

18.2.2

Shortest path via number of hops

Shortest Paths and Recursion

- 1 Compute the shortest path distance from s to t recursively?
- 2 What are the smaller sub-problems?

Lemma 18.2.

Let G be a directed graph with arbitrary edge lengths. If

$s = v_0 \rightarrow v_1 \rightarrow v_2 \rightarrow \dots \rightarrow v_k$ is a shortest path from s to v_k then for $1 \leq i < k$:

- 1 $s = v_0 \rightarrow v_1 \rightarrow v_2 \rightarrow \dots \rightarrow v_i$ is a shortest path from s to v_i

Sub-problem idea: paths of fewer hops/edges

Shortest Paths and Recursion

- 1 Compute the shortest path distance from s to t recursively?
- 2 What are the smaller sub-problems?

Lemma 18.2.

Let G be a directed graph with arbitrary edge lengths. If

$s = v_0 \rightarrow v_1 \rightarrow v_2 \rightarrow \dots \rightarrow v_k$ is a shortest path from s to v_k then for $1 \leq i < k$:

- 1 $s = v_0 \rightarrow v_1 \rightarrow v_2 \rightarrow \dots \rightarrow v_i$ is a shortest path from s to v_i

Sub-problem idea: paths of fewer hops/edges

Shortest Paths and Recursion

- 1 Compute the shortest path distance from s to t recursively?
- 2 What are the smaller sub-problems?

Lemma 18.2.

Let G be a directed graph with arbitrary edge lengths. If

$s = v_0 \rightarrow v_1 \rightarrow v_2 \rightarrow \dots \rightarrow v_k$ is a shortest path from s to v_k then for $1 \leq i < k$:

- 1 $s = v_0 \rightarrow v_1 \rightarrow v_2 \rightarrow \dots \rightarrow v_i$ is a shortest path from s to v_i

Sub-problem idea: paths of fewer hops/edges

Hop-based Recursion: Bellman-Ford Algorithm

Single-source problem: fix source s .

Assume that all nodes can be reached by s in G

Assume G has no negative-length cycle (for now).

$d(v, k)$: shortest walk length from s to v using at most k edges.

Note: $dist(s, v) = d(v, n - 1)$. Recursion for $d(v, k)$:

$$d(v, k) = \min \begin{cases} \min_{u \in V} (d(u, k - 1) + \ell(u, v)). \\ d(v, k - 1) \end{cases}$$

Base case: $d(s, 0) = 0$ and $d(v, 0) = \infty$ for all $v \neq s$.

Hop-based Recursion: Bellman-Ford Algorithm

Single-source problem: fix source s .

Assume that all nodes can be reached by s in G

Assume G has no negative-length cycle (for now).

$d(v, k)$: shortest walk length from s to v using at most k edges.

Note: $dist(s, v) = d(v, n - 1)$. Recursion for $d(v, k)$:

$$d(v, k) = \min \begin{cases} \min_{u \in V} (d(u, k - 1) + \ell(u, v)). \\ d(v, k - 1) \end{cases}$$

Base case: $d(s, 0) = 0$ and $d(v, 0) = \infty$ for all $v \neq s$.

Hop-based Recursion: Bellman-Ford Algorithm

Single-source problem: fix source s .

Assume that all nodes can be reached by s in G

Assume G has no negative-length cycle (for now).

$d(v, k)$: shortest walk length from s to v using at most k edges.

Note: $dist(s, v) = d(v, n - 1)$. Recursion for $d(v, k)$:

$$d(v, k) = \min \begin{cases} \min_{u \in V} (d(u, k - 1) + \ell(u, v)). \\ d(v, k - 1) \end{cases}$$

Base case: $d(s, 0) = 0$ and $d(v, 0) = \infty$ for all $v \neq s$.

Hop-based Recursion: Bellman-Ford Algorithm

Single-source problem: fix source s .

Assume that all nodes can be reached by s in G

Assume G has no negative-length cycle (for now).

