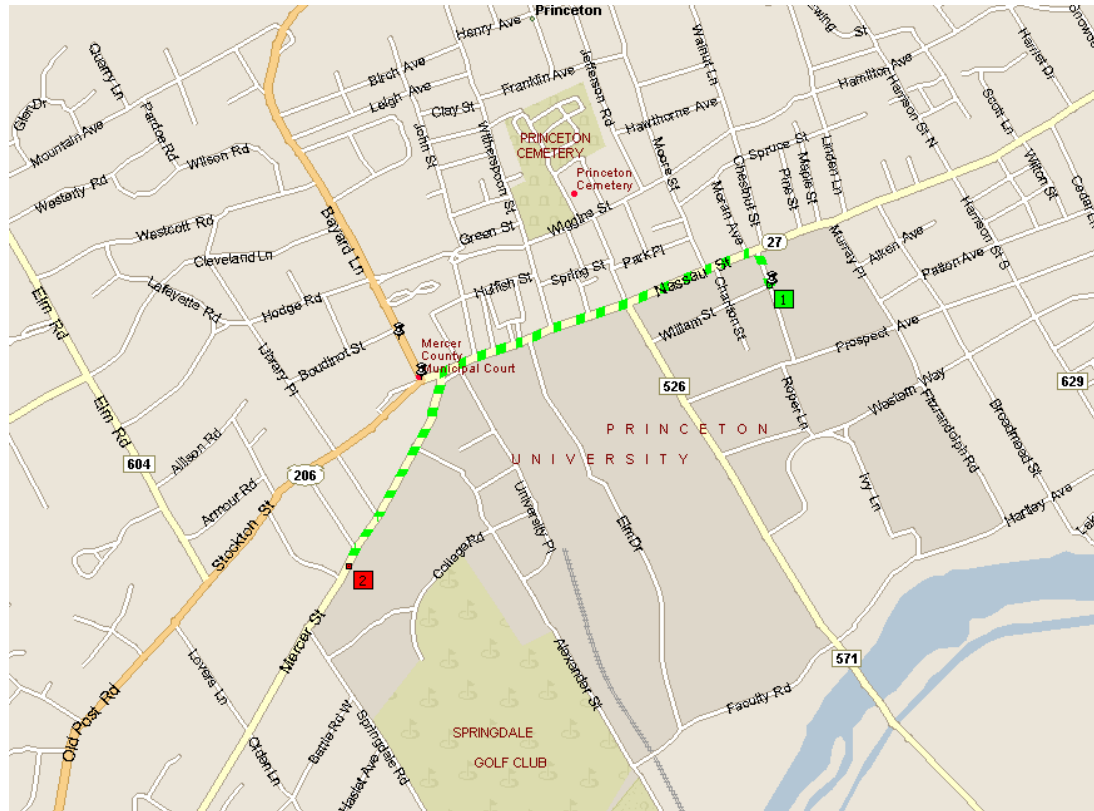


4.4 Shortest Paths in a Graph



shortest path from Princeton CS department to Einstein's house

Edsger W. Dijkstra

The question of whether computers can think is like the question of whether submarines can swim.

Do only what only you can do.

In their capacity as a tool, computers will be but a ripple on the surface of our culture. In their capacity as intellectual challenge, they are without precedent in the cultural history of mankind.

The use of COBOL cripples the mind; its teaching should, therefore, be regarded as a criminal offence.

APL is a mistake, carried through to perfection. It is the language of the future for the programming techniques of the past: it creates a new generation of coding bums.



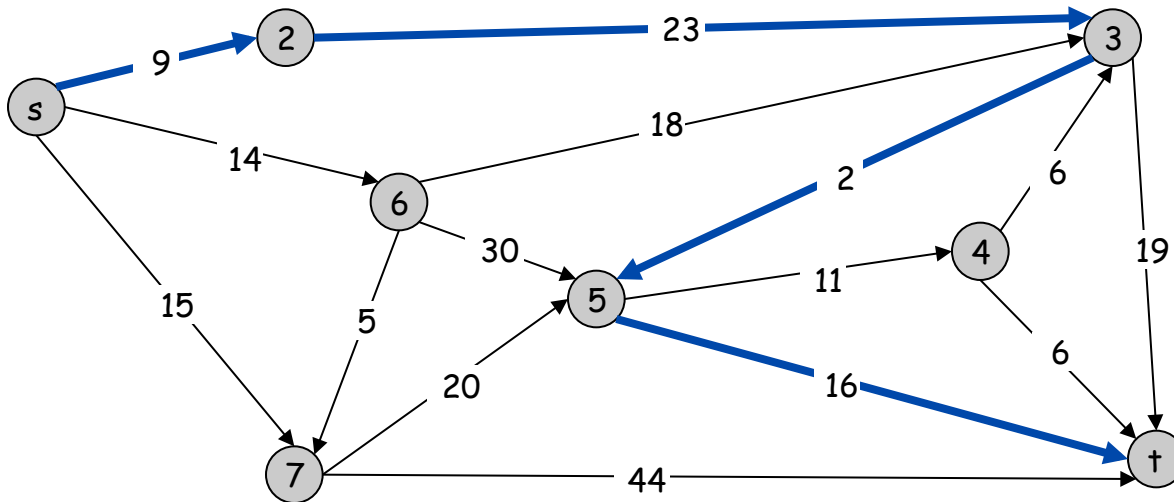
Shortest Path Problem

Shortest path network.

