# Shortest Path Problem

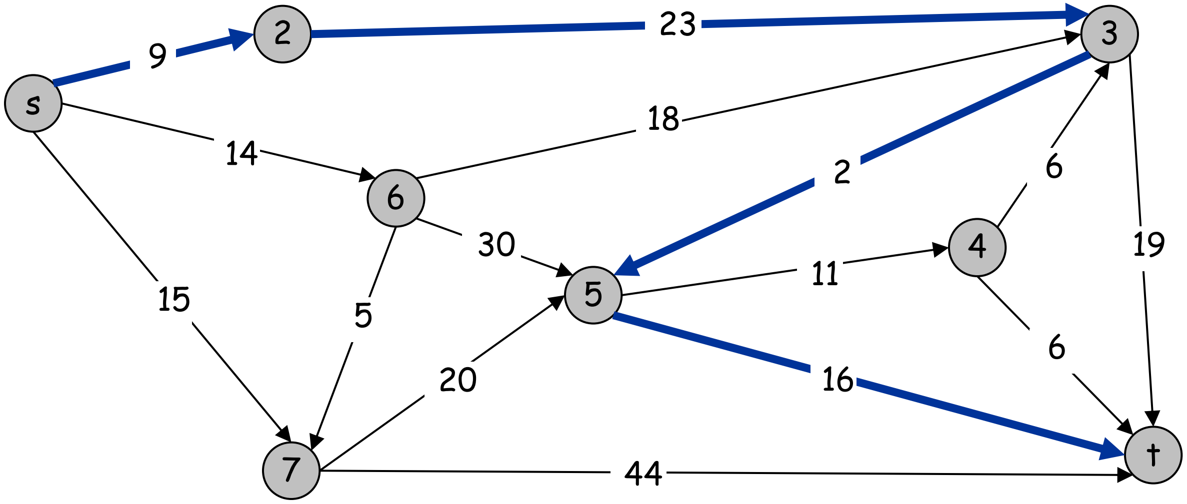
Shortest path in a network.

- Directed graph $G = (V, E)$.
- Source $s$, destination $t$.
- Length $\ell_e$ = length of edge e.

cost of path = sum of edge costs in path

Shortest path problem: find shortest (directed) path from $s$ to $t$.

Single source shortest path problem: find shortest directed path from $s$ to every node in V



Cost of path s-2-3-5-t
= 9 + 23 + 2 + 16
= 48.

# Dijkstra's Algorithm

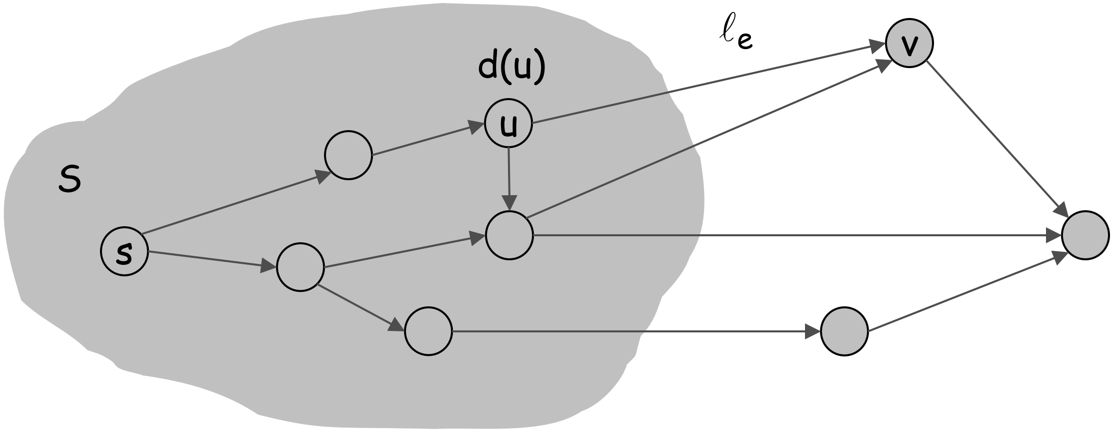
Dijkstra's algorithm.

