

4. GREEDY ALGORITHMS II

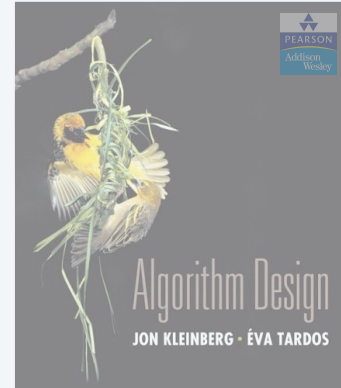
- ▶ Dijkstra's algorithm demo
- ▶ Dijkstra's algorithm demo (efficient implementation)

Lecture slides by Kevin Wayne

Copyright © 2005 Pearson-Addison Wesley

<http://www.cs.princeton.edu/~wayne/kleinberg-tardos>

Last updated on 11/15/17 10:10 AM



4. GREEDY ALGORITHMS II

- ▶ Dijkstra's algorithm demo
- ▶ Dijkstra's algorithm demo (efficient implementation)

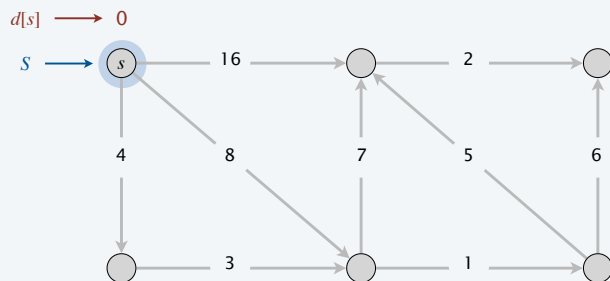
Dijkstra's algorithm demo

- Initialize $S \leftarrow \{s\}$ and $d[s] \leftarrow 0$.
- Repeatedly choose unexplored node $v \notin S$ which minimizes

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

the length of a shortest path from s to some node u in explored part S , followed by a single edge $e = (u, v)$

add v to S ; set $d[v] \leftarrow \pi(v)$ and $pred[v] \leftarrow \text{argmin}$.



3

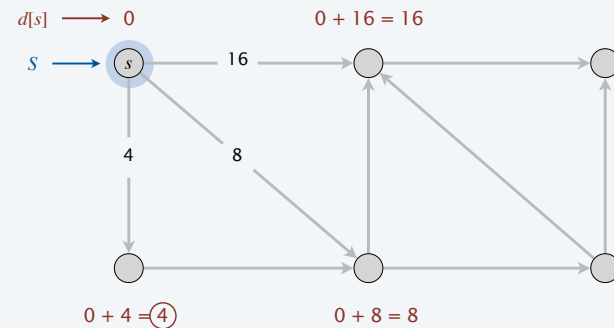
Dijkstra's algorithm demo

- Initialize $S \leftarrow \{s\}$ and $d[s] \leftarrow 0$.
- Repeatedly choose unexplored node $v \notin S$ which minimizes

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

the length of a shortest path from s to some node u in explored part S , followed by a single edge $e = (u, v)$

add v to S ; set $d[v] \leftarrow \pi(v)$ and $pred[v] \leftarrow \text{argmin}$.



4

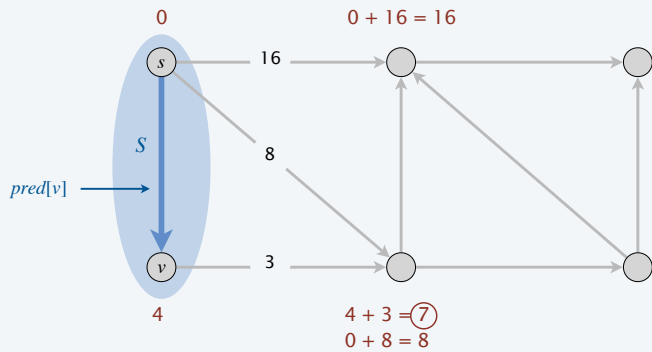
Dijkstra's algorithm demo

- Initialize $S \leftarrow \{s\}$ and $d[s] \leftarrow 0$.
- Repeatedly choose unexplored node $v \notin S$ which minimizes

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

the length of a shortest path from s to some node u in explored part S , followed by a single edge $e = (u, v)$

add v to S ; set $d[v] \leftarrow \pi(v)$ and $pred[v] \leftarrow \operatorname{argmin}$.



