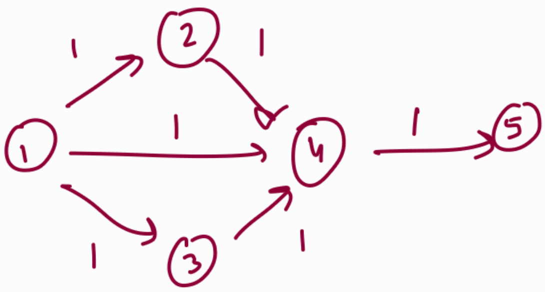
• Shortest path.

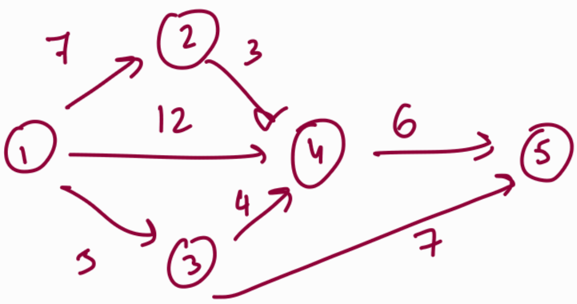
Tut wed 2pm.
Tut this Sat 2pm.



(1, 4).

If all wts. are equal, then BFS gives the shortest path.

Now, edges have wts. BFS no longer gives the shortest path

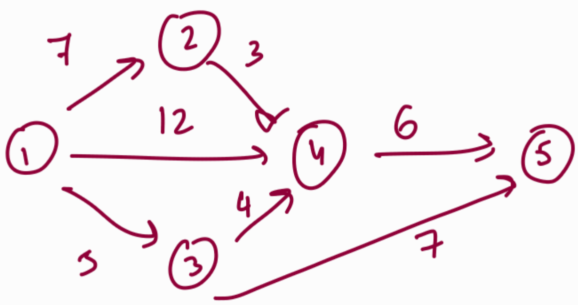


• Shortest paths in DAGs

$G = (V, E)$, $wt : E \to \mathbb{R}$

If P is a path with edges $e_1, \dots, e_k$ in it, then

$wt(P) = wt(e_1) + wt(e_2) + \dots + wt(e_k)$.



Q^n: What is the shortest path from 1 to 5?

• Topological sort.

1) Place the nodes in topological order.

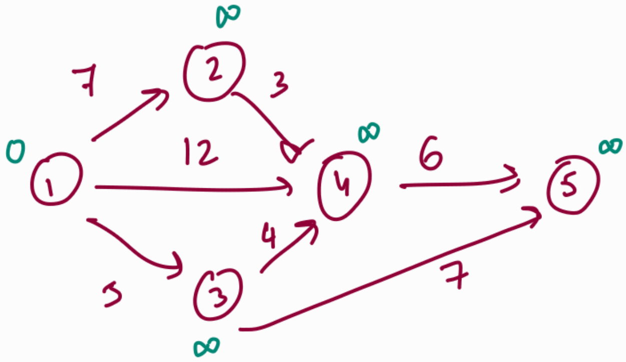
2) $\forall v \in V \setminus \{s\}$, $dist(s, t) \leftarrow \infty$

3) Visited $\leftarrow \{s\}$

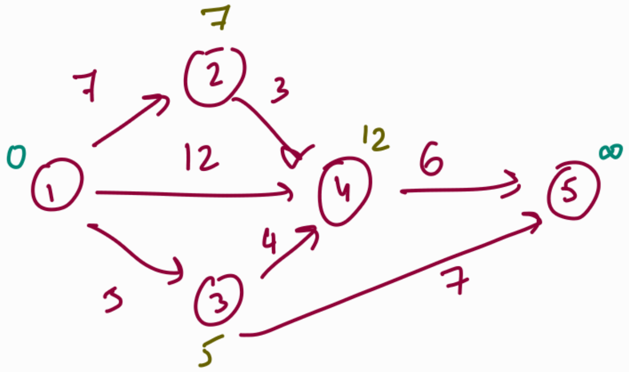
4) $\boxed{\text{while visited} \neq V:}$ → Not the exact condition. Have to modify.