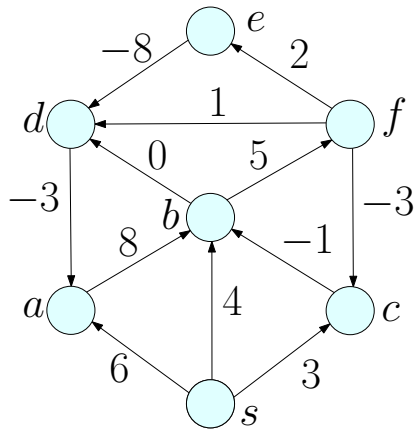
$d(v, k)$: shortest walk length from s to v using at most k edges.

Note: $dist(s, v) = d(v, n - 1)$. Recursion for $d(v, k)$:

$$d(v, k) = \min \begin{cases} \min_{u \in V} (d(u, k - 1) + \ell(u, v)). \\ d(v, k - 1) \end{cases}$$

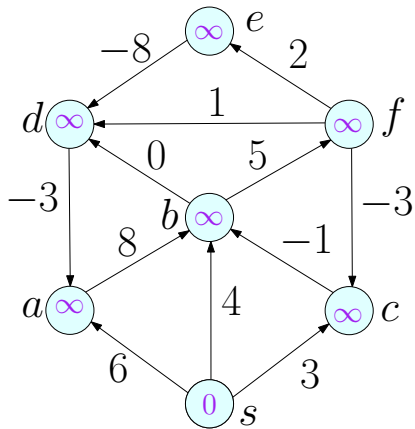
Base case: $d(s, 0) = 0$ and $d(v, 0) = \infty$ for all $v \neq s$.

Example



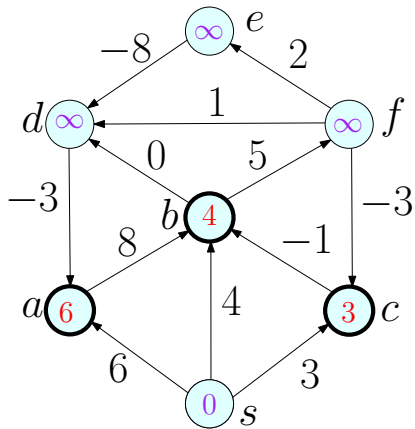
round	s	a	b	c	d	e	f

Example



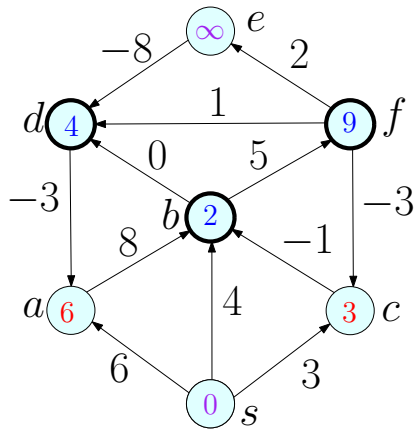
round	s	a	b	c	d	e	f
0	0	∞	∞	∞	∞	∞	∞

Example



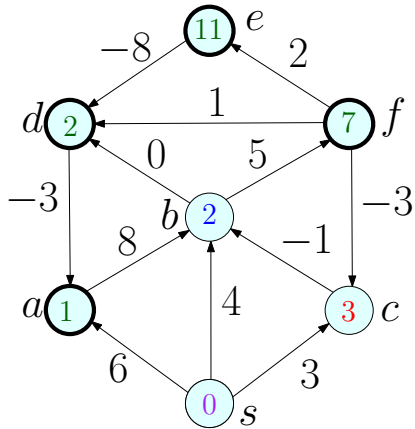
round	s	a	b	c	d	e	f
0	0	∞	∞	∞	∞	∞	∞
1	0	6	4	3	∞	∞	∞