- Maintain a set of explored nodes *S* for which we have determined the shortest path distance *d(u)* from *s* to *u*.
- Initialize $S = \{ s \}$, $d(s) = 0$.
- Repeatedly choose unexplored node v which minimizes

$$\pi(v) = \min_{e = (u,v) : u \in S} d(u) + l_e ,$$

shortest path to some u in explored part, followed by a single edge (u, v)

add *v* to *S*, and set $d(v) = \pi(v)$.



# Dijkstra's Algorithm

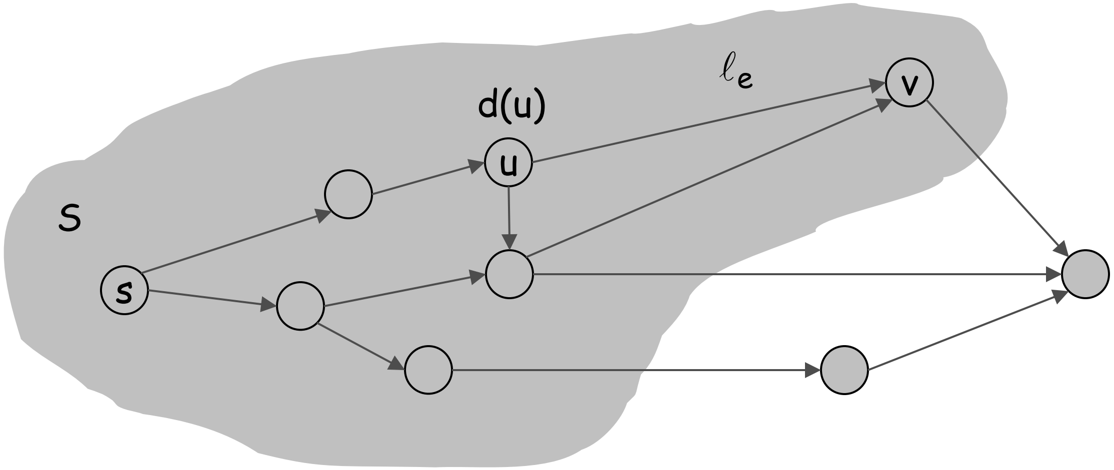
Dijkstra's algorithm.

- Maintain a set of explored nodes $S$ for which we have determined the shortest path distance $d(u)$ from $s$ to $u$.
- Initialize $S = \{ s \}$, $d(s) = 0$.
- Repeatedly choose unexplored node v which minimizes

$$\pi(v) = \min_{e = (u,v) : u \in S} d(u) + l_e,$$

add $v$ to $S$, and set $d(v) = \pi(v)$.

shortest path to some u in explored part, followed by a single edge (u, v)



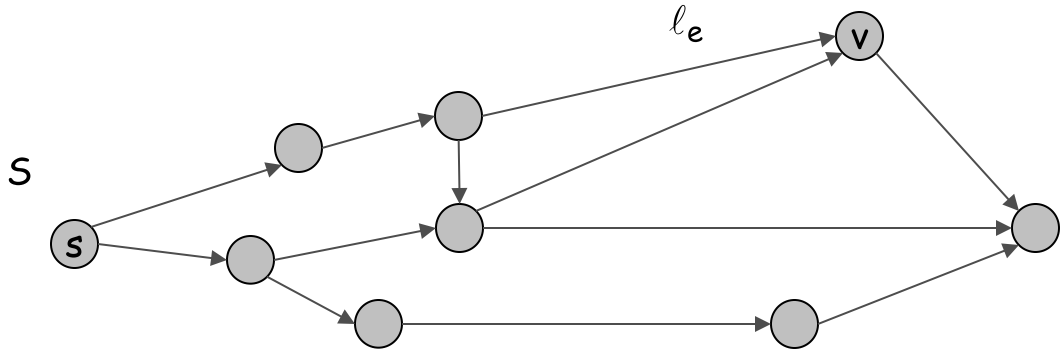
# Dijkstra's Algorithm

Dijkstra's algorithm.