5

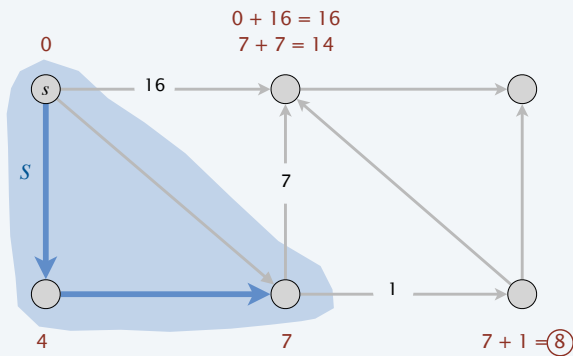
Dijkstra's algorithm demo

- Initialize $S \leftarrow \{s\}$ and $d[s] \leftarrow 0$.
- Repeatedly choose unexplored node $v \notin S$ which minimizes

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

the length of a shortest path from s to some node u in explored part S , followed by a single edge $e = (u, v)$

add v to S ; set $d[v] \leftarrow \pi(v)$ and $pred[v] \leftarrow \operatorname{argmin}$.



6

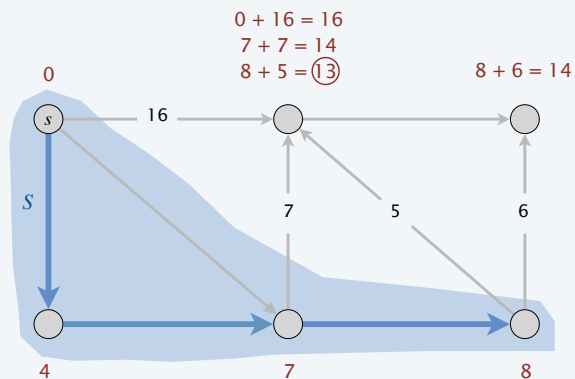
Dijkstra's algorithm demo

- Initialize $S \leftarrow \{s\}$ and $d[s] \leftarrow 0$.
- Repeatedly choose unexplored node $v \notin S$ which minimizes

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

the length of a shortest path from s to some node u in explored part S , followed by a single edge $e = (u, v)$

add v to S ; set $d[v] \leftarrow \pi(v)$ and $pred[v] \leftarrow \operatorname{argmin}$.



7

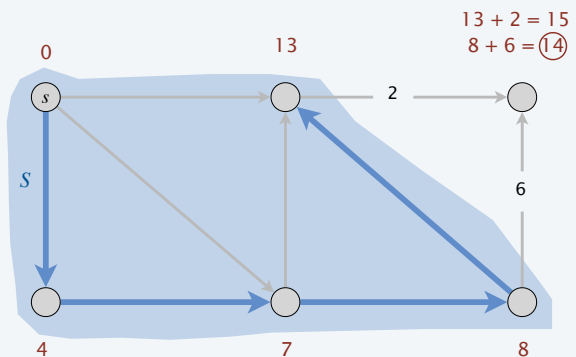
Dijkstra's algorithm demo

- Initialize $S \leftarrow \{s\}$ and $d[s] \leftarrow 0$.
- Repeatedly choose unexplored node $v \notin S$ which minimizes

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

the length of a shortest path from s to some node u in explored part S , followed by a single edge $e = (u, v)$

add v to S ; set $d[v] \leftarrow \pi(v)$ and $pred[v] \leftarrow \operatorname{argmin}$.



