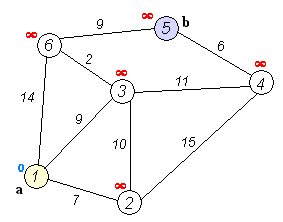
אלגוריתם דייקסטרה:

מקבלים גרף הנתון כרשימת שכנויות:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ... | ... | ... | …. |  |  |



**מטרה**: למצוא מסלול קצר(קל) ביותר בין קודקוד מקור S לבין שאר הקודקודים. (לבין קודקוד יעד T)

**אלגוריתם**: מקבלים גרף G ומקור s ויעד t

1. מגדירים לכל קודקוד עדיפות: .
2. מגדירים תור עדיפויות: Q. התור מכיל זוגות של int המייצג את הקודקוד ו - int המייצג את העדיפות.
3. מאתחלים: ומכניסים את s ל Q עם עדיפות 0.
4. כל עוד התור לא ריק:
   1. מוצאים את הקודקוד v בעל עדיפות הנמוכה ביותר בתור ומסמנים אותו.
   2. עבור כל שכן u של v (מגיעים לשכנים ולצלעות דרך הגרף - מערך רשימות)

אם הוא לא מסומן בדוק:

אם (כלומר אם עדיף להגיע ל v + לעבור בצלע בין v ל u מאשר המרחק שהיה עד עכשיו אצל u) כאשר היא הצלע בין v ל u.

* + - * 1. אם אז הכנס את u לתור Q עם עדיפות:
        2. אחרת (היה כבר בתור) עדכן את u בתוך התור לעדיפות .

1. החזר את

**סיבוכיות**: אתחול תור: . אתחול לכל קודקוד - כאשר: .  
נשים לב שכל קודקוד נכנס בדיוק פעם אחת לתור ויוצא בדיוק פעם אחת. בכל פעם שיוצא קודקוד, עוברים על כל שהכנים שלו ולכן סה"כ כמות האיטרציות שתהיה מ 2 הלולאות היא כמספר הצלעות בגרף: .  
לכל קודקוד מבצעים בתוך הלולאה החיצונה הוצאה מהתור ב (עדכון המינימום הבא בתור בערימה בינארית) סה"כ הפעולה הזאת תתבצע: פעמים.  
מכיוון שכל קודקוד שיוצא מהתור מסומן מיד, פעולות הלולאה הפנימית יתבצעו על כל צלע פעם אחת (כי עבור הכיוון השני - נקבל שכן מסומן ולא ניכנס פנימה)  
עלות הפעולות הפנימיות (של הלולא הפנימית) היא: עבור השאלות + עבור עדכון/הכנסה.  
לכן סה"כ יבוצעו לכל היותר: פעולות פנימיות.  
סה"כ פעולות:

**הוכחה**: נוכיח באינדוקציה שבכל איטרציה, אם קודקוד יוצא מהתור ומסומן אז: .  
בסיס: באיטרציה הראשונה, יוצא מהתור עם . לפי האיתחול ולכן מתקיים.  
צעד: נניח שבאיטרציה ה k קודקוד יוצא מהתור. כמובן מתקיים: .  
יהי המסלול הקצר ביותר מ s ל v. קודקוד s בהכרח סומן כי הוא יצא ראשון. קודקוד v יצא רק עכשיו ולכן אינו מסומן. מכאן: קיים קודקוד u על P כך שהוא הראשון שאינו מסומן ויהא z הקודקוד הקודם לו במסלול שלפי ההנחה, z מסומן. לפי הנחת האינדוקציה: . כי z יצא באחת האיטרציות הקודמות.  
מכיוון ש P הוא הכי קצר אז גם תת המסלול שלו מ s ל u גם הוא הכי קצר. מכאן: . מכאן, קודקוד u קיבל את הערך המינימאלי לפי האלגוריתם באיטרציה של z, כלומר:. באיטרציה ה k גם u וגם v היו בתוך התור אבל האלגוריתם בחר ב v. מכאן:  
 לכן: . מש"ל.  
כל קודקוד ייכנס לתור ויצא ולכן האלגוריתם יעדכן את כולם.

אלגוריתם Dijkstra הפוך

**מטרה**: למצוא מסלול קצר(קל) ביותר בין קודקוד מקור S לבין קודקוד יעד T.

הרעיון: להריץ את אלגוריתם דייקסטרה החל מ S והחל מ T (בכיוון ההפוך) במקביל.