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

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

↑
cost of path = sum of edge costs in path



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

Dijkstra's Algorithm

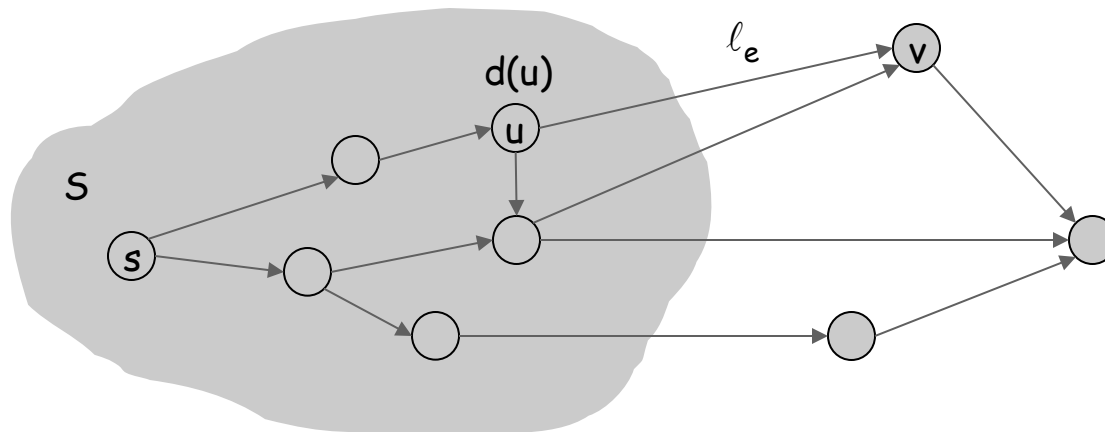
Dijkstra's algorithm.

- 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

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

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

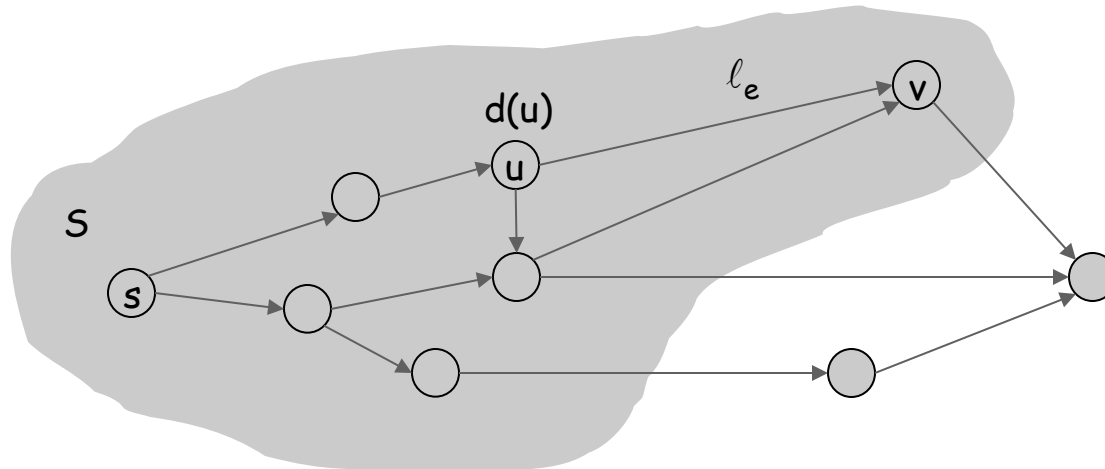
Dijkstra's algorithm.

- 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

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

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: Proof of Correctness

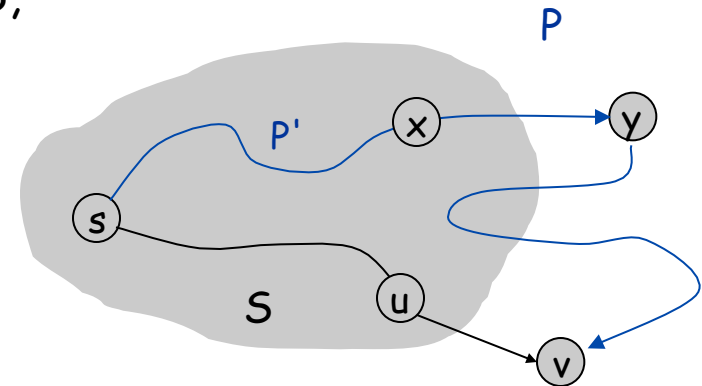
Invariant. For each node $u \in S$, $d(u)$ is the length of the shortest s - u path.

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

Base case: $|S| = 1$ is trivial.

Inductive hypothesis: Assume true for $|S| = k \geq 1$.

- Let v be next node added to S , and let u - v be the chosen edge.
- The shortest s - u path plus (u, v) is an s - v path of 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 , and let P' be the subpath to x .
- P is already too long as soon as it leaves S .



$$\ell(P) \geq \ell(P') + \ell(x, y) \geq d(x) + \ell(x, y) \geq \pi(y) \geq \pi(v)$$

\uparrow nonnegative weights \uparrow inductive hypothesis \uparrow defn of $\pi(y)$ \uparrow Dijkstra chose v instead of y

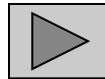
Dijkstra's Algorithm: Implementation

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 $\pi(v)$.



PQ Operation	Dijkstra	Array	Binary heap	d-way Heap	Fib heap [†]
Insert	n	n	$\log n$	$d \log_d n$	1
ExtractMin	n	n	$\log n$	$d \log_d n$	$\log n$
ChangeKey	m	1	$\log n$	$\log_d n$	1
IsEmpty	n	1	1	1	1
Total		n^2	$m \log n$	$m \log_{m/n} n$	$m + n \log n$

[†] Individual ops are amortized bounds