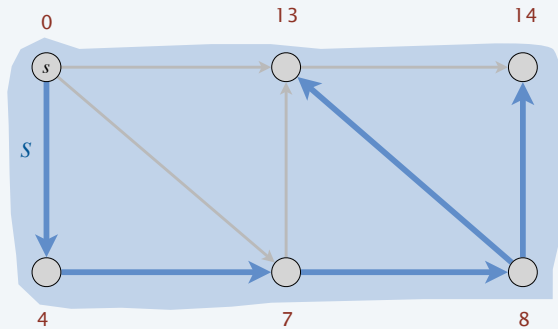
8

Dijkstra's algorithm demo

- Initialize $S \leftarrow \{s\}$ and $d[s] \leftarrow 0$.
- Repeatedly choose unexplored node $v \notin S$ which minimizes

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

add v to S ; set $d[v] \leftarrow \pi(v)$ and $pred[v] \leftarrow \operatorname{argmin}$.



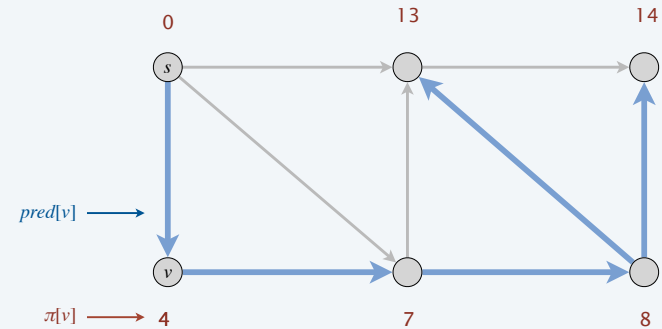
9

Dijkstra's algorithm demo

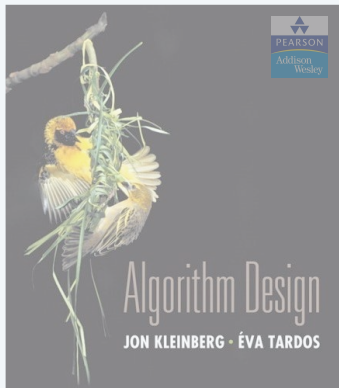
- Initialize $S \leftarrow \{s\}$ and $d[s] \leftarrow 0$.
- Repeatedly choose unexplored node $v \notin S$ which minimizes

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

add v to S ; set $d[v] \leftarrow \pi(v)$ and $pred[v] \leftarrow \operatorname{argmin}$.



10



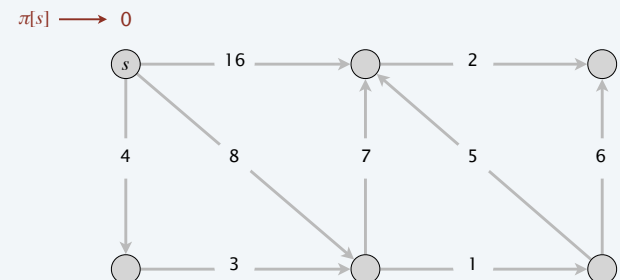
4. GREEDY ALGORITHMS II

- ▶ *Dijkstra's algorithm demo*
- ▶ *Dijkstra's algorithm demo (efficient implementation)*

Dijkstra's algorithm demo (efficient implementation)

Initialization.

- For all $v \neq s$: $\pi[v] \leftarrow \infty$.
- For all $v \neq s$: $pred[v] \leftarrow \text{null}$.
- $S \leftarrow \emptyset$ and $\pi[s] \leftarrow 0$.

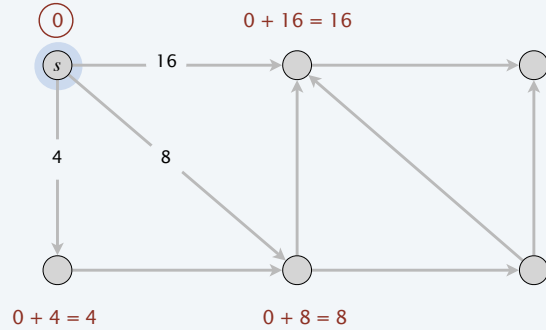


12

Dijkstra's algorithm demo (efficient implementation)

Basic step. Choose unexplored node $u \notin S$ with minimum $\pi[u]$.

