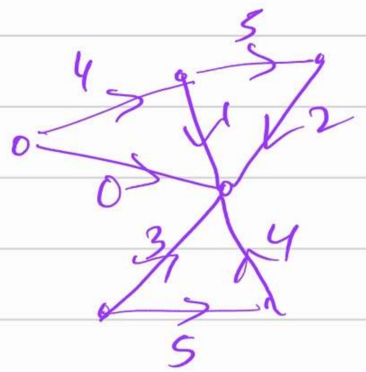
14/03/2022

Shortest path in weighted directed graphs.



$G = (V, E)$

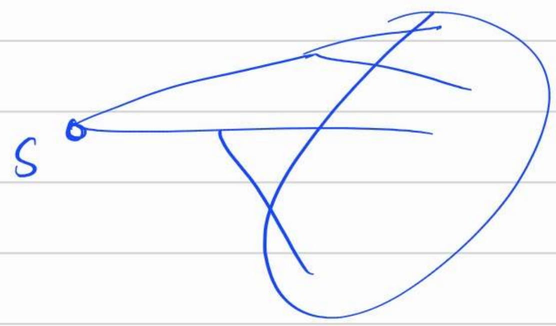
$w : E \to \mathbb{R}^{\geq 0}$

a vertex s.

Goal : to find shortest path to all vertices.

Later! We will handle negative wts. with the assumption that there are no negative cycles in the graph.

Single-Source-Shortest path



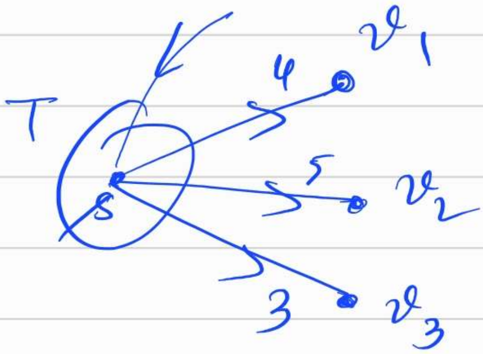
$v \in V$.

→ maintain the length of the shortest path a vertex.

→ $d(v)$ = length of a shortest path from $s \rightsquigarrow v$.

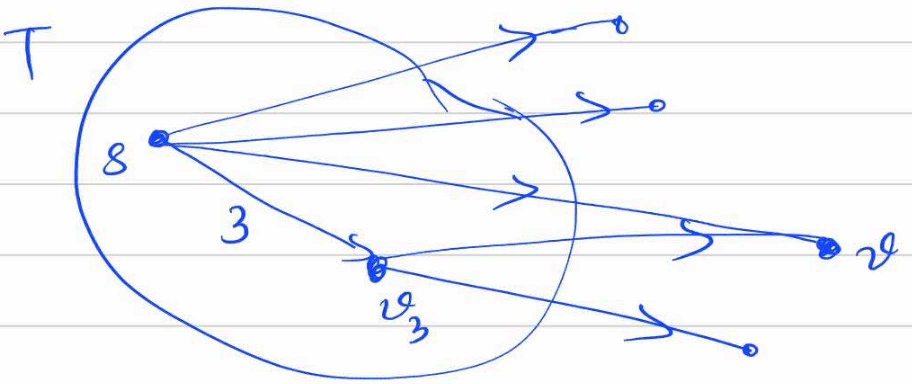
→ $d(s) = 0$.

Let T be the set of vertices to which we know the length of a shortest path.



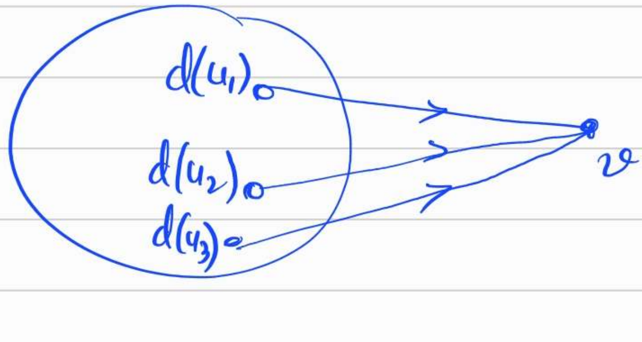
Consider the outgoing edges from T.

and add a minimum weight edge incident on $T = \{s\}$.



for each vertex $v \in V \setminus T$.

$$d(v) = \min_{\substack{u : (u,v) \in E \\ u \in T}} \left(d(u) + w((u,v)) \right)$$



Dijkstra's Algo.