- Dijkstra’s algorithm is a greedy algorithm.
  - What defines a “step” towards our goal?
  - What is our optimization criteria at each step?

- The result is a globally optimal solution to the SSSP problem!

- How to implement?



# Dijkstra's Algorithm: Implementation

For each unexplored node, explicitly maintain $\pi(v) = \min_{e = (u,v) : u \in S} d(u) + \ell_e$ .

- Next node to explore = node with minimum $\pi(v)$.
- When exploring v, for each incident edge e = (v, w), update

$$\pi(w) = \min \{ \pi(w), \pi(v) + \ell_e \}.$$

**Efficient implementation.** Maintain a priority queue of unexplored nodes, prioritized by $\pi(v)$.

| | Priority Queue | |
|---|---|---|
| PQ Operation | Array | Binary heap |
| Insert | n | log n |
| ExtractMin | n | log n |
| ChangeKey | 1 | log n |
| IsEmpty | 1 | 1 |
| Total | $n^2$ | m log n |

# Dijkstra's Algorithm Pseudocode

```
Dijkstra(G, s):
    let T ← ({s}, ∅)
    let PQ be an empty priority queue

    for each neighbor v of s, add edge (s,v) to PQ with priority l(e)
    while T doesn’t have all vertices of G and PQ is non-empty:
            repeat {
                    e ← PQ.removeMin() // skip edges with both ends in T
            } until PQ is empty or e=(u,v) for u∈T, v ∉ T
            if e=(u,v) for u∈T, v ∉ T
                    add e (and v) to T
                    for each neighbor w of v
                            add edge (v,w) to PQ with weight/key d(s,v) + l(v,w)
```

# Dijkstra's Algorithm: Proof of Correctness

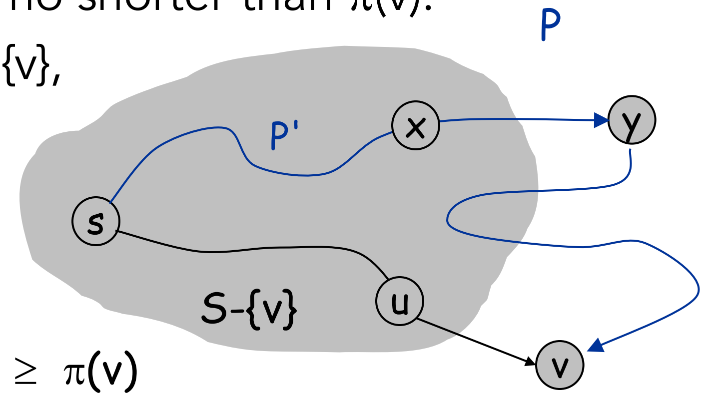
Invariant. For each node $u \in S$, $d(u)$ is the length of the shortest s-u path.

Pf. (by induction on $|S|$)

Base case: $|S| = 1$ and $d(s)=0$, which is true.

Inductive hypothesis: Assume true for $|S| = k \leq n$. Consider $|S|=k+1$

- Let v be last node added to S, and let u-v be the chosen edge.
- By inductive hypothesis, all nodes in S-{v} have correct shortest path dis.
- Claim: the s-u path plus (u, v) is an s-v path of shortest length $\pi(v)$.
  - Consider any s-v path P. We'll see that it's no shorter than $\pi(v)$.
  - Let x-y be the first edge in P that leaves S-{v}, and let P' be the subpath to x.



$$\ell(P) \underset{\substack{\uparrow\\ \text{nonnegative}\\ \text{weights}}}{\geq} \ell(P') + \ell(x,y) \underset{\substack{\uparrow\\ \text{inductive}\\ \text{hypothesis}}}{\geq} d(x) + \ell(x, y) \underset{\substack{\uparrow\\ \text{defn of } \pi(y)}}{\geq} \pi(y) \underset{\substack{\uparrow\\ \text{Dijkstra chose v}\\ \text{instead of y}}}{\geq} \pi(v)$$