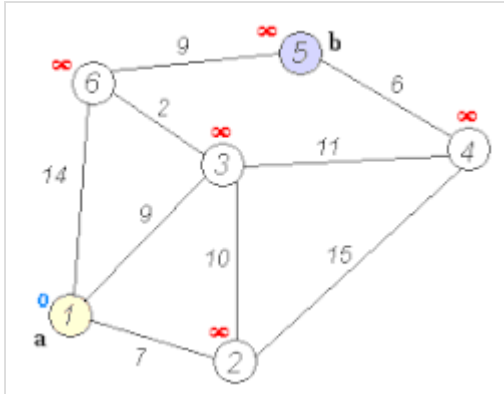


אלגוריתם דייקסטרה



אנימציה להמחשת האלגוריתם

אלגוריתם דייקסטרה, פרי יצירתו של אדסחר דייקסטרה^[1], הוא אלגוריתם למציאת המסלול הקל ביותר (כלומר שסכום משקלות קשתותיו הוא המינימלי האפשרי) מקדקוד (צומת) מקור לקדקוד יעד בגרף ממושקל, או למציאת כל המסלולים הקלים ביותר בגרף מקודקוד מקור לשאר הקודקודים. לכן הוא נקרא לעיתים אלגוריתם למציאת המסלולים הקלים ממקור יחיד.

האלגוריתם עובד על גרף מכוון או לא מכוון, בעל משקולות אי-שליליות על הקשתות. המשקולות בגרף מסמלות מרחק. משמעותו של המסלול הקצר ביותר בין שתי נקודות היא המסלול בעל סכום המשקולות הנמוך ביותר בין שתי הנקודות.

תוצאת האלגוריתם של דייקסטרה זהה לתוצאת אלגוריתם בלמן-פורד אך אלגוריתם בלמן-פורד פועל גם על גרפים הכוללים קשתות שמשקלן שלילי. לעומת זאת, זמן הריצה של אלגוריתם דייקסטרה מהיר יותר.

פעולת האלגוריתם

בתמצית ניתן לסכם את פעולת האלגוריתם כך:

עבור כל קודקוד, מסומן האם ביקרו בו או לא ומה מרחקו מקודקוד המקור, אותו נסמן ב-S. בהתחלה כל הקודקודים מסומנים שלא ביקרו בהם, ומרחקם מוגדר כאינסוף.

לולאת האלגוריתם:

- כל עוד נותרו קודקודים שלא ביקרנו בהם:
- מסמנים את X (הקודקוד הנוכחי. באיטרציה הראשונה זהו קודקוד המקור S) כקודקוד שביקרו בו.
- עבור כל קודקוד Y שהוא שכן של X ועדיין לא ביקרנו בו:
 - Y מעודכן, כך שמרחקו יהיה שווה לערך המינימלי בין שני ערכים: בין מרחקו הנוכחי, לבין משקל הקשת המחברת בין X לבין Y ועוד המרחק בין S ל-X.
- אם כל הקודקודים בוקרו האלגוריתם עוצר. אחרת בוחרים קודקוד X חדש בתור הקודקוד שעדיין לא בוקר ושמרחקו בשלב הזה מצומת המקור S הוא הקל ביותר מבין כל הקודקודים בגרף שטרם ביקרנו בהם.

פסאודו קוד:

```

1 function Dijkstra(Graph, source):
2
3     create vertex set Q
4
5     for each vertex v in Graph:           // Initialization
6         dist[v] ← INFINITY                // Unknown distance from source to v
7         prev[v] ← UNDEFINED              // Previous node in optimal path from source
8         add v to Q                        // All nodes initially in Q (unvisited nodes)
    
```

```

9
10     dist[source] ← 0 // Distance from source to source
11
12     while Q is not empty:
13         u ← vertex in Q with min dist[u] // Node with the least distance will be
selected first
14         remove u from Q
15
16         for each neighbor v of u: // where v is still in Q.
17             alt ← dist[u] + length(u, v)
18             if alt < dist[v]: // A shorter path to v has been found
19                 dist[v] ← alt
20                 prev[v] ← u
21
22     return dist[], prev[]