For each vertex v, $d(v) := \infty$ if $v \neq s$
$d(s) = 0$.

Let T be the set of explored vertices.
Initialize $T := \{s\}$.

while $T \neq V$

Select a vertex $v \in V \setminus T$ s.t.
there exists an incoming edge incident on v from T and

$$d(v) = \min_{\substack{u \in T \\ (u,v) \in E(T, V \setminus T)}} \left(d(u) + w((u,v)) \right)$$

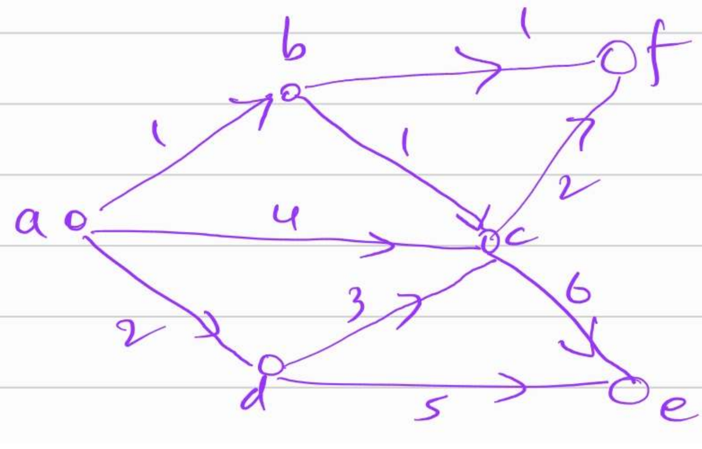
is minimal.

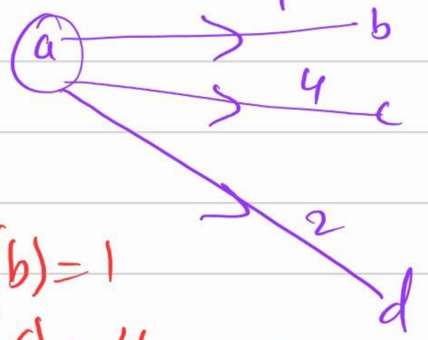
(In other words, you only consider edges in the cut $(T, V \setminus T)$)

Add v to T. ; pred(v) = the vertex u for which $d(v)$ attained the minimal value.

End while.

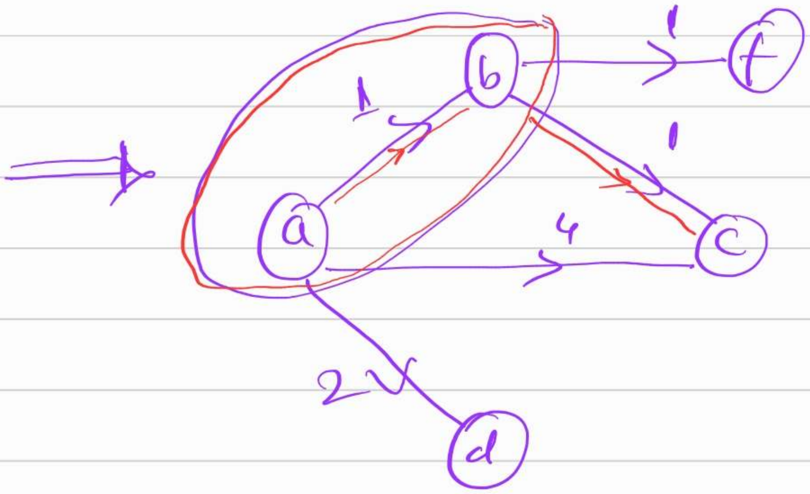
Example:





