자료구조

L11: Graph (2)

2022년 1학기

국민대학교 소프트웨어학부

Overview

- Shortest paths problems
- Dijkstra’s algorithm
- Cost analysis of Dijkstra’s algorithm

Shortest Paths Problems

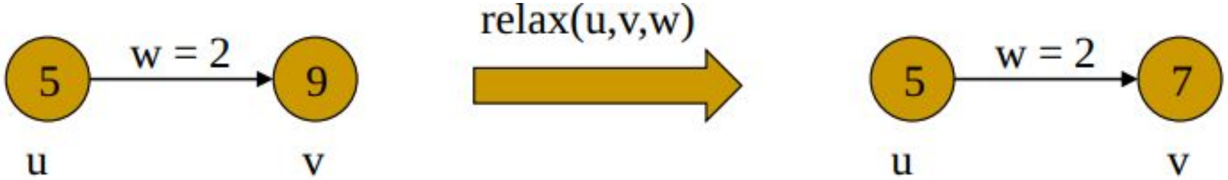
- Input: A graph with <u>weights</u> associated with each edge
- Output: The list of edges forming the shortest path
- Sample problems:
  - Find shortest path between two named vertices
  - Find shortest paths from a vertex s to all other vertices
  - Find shortest paths between all pairs of vertices
- Will calculate shortest distances.

Shortest Paths Definitions

- **δ(A, B)** is the shortest distance from vertex A to B.
- **w(A, B)** is the weight of the edge connecting A to B.
    - If there is no such edge, then w(A, B) = ∞.

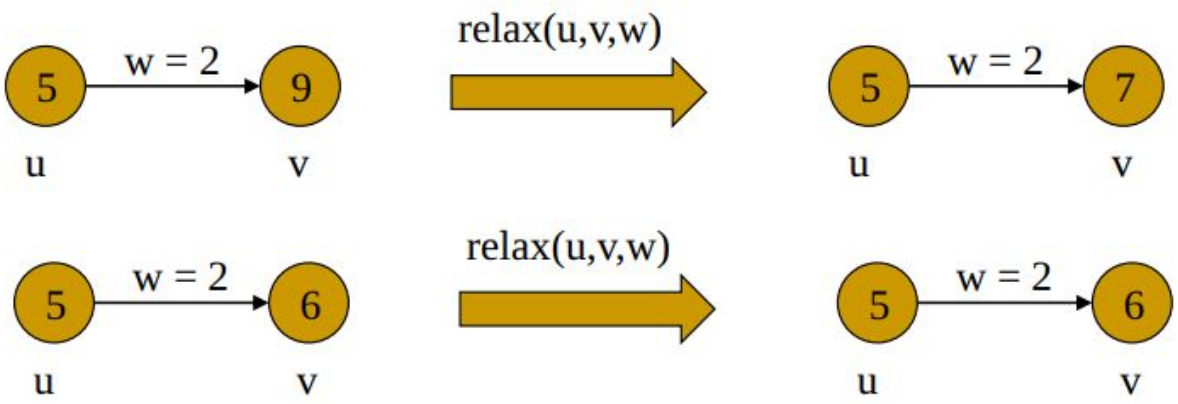
Single-Source Shortest Paths

- Given start vertex s, find the shortest path from s to all other vertices.
- Algorithm by Dijkstra
  - Maintain a set **S** of *visited vertices*. Also maintain distance array **D** of size n (# of vertices)
    - D[i] stores current estimate of distance d(s,i) between s and i
  - Initially, S is empty, and D[s] = 0
  - In each iteration (run this from n times)
    - Use D to do that
  - Update D for u's neighbors v by the following relax operation



Relax Operation

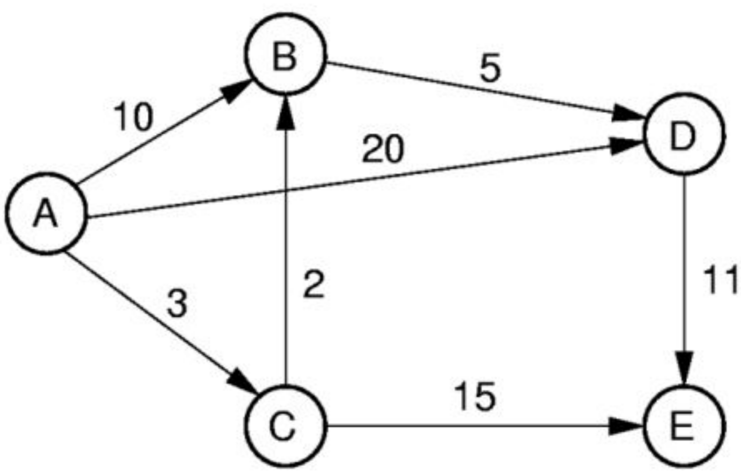
- Update D for u's neighbors v by the following relax operation
  - relax(u, v, w)
    - If d(s, v) > d(s, u) + w
      - d(s, v) = d(s, u) + w
      - prev(v) = u