```

האלגוריתם מסתיים כאשר קודקוד X החדש הוא היעד או (למציאת כל המסלולים המהירים ביותר) כאשר ביקרנו בכל הקודקודים. סיבוכיות האלגוריתם תלויה במבנה הנתונים השומר את הקודקודים שטרם ביקרנו בהם, אם מדובר ברשימה או במערך, הסיבוכיות היא ב- $O(|V|^2)$, אם משתמשים בערימה בינארית סיבוכיות הריצה נהיית $O(|E| \log |V| + |V| \log |V|)$ ואם משתמשים בערימת פיבונאצ'י הסיבוכיות משתפרת ל- $O(|E| + |V| \log |V|)$.

רעיון האלגוריתם

אלגוריתם דייקסטרה מבוסס על שני רעיונות אלגוריתמיים. הראשון הוא שימוש בתת-בעיות כדי לפתור את בעיית המסלול הקצר ביותר.

יהי P המסלול הקצר ביותר מקודקוד S לקודקוד T . ייתכן ש- P עובר בקודקודים רבים. עבור כל קודקוד V , החלק ממסלול P שמגיע מ- S אליו מהווה את הדרך הקצרה ביותר האפשרית להגיע מ- S ל- V . אם לא כן, ניתן היה לקצר את המסלול מ- S ל- V ובכך להקטין את המרחק הכולל מ- S ל- T , ואם כך P אינו המסלול הקצר ביותר, בסתירה להגדרתו.

עיקרון זה מאפשר לבסס את מציאת המסלול הקצר ביותר על מעין תכנון דינאמי. הפתרון לבעיה מצוי בפתרון לתת הבעיות שלה. המסלול הקצר ביותר מ- S ל- T יימצא על סמך המסלולים הקצרים ביותר מ- S לקודקודים שקדמו ל- T .

עיקרון שני הוא חמדני. בכל שלב מסוים במצב שבו נקבע מרחקם הקצר ביותר מ- S של חלק מהקודקודים, אך לא נקבע מרחקם של קודקודים אחרים. הרעיון החמדני קובע כי בכל שלב יש להתבונן באלו מהקודקודים שלא נקבע מרחקם ושיש אליהם קשת מקודקודים שמרחקם כבר נקבע, וליטול מקבוצה זו את הקודקוד שהדרך אליו מ- S , שעוברת כולה בקודקודים שכבר נקבע מרחקם, היא הקצרה ביותר.

ההגיון בבסיס בחירה זו הוא שאין קצר יותר מהקצר ביותר. לא ניתן להגיע אל קודקוד בדרך קצרה יותר מאשר על ידי בחירת הדרכים הקצרות ביותר הזמינות בכל שלב. עיקרון זה אינו נכון כאשר יש משקולות שליליות לקשתות. ייתכן שדווקא קשת ארוכה תוביל אותנו לקשת שתסמל מרחק שלילי גדול, וסכום המרחק הכולל לקודקוד יהיה נמוך יותר מאשר אם נלך בעקבות הקשת הקצרה שנבחרה מיידית.

ראו גם


- [בעיית המסלול הארוך ביותר](#)

קישורים חיצוניים

[מצגת הסבר - אלגוריתם דייקסטרה](https://web.archive.org/web/20190903162852/https://skydrive.live.co) (<https://web.archive.org/web/20190903162852/https://skydrive.live.co>)
([m/#!/view.aspx?cid=80490E0FD59409CF&resid=80490E0FD59409CF%21107&app=PowerPoint](https://web.archive.org/web/20190903162852/https://skydrive.live.co))

מדיה וקבצים בנושא [אלגוריתם דייקסטרה](#) בוויקישיתוף 

הערות שוליים

1. [Dijkstra, E. W. \(1959\)](#).  "A note on two problems in connexion with graphs" (<http://www-m3.ma.tum.de/twiki/pub/MN0506/WebHome/dijkstra.pdf>) (PDF). *Numerische Mathematik*. **1**: 269–271. doi:10.1007/BF01386390 (<https://doi.org/10.1007%2FBF01386390>).

▪