for all $v \in N(\text{visited})$:

$$\text{dist}(s,v) = \min\left\{ \{ \text{dist}(s,u) + \text{wt}(u,v) \mid (u,v) \in E,\ u \in \text{visited} \} \cup \{ \text{dist}(s,t) \} \right\}$$



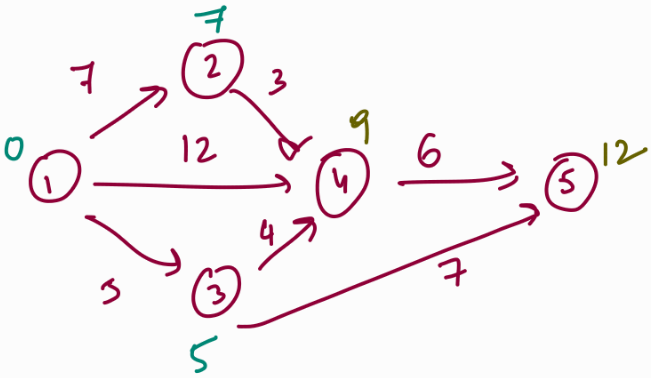
→



$N_{out}(1,2,3,4) = \{2,3,4,5\}$.

$$\text{dist}(s,4) = \min\{ \overset{12}{\text{dist}(1,4)},\ \underset{10}{\text{dist}(1,2)+\text{wt}(2,4)},\ \underset{9}{\text{dist}(1,3)+\text{wt}(3,4)} \}$$

$$\text{dist}(s,5) = \min\{ \text{dist}(1,4)+6,\ \text{dist}(1,3)+7,\ \infty \}$$



Direct acyclicness guarantees that the algo ends at some point of time.

→ Even if there are -ve wts., it'll still work because of the acyclicness.

→ Assumed DAG because if it is cyclic, then a cyclic dependency forms in recursion.

Eg: If dist(1,4) req. calc. of dist(1,5) and dist(1,5) req. dist(1,4). Basically a cycle. Unresolvable recursion.

See notes for precise code. Above code not precise. (Recursion better instead of while).

- Single source graphs with no negative edges (Can have cycles).
(Works with indirected graphs as well)

- Djikstra's algo :

Try it out using eg.

Visited $\leftarrow \{s\}$, $d[s] = 0$

For every $v \in V \setminus \{s\}$

$d(v) \leftarrow \infty$

while visited $\neq V$:

For all $v \in (V \setminus \text{visited}) \cap N(\text{visited})$:

$$d'[v] = \min_{(u,v) \in E,\ u \in \text{visited}} \{ d(u) + wt.(u,v) \}$$

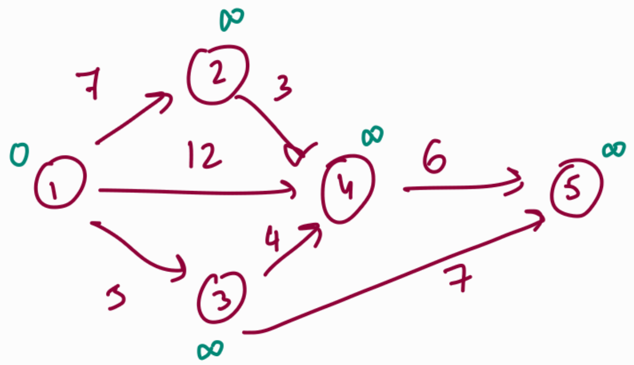
select $v \in (V \setminus \text{visited}) \cap (N(\text{visited}))$ s.t

$d'(v) = \min \{ d'[v] \mid v \in A \}$

set $d[v] \leftarrow d'[v]$

add v to visited.

The while loop runs $(n-1)$ times



→ Does this always give the correct output of shortest path?

(Req. proof of correctedness).

Observations:

1. Visited set grows by 1 vertex in each iteration, but elements once added are not disturbed in the later stage of the algorithm.

2. Shortest dist. once computed are not updated ever again. ($d[u]$ is never updated once it attains a value)

Correctness lemma :

Consider the set visited at an arbitrary point of time in the algo's execution. For all $u \in$ visited , $d[u]$ is the shortest dist. from $s \rightsquigarrow u$.

(Proof in the next doc.)

Dry run with examples.