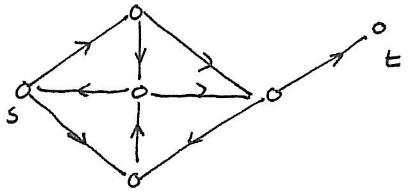
SHORTEST PATHS

$D = (V, A)$, directed graph

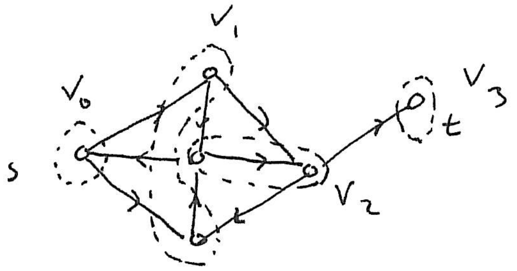


s-t walk

s-t path

dist(s,t) : min. length of any s-t path

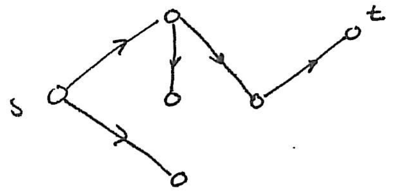
- Let $V_i = \{v \in V : \text{dist}(s,v) = i\}$



$$V_{i+1} = \{v \in V \setminus (V_0 \cup V_1 \cup \ldots \cup V_i) : (u,v) \in A \text{ for some } u \in V_i\}$$

Algorithm :
1. $V_0 := \{s\}$
2. $V_{i+1} = \{\ldots\}$
3. Stop if $V_{i+1} = \emptyset$, else 2

Running time : $O(|A|)$

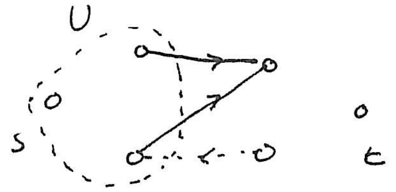


Rooted tree with root s

Gives shortest path to all vertices.

- Min-max relation

$A' \subseteq A$ is called an <u>s-t cut</u> if $A' = \delta^{out}(U)$ ~~for~~ some $U \subseteq V$ with $s \in U$ and $t \notin U$.



<u>Theorem</u>

The minimum length of an s-t path is equal to the maximum number of pairwise disjoint s-t cuts.

<u>Proof</u>

minimum $\leq$ maximum since each s-t path intersects each s-t cut

That min = max follows by considering s-t cuts $\delta^{out}(U_i)$, $i = 0, 1, \ldots, d-1$ where $d = dist(s,t)$ and $U_i := \{v \in V : dist(s,v) \leq i\}$

□

- **Arcs with non-negative lengths**

Length function : $l : A \to \mathbb{Q}_+$, i.e. $l(a) \geq 0$

Length of a walk P :

$$l(P) := \sum_{i=1}^{m} l(a)$$

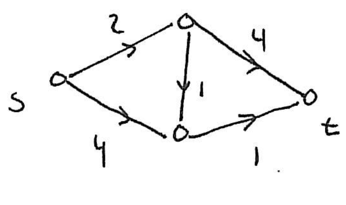
dist (s,t) : minimum length of any s-t path with regard to l (or $+\infty$ if no path exists)

Algorithm (Dijkstra, 1959)

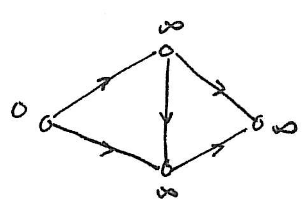
1. $U := V$, $f(s) := 0$, $f(v) := \infty$, $v \neq s$
2. find $u \in U$ minimizing $f(u)$
3. For each $a = (u,v) \in A$

   $f(v) := \min \{ f(v), f(u) + \cancel{l(u,v)}\ l(a) \}$
4. $U := U \setminus \{u\}$
5. If $U \neq \emptyset$ stop, else 2.

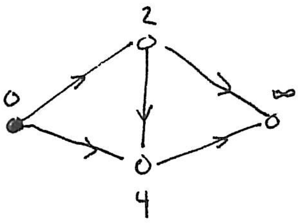
Ex



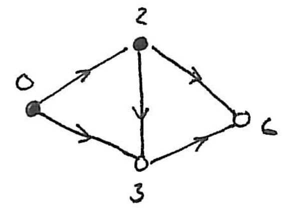
1.