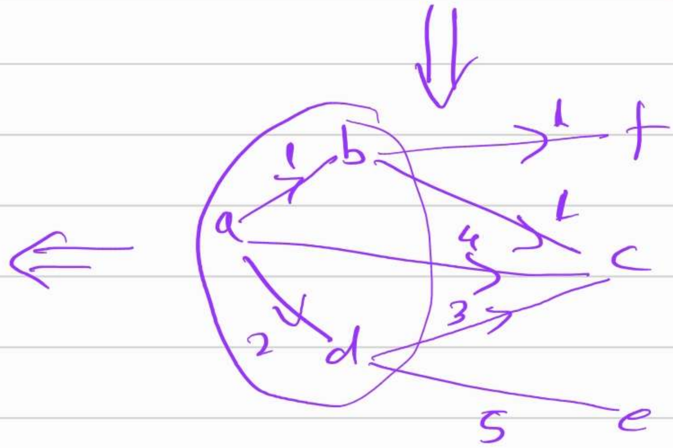
$d(b) = 1$
$d(c) = 4$
$d(d) = 2$

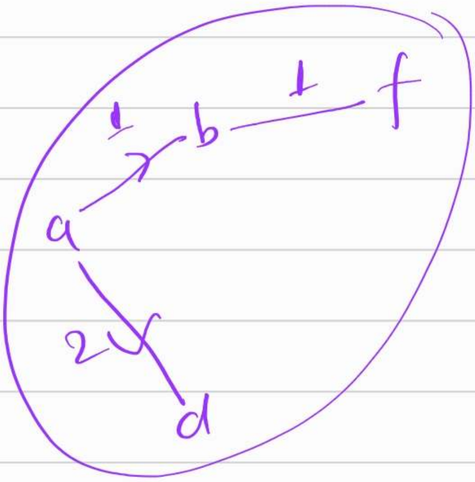
$d(e) = \infty$
$d(f) = \infty$

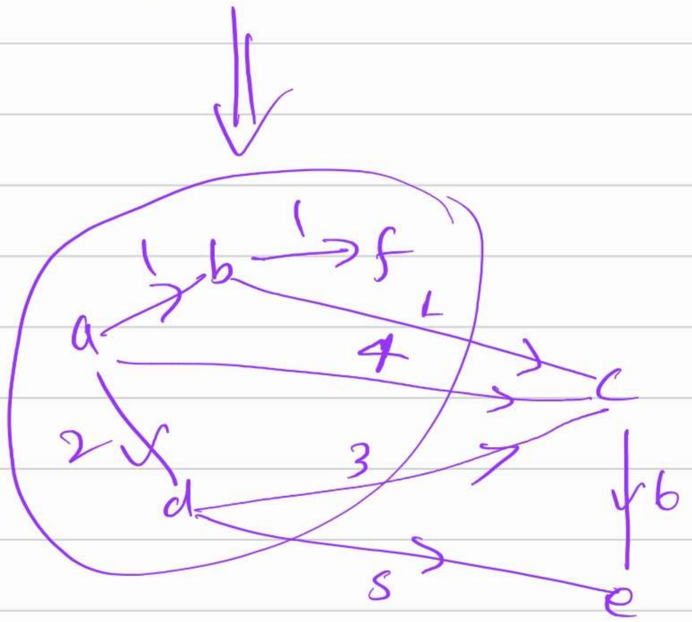


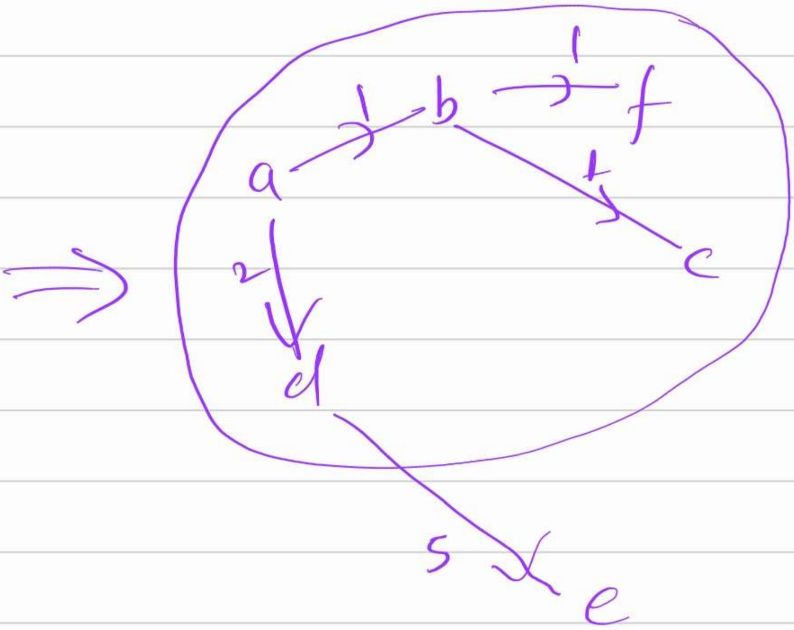
$d(f) = 2$, $\underline{d(c) = 2}$, $d(d) = 2$