- Add u to S .
- For each edge $e = (u, v)$ leaving u , if $\pi[v] > \pi[u] + \ell_e$ then:
 - $\pi[v] \leftarrow \pi[u] + \ell_e$
 - $pred[v] \leftarrow e$

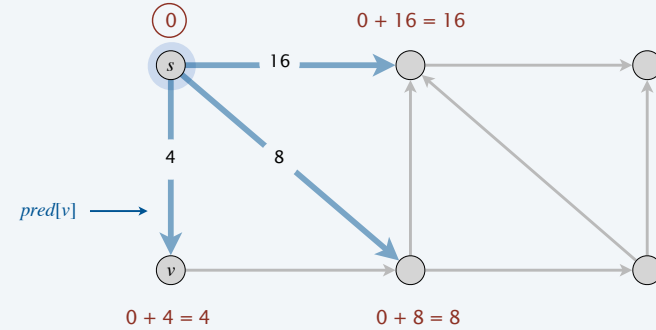


13

Dijkstra's algorithm demo (efficient implementation)

Basic step. Choose unexplored node $u \notin S$ with minimum $\pi[u]$.

- Add u to S .
- For each edge $e = (u, v)$ leaving u , if $\pi[v] > \pi[u] + \ell_e$ then:
 - $\pi[v] \leftarrow \pi[u] + \ell_e$
 - $pred[v] \leftarrow e$

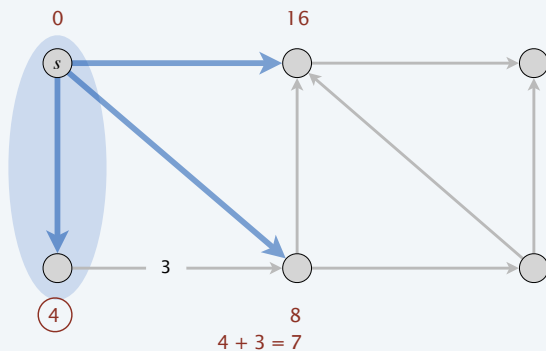


14

Dijkstra's algorithm demo (efficient implementation)

Basic step. Choose unexplored node $u \notin S$ with minimum $\pi[u]$.

- Add u to S .
- For each edge $e = (u, v)$ leaving u , if $\pi[v] > \pi[u] + \ell_e$ then:
 - $\pi[v] \leftarrow \pi[u] + \ell_e$
 - $pred[v] \leftarrow e$

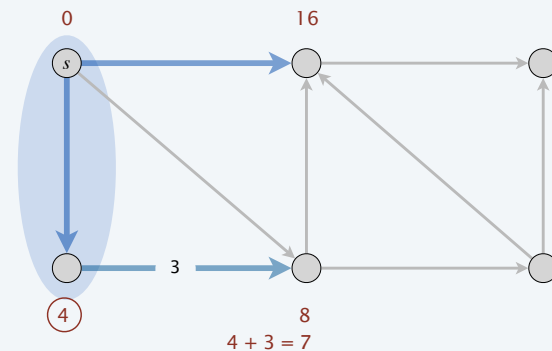


15

Dijkstra's algorithm demo (efficient implementation)

Basic step. Choose unexplored node $u \notin S$ with minimum $\pi[u]$.

- Add u to S .
- For each edge $e = (u, v)$ leaving u , if $\pi[v] > \pi[u] + \ell_e$ then:
 - $\pi[v] \leftarrow \pi[u] + \ell_e$
 - $pred[v] \leftarrow e$

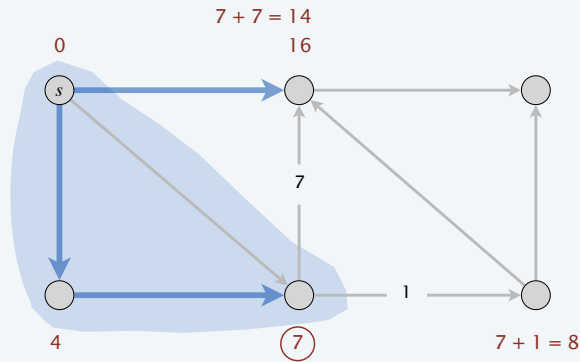


16

Dijkstra's algorithm demo (efficient implementation)

Basic step. Choose unexplored node $u \notin S$ with minimum $\pi[u]$.