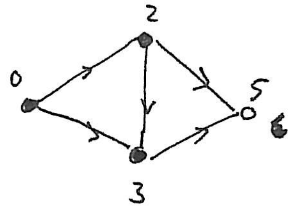
2.



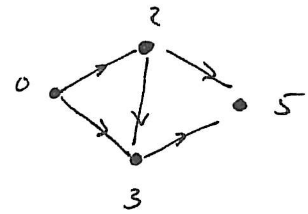
3.



4.



5.



Theorem

$f(v) = \text{dist}(s, v)$ upon completion

Proof

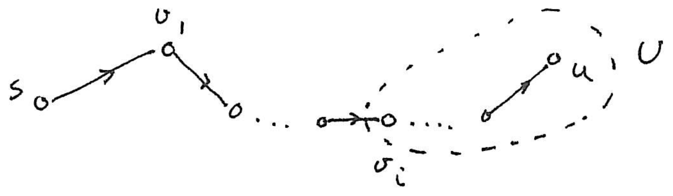
Let $d(v) := \text{dist}(s, v)$. We have that $f(v) \geq d(v)$ for all v throughout the iterations.

Claim: $f(v) = d(v)$ for each $v \in V \setminus U$ throughout the iterations

Trivial at the start of the algorithm.

Consider any iteration, suffices to show that $f(u) = d(u)$ for the $u \in U$ chosen in step 2.

Suppose $f(u) > d(u)$, and let $s = v_0, v_1, \ldots, v_k = u$ be a shortest s-u path. Let i be the smallest index with $v_i \in U$.



If $i = 0$, $\quad f(v_i) = f(s) = 0 = d(s) = d(v_i)$

If $i > 0$;

$$f(v_i) \leq f(v_{i-1}) + l(v_{i-1}, v_i) = d(v_{i-1}) + l(v_{i-1}, v_i)$$

$$= d(v_i)$$

This implies $f(v_i) \leq d(v_i) \leq d(u) < f(u)$,

contradicting the choice of u in Step 2.

$\square$

- Running time : $O(|V|^3)$

  - Improvement using heaps : $O(|A| \log |V|)$
    (Johnson, 1977)

  ~~- Improvement~~

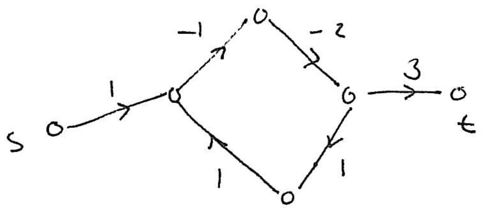
  - Fibonacci heaps : $O(|A| + |V| \log |V|)$

    (Fredman and Tarjan, 1984)

- **Arcs with arbitrary lengths**

  - Lengths of the arcs may take negative values

Ex



A shortest s-t walk may not always exist

- directed circuit of negative length

[Shortest path problem with arbitrary lengths is NP-hard, equivalent with longest path problem]

Theorem

If each directed circuit have non-negative length, and there exists at least one s-t walk, then there exists a shortest s-t walk, which is a path.

Algorithm (Bellman-Ford, 1958, 1956)

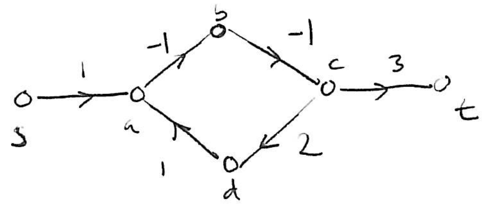
1. $f_0(s) := 0$, $f_0(v) := \infty$, $v \neq s$

2. $k = 1, \dots, n-1$

$$f_{k+1}(v) := \min \{ f_k(v), \min_{(u,v) \in A} (f_k(u) + l(u,v)) \}$$

- $f_n(v)$ gives the length of a shortest s-t walk

Ex



f	s	a	b	c	d	t
0	0	∞	∞	∞	∞	∞
1	0	1	∞	∞	∞	∞
2	0	1	0	∞	∞	∞
3	0	1	0	-1	∞	∞
4	0	1	0	-1	1	2
5	0	1	0	-1	1	2

$l(a,d) = -1$

⋮

5	0	0	0	-1	1	2
6	0	0	-1	-1	1	2

} neg. cycle

⋮

- Running time: $O(|V|\,|A|)$

Theorem

$f_k(v) = \min\{\, l(P) \mid P$ is an s-v walk traversing at most k arcs $\}$

Carollary

The algorithm finds a shortest s-t path if the graph has no negative-length directed circuit.

- Bellman-Ford is a dynamic programming algorithm