

Dijkstra's Algorithm

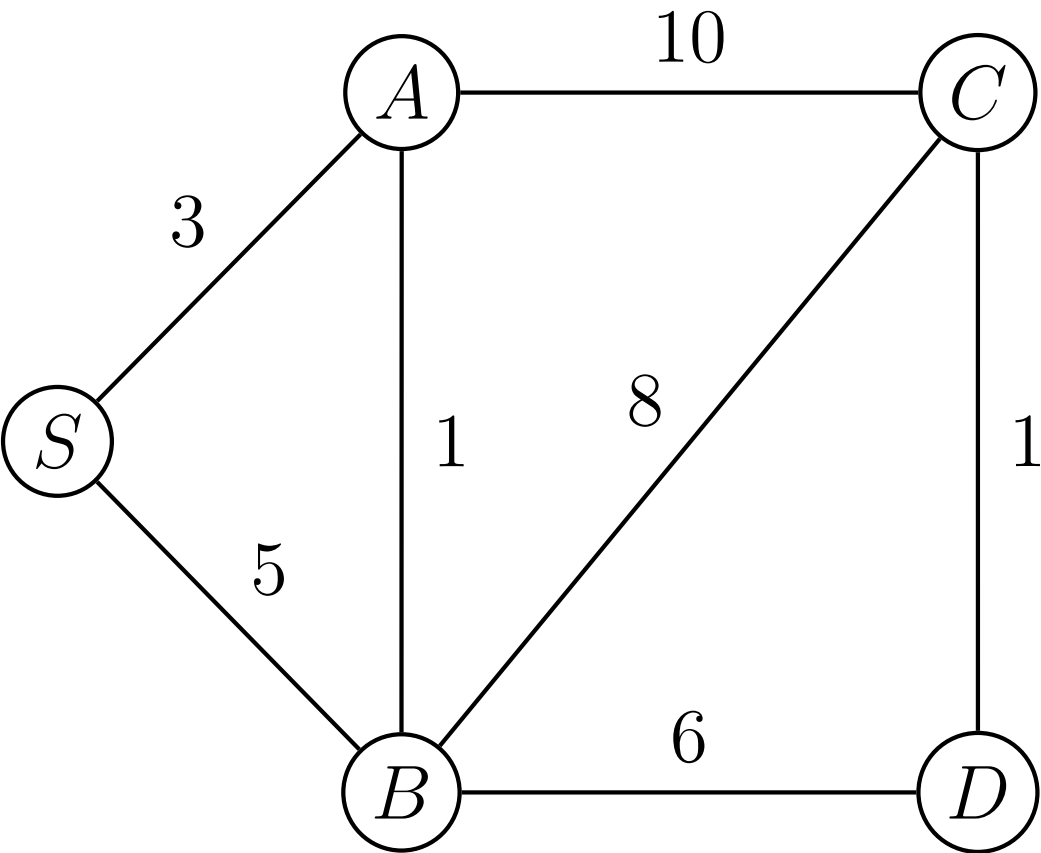
Dijkstra's Algorithm for Shortest Path

- $V_0, \dots, V_{n-1}$ are the vertices
- $d_{i,j}$ is the direct distance from V_i to V_j
- l_i is the temporary label on V_i
- l_i^* is the permanent label on V_i

Dijkstra's Algorithm

- Step 0. Label vertex V_0 with $l_0^* = 0$
 V_i get temp labels
 $l_i = d_{0,i}$ (direct arc)
 $l_i = \infty$ (no direct arc)
- Step 1. Pick $l_k = \min_i l_i$
Make l_k permanent $l_k^* \leftarrow l_k$
- Step 2. Update labels
 $l_i \leftarrow \min(l_i, l_k^* + d_{k,i})$

Dijkstra's Algorithm Example



$$\log_2 n = \lg n$$

$$\log_e n = \ln n$$

n	$=$	16	$=$	16
$n \lg n$	$=$	$16 \lg 16$	$=$	64
n^2	$=$	16^2	$=$	256
n^3	$=$	16^3	$=$	$4,096$
2^n	$=$	2^{16}	$=$	$65,536$
$n!$	$=$	$16!$	$=$	$20,922,789,888,000$

Big O

$f(n), g(n)$ nonnegative for $n \geq 0$

$$f(n) = O(g(n))$$

$$f(n) \in O(g(n))$$

$f(n)$ belongs to a class of functions
 $0 \leq f(n) \leq cg(n)$ for some constant c and for
all $n \geq n_0$.

$$\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = K$$

for some constant K which may be 0.

$$f(n) = \Omega(g(n))$$

$$f(n) \in \Omega(g(n))$$

$f(n)$ belongs to a class of functions
 $0 \leq cg(n) \leq f(n)$ for some constant $c > 0$
and for all $n \geq n_0$.

$$\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = K > 0$$

for some constant K which may be ∞ .

$$f(n) = \Theta(g(n))$$

$$f(n) \in \Theta(g(n))$$

$f(n)$ belongs to a class of functions

$0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n)$ for some constants $c_2 \geq c_1 > 0$ and for all $n \geq n_0$.

$$f(n) \in \Theta(g(n)) \Leftrightarrow \begin{cases} f(n) \in O(g(n)) \\ f(n) \in \Omega(g(n)) \end{cases}$$

Little o

$$f(n) = o(g(n))$$

$$f(n) \in o(g(n))$$

$$f(n) \in \text{a class of functions}$$

for any $c > 0$ there exists a constant $n_0 > 0$
such that $0 \leq f(n) < cg(n)$ for all $n \geq n_0$.

$$\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = 0$$

Prim's Algorithm

Prim's Algorithm to find a Minimum Spanning Tree

- $V_0, \dots, V_{n-1}$ are the vertices
- $d_{i,j}$ is the direct distance from V_i to V_j
- l_i is the temporary label on V_i
- l_i^* is the permanent label on V_i

Prim's Algorithm

- Step 0. Label vertex V_0 with $l_0^* = 0$
 V_i get temp labels
 $l_i = d_{0,i}$ (direct arc)
 $l_i = \infty$ (no direct arc)
- Step 1. Pick $l_k = \min_i l_i$
Make l_k permanent $l_k^* \leftarrow l_k$
- Step 2. Update labels $l_i \leftarrow \min(l_i, d_{k,i})$