$d(e) = \infty$









Implementation & running time:

- Computing $d(v)$ for each vertex which has an edge incident from T

$$v \in V \setminus T, \quad d(v) = \min_{\substack{u \in T \\ (u,v) \in E}} \left(d(u) + w((u,v)) \right)$$

- Choose a v s.t. $d(v)$ is minimal

Clearly the algorithm can be implemented in $O(m \cdot n)$ time

where $|V| = n$, $|E| = m$.

<u>Better implementation:</u> $O(m \log n)$.

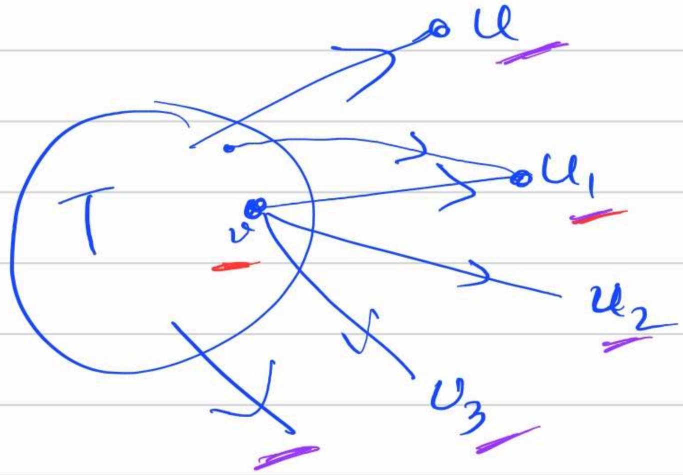
- maintain $v \in V \setminus T$

in a priority queue with

$Key(v) = d(v)$.

▷ ExtractMin $\leftarrow$ $n-1$

▷

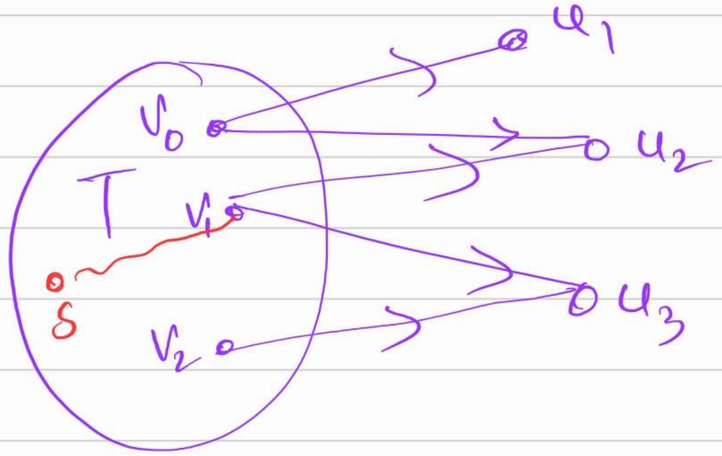


if $d(u_1) > d(v) + w((v, u_1))$

then $d(u_1) = d(v) + w((v, u_1))$

otherwise.

$d(u_1)$ remains the same.



$$d(u_3) = \min \left\{ \begin{array}{l} d(v_1) + w((v_1, u_3)), \\ d(v_2) + w((v_2, u_3)) \end{array} \right\}$$

$$d(u_2) = \min \left\{ \begin{array}{l} d(v_0) + w(v_0, u_2), \\ d(v_1) + w(v_1, u_2) \end{array} \right\}$$

- if

$$d(u_1) > d(v) + w((v, u_1))$$

then $d(u_1) = d(v) + w((v, u_1))$

then decreasekey$(u_1, d(u_1))$.

Decreasekey is called at most m times.

- Total running time:

$$O(n) + O(m \log n)$$