ברגע שמגיעים לקודקוד הראשון שיצא מתור העדיפויות גם בשלב החל מ S וגם בשלב החל מ T - הוא מייצג את המסלול הקצר (הקל) ביותר מ S ל T ואורך המסלול הכולל יהיה אורך המסלול מ S עד לאותו קודקוד + מ T עד לאותו קודקוד בכיוון ההפוך.

אם הגרף הוא מכוון אז איך נעשה דייקסטרה רגיל החל מ T? נבנה את הגרף שבו כיווני החיצים הפוכים: .

האלגוריתם:

בהינתן גרף כרשימת שכנויות, קודקוד מקור, קודקוד יעד.

1. בנה את הגרף (אותן רשימות קודקודים רק שאם היה ברשימה של אז כעת נכניס את לרשימה של אבל לא את u לרשימה של v)
2. הרץ את אלגוריתם דייקסטרה על ובמקביל את אלגוריתם דייקסטרה על שלב מהאלגוריתם הראשון ושלב מהאלגוריתם השני לסירוגין.
3. אם הגענו בפעם הראשונה לקודקוד שיצא מ 2 התורים (ביקרנו בו ב 2 האלגוריתמים). עבור על כל הקודקודים והחזר את הסכום המינימאלי עבור: לכל קודקוד .

סיבוכיות:

בניית בסיבוכיות: .

הרצת אלגוריתמי דייקסטרה: + עבור הבדיקה.

סה"כ:. (נציין שמכיוון שעובדים במקביל ועוצרים כאשר מגיעים לקודקוד הראשון שטופל ב 2 האלגוריתמים אז נבצע במקרה הטוב כחצי מהאיטרציות שביצענו בדייקסטרה הרגיל.

הוכחה:

לפי נכונות אלגוריתם דייקסטרה, אם יש מסלול מ ל אז בהכרח, כל קודקוד במסלול מ ל יצא מהתור באיזשהו שלב. באותו אופן עבור המסלול ההפוך.

וכן, אם יש מסלול קצר ביותר ב G בין ל אז אותו מסלול הוא מסלול קצר ביותר מ ל ב . (אחרת, אם יש קצר יותר, נהפוך צלעות בחזרה ונקבל מסלול קצר יותר בכיוון השני)

נותר להוכיח: ברגע שיצא הקודקוד הראשון מ 2 התורים קיים קודקוד על המסלול הקצר ביותר בין ל וחיבור המסלולים עד אליו וממנו, נותן את המסלול הקצר ביותר. כלומר קיים u עבורו: .

הוכחת הטענה: יהי הקודקוד הראשון שטופל ב 2 האלגוריתמים. נסמן ב את מרחקו מ שהתקבל ע"י דייקסטרה הראשון. נסמן ב את מרחקו מ שהתקבל ע"י דייקסטרה בכיוון ההפוך.  
יהי מסלול קצר ביותר בין ל , נחלק למקרים לפי מה שקרה בזמן ש יצא:  
מקרה 1: קיים קודקוד ב שלא יצא מאף אחד מהתורים. ( לא טופל באף איטרציה ובאף כיוון)  
מכאן, לפי נכונות דייקסטרה, כי לפי הנכונות, מי שיוצא ראשון מהתור, יש לו מרחק קטן יותר מ . באותו אופן: אבל מכאן:

אבל אם כך, . ולכן עבור v קיבלנו את התשובה הנכונה.

מקרה 2: כל קודקוד ב יצא מ (התור החל מ ) ולא מ (התור החל מ ) או להיפך . (לא ייתכן קודקוד שיצא מ 2 התורים כי הוא הראשון ובו עוצרים) מכיוון ש יוצא ראשון מ וגם יוצא ראשון מ ורצים לסירוגין, אזי קיימים 2 קודקודים סמוכים ב כך ש: יצא מ ו יצא מ . נבחר את הראשונים שמקיימים זאת. לפי נכונות דייקסטרה, ומתקיים כי u,w על המסלול הקצר ביותר.  
נשים לב: . כי כאשר u יצא מ הוא עדכן את w.  
אבל מכאן:

ומכאן: . כי מתאר אורך מסלול בין s ל t.