Dijkstra’s Algorithm Example

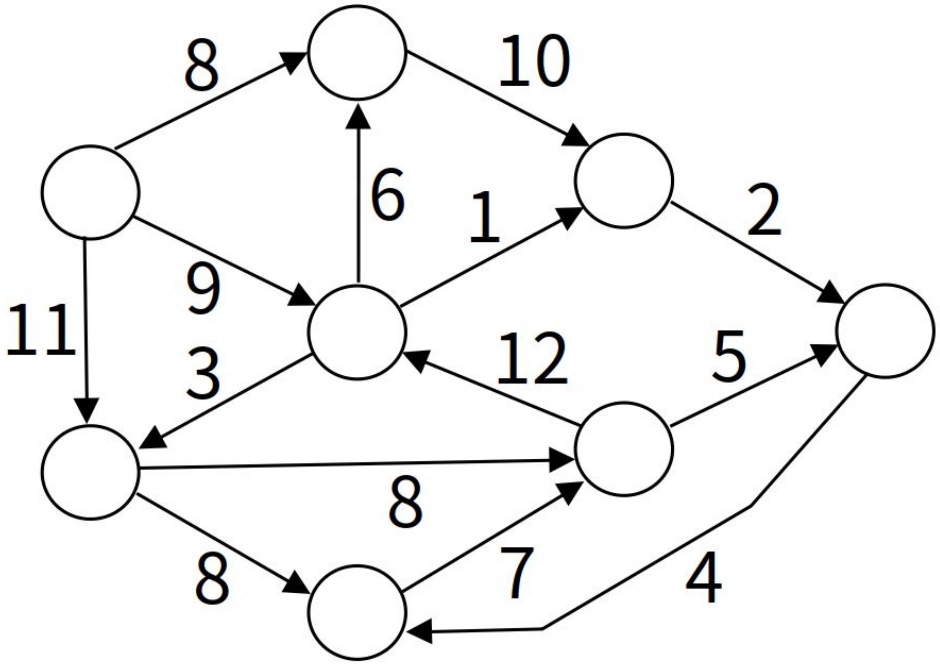
D vector

	A	B	C	D	E
Initial	0	∞	∞	∞	∞
Process A	0	10	3	20	∞
Process C	0	5	3	20	18
Process B	0	5	3	10	18
Process D	0	5	3	10	18
Process E	0	5	3	10	18



Start vertex: A

Dijkstra's Algorithm Example

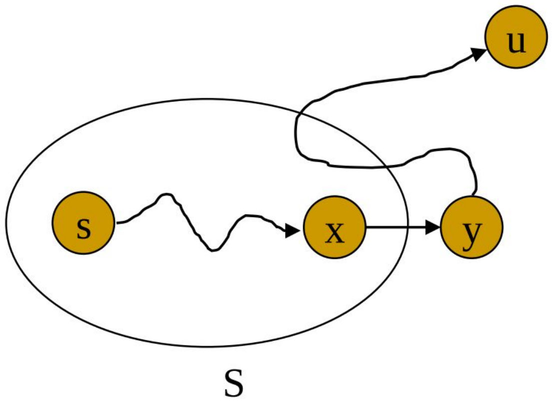


Correctness of Dijkstra

- Correctness
  - Claim: when u is added to S, $d(s, u) = \delta(s, u)$
    - This means that when u is added to S, $d(s, u)$ is the shortest distance, and we found the shortest path to u
  - Key idea (from Dynamic Programming)
    - Assume there is a node x in the shortest path P from s to y
    - Then, the path from s to x in P is a shortest path from s to x

Correctness of Dijkstra

- Claim: when u is added to S, $d(s, u) = \delta(s, u)$
  - (Proof by Induction)
    - (Base case) When $u = s$, the claim is trivially true
    - (Induction Hypothesis) Assume the claim is true for S
    - Now we add u to S. We need to show that $d(s, u) = \delta(s, u)$
    - Assume a shortest path s to u, and $x \to y$ are the first boundary vertices between S and V-S in the shortest path
    - It can be shown that $d(s, u) = \delta(s, u)$
      - $d(s,y) \le d(s,x) + w(x,y)$ (from relax on x)
      - $= \delta(s, x) + w(x,y)$ (from I.H.)
      - $= \delta(s, y)$ (key idea)
    - That implies y and u are the same
      - $d(s, y) = \delta(s, y) \le \delta(s, u) \le d(s, u)$
      - But, $d(s, u) \le d(s, y)$ since u is added to S
    - Thus, $d(s, u) = \delta(s, u)$



Dijkstra’s Implementation

```
function Dijkstra(G, r):
    S = {r}
    initialize an array D of size |V| with infinity
    initialize an array prev of size |V|

    D[r] = 0
    while S != V:
        u = minNode(V-S, D)
        add u to S
        for v in G.neighbors(v):
            if v is not in S and D[u] + w(u,v) < D[v]:
                D[v] = D[u] + w(u,v)
                prev[v] = u
```

Implementing minNode

- Issue: How to determine the next-closest vertex?
  - (i.e., implement minNode)
- Approach 1: Scan through the table of current distances.
  - Total cost of Dijkstra: $\Theta(|V|^2+|E|) = \Theta(|V|^2)$
- Approach 2: Store unprocessed vertices using a **min-heap** to implement a priority queue ordered by D value. <u>Must update priority queue for each edge.</u>
  - Cost: $\Theta((|V|+|E|)\log|V|)$

Questions?