Example



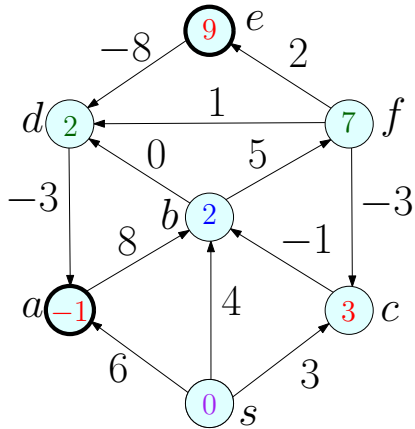
round	s	a	b	c	d	e	f
0	0	∞	∞	∞	∞	∞	∞
1	0	6	4	3	∞	∞	∞
2	0	6	2	3	4	∞	9

Example



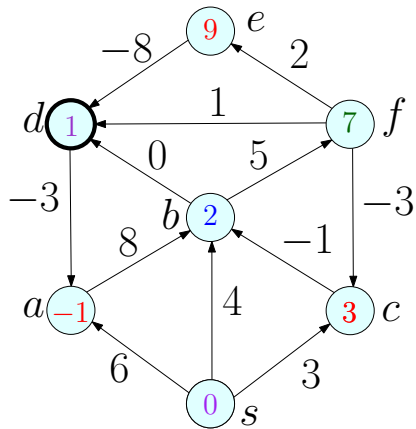
round	s	a	b	c	d	e	f
0	0	∞	∞	∞	∞	∞	∞
1	0	6	4	3	∞	∞	∞
2	0	6	2	3	4	∞	9
3	0	1	2	3	2	11	7

Example



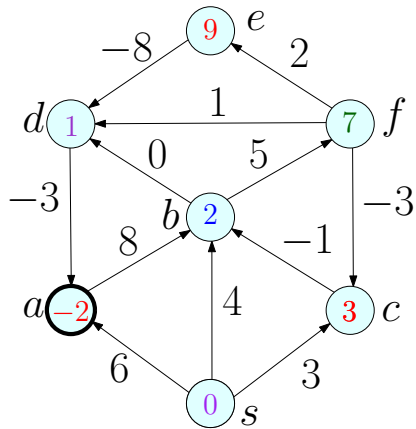
round	s	a	b	c	d	e	f
0	0	∞	∞	∞	∞	∞	∞
1	0	6	4	3	∞	∞	∞
2	0	6	2	3	4	∞	9
3	0	1	2	3	2	11	7
4	0	-1	2	3	2	9	7

Example



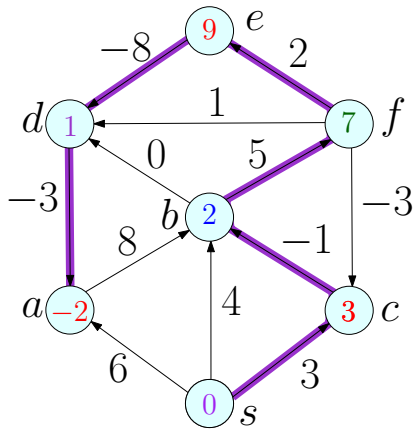
round	s	a	b	c	d	e	f
0	0	∞	∞	∞	∞	∞	∞
1	0	6	4	3	∞	∞	∞
2	0	6	2	3	4	∞	9
3	0	1	2	3	2	11	7
4	0	-1	2	3	2	9	7
5	0	-1	2	3	1	9	7

Example



round	s	a	b	c	d	e	f
0	0	∞	∞	∞	∞	∞	∞
1	0	6	4	3	∞	∞	∞
2	0	6	2	3	4	∞	9
3	0	1	2	3	2	11	7
4	0	-1	2	3	2	9	7
5	0	-1	2	3	1	9	7
6	0	-2	2	3	1	9	7

Example



round	s	a	b	c	d	e	f
0	0	∞	∞	∞	∞	∞	∞
1	0	6	4	3	∞	∞	∞
2	0	6	2	3	4	∞	9
3	0	1	2	3	2	11	7
4	0	-1	2	3	2	9	7
5	0	-1	2	3	1	9	7
6	0	-2	2	3	1	9	7

THE END

...

(for now)