- Add u to S .
- For each edge $e = (u, v)$ leaving u , if $\pi[v] > \pi[u] + \ell_e$ then:
 - $\pi[v] \leftarrow \pi[u] + \ell_e$
 - $pred[v] \leftarrow e$

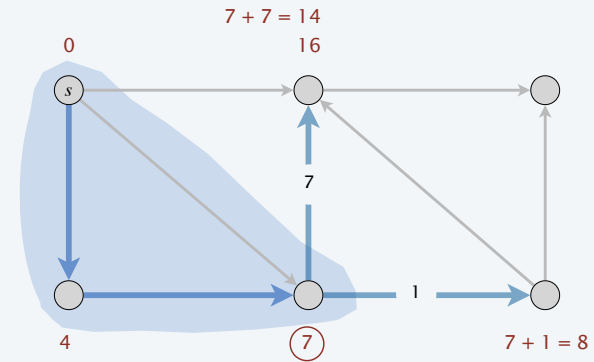


17

Dijkstra's algorithm demo (efficient implementation)

Basic step. Choose unexplored node $u \notin S$ with minimum $\pi[u]$.

- Add u to S .
- For each edge $e = (u, v)$ leaving u , if $\pi[v] > \pi[u] + \ell_e$ then:
 - $\pi[v] \leftarrow \pi[u] + \ell_e$
 - $pred[v] \leftarrow e$

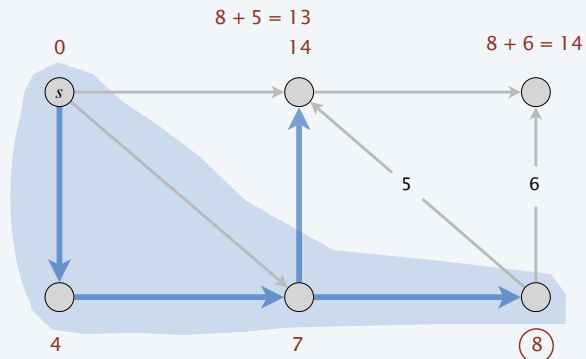


18

Dijkstra's algorithm demo (efficient implementation)

Basic step. Choose unexplored node $u \notin S$ with minimum $\pi[u]$.

- Add u to S .
- For each edge $e = (u, v)$ leaving u , if $\pi[v] > \pi[u] + \ell_e$ then:
 - $\pi[v] \leftarrow \pi[u] + \ell_e$
 - $pred[v] \leftarrow e$

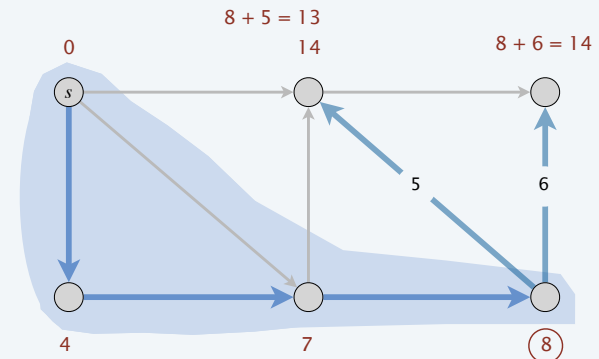


19

Dijkstra's algorithm demo (efficient implementation)

Basic step. Choose unexplored node $u \notin S$ with minimum $\pi[u]$.

