active מכיוון שאורך המסלול עליו הוא מינימלי מבין הצמתים שב-active. C כבר נמצא ב-active ולכן D בלבד נוסף ל-active. המסלול שנוצר על ידי הוספת D לסוף המסלול ב-path שמתאים ל-B, ממופה ב-paths ל-D. B נכנס ל- finished.

**בשלב השלישי:** D נשלף מ-active מכיוון שאורך המסלול עליו הוא מינימלי מבין הצמתים שב-active. E הוא הבן היחד של D והוא נוסף ל-active . המסלול שנוצר על ידי הוספת E לסוף המסלול ב-path שמתאים ל-D, ממופה ב-paths ל-E. D נכנס ל- finished.

**בשלב הרביעי:** C נשלף מ-active מכיוון שאורך המסלול עליו הוא מינימלי מבין הצמתים שב-active. E כבר נמצא ב-active ו-D נמצא ב-finished ולכן אף צומת לא נוסף ל- active ולא חל שינוי ב-paths. C נכנס ל-finished.

**בשלב החמישי:** E נשלף מ-active ומכיוון שהוא נמצא ב-goals המסלול המתאים ל-E ב-paths מוחזר.

**סעיף ג**