- Add u to S .
- For each edge $e = (u, v)$ leaving u , if $\pi[v] > \pi[u] + \ell_e$ then:
 - $\pi[v] \leftarrow \pi[u] + \ell_e$
 - $pred[v] \leftarrow e$

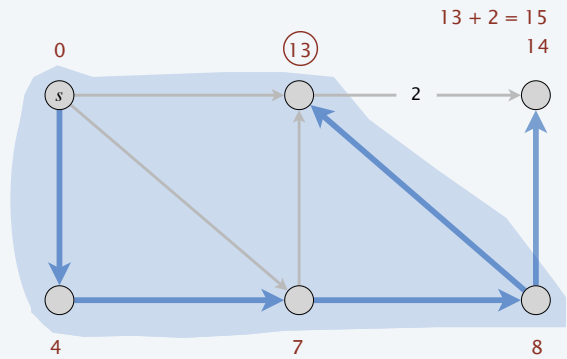


20

Dijkstra's algorithm demo (efficient implementation)

Basic step. Choose unexplored node $u \notin S$ with minimum $\pi[u]$.

- Add u to S .
- For each edge $e = (u, v)$ leaving u , if $\pi[v] > \pi[u] + \ell_e$ then:
 - $\pi[v] \leftarrow \pi[u] + \ell_e$
 - $pred[v] \leftarrow e$

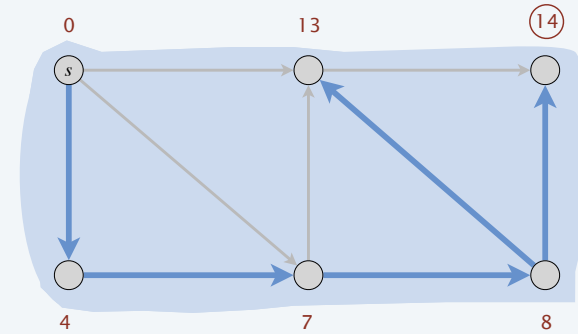


21

Dijkstra's algorithm demo (efficient implementation)

Basic step. Choose unexplored node $u \notin S$ with minimum $\pi[u]$.

- Add u to S .
- For each edge $e = (u, v)$ leaving u , if $\pi[v] > \pi[u] + \ell_e$ then:
 - $\pi[v] \leftarrow \pi[u] + \ell_e$
 - $pred[v] \leftarrow e$

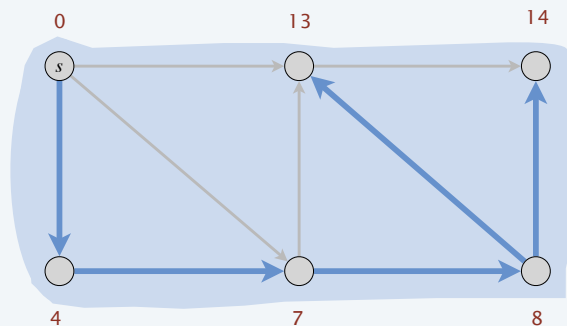


22

Dijkstra's algorithm demo (efficient implementation)

Basic step. Choose unexplored node $u \notin S$ with minimum $\pi[u]$.

- Add u to S .
- For each edge $e = (u, v)$ leaving u , if $\pi[v] > \pi[u] + \ell_e$ then:
 - $\pi[v] \leftarrow \pi[u] + \ell_e$
 - $pred[v] \leftarrow e$

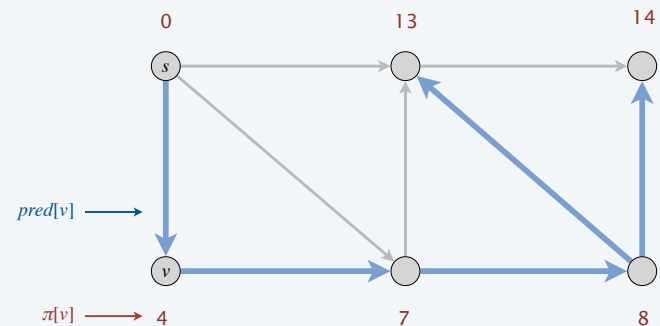


23

Dijkstra's algorithm demo (efficient implementation)

Termination.

- $\pi[v]$ = length of a shortest $s \rightarrow v$ path.
- $pred[v]$ = last edge on a shortest $s \rightarrow v$ path.



24