Shortest Paths

Input: Directed graph $G = (V, E)$

$s \in V$

wt on edges (positive)

Output: Shortest path tree rooted at s.

Dijkstra's alg.

1. **for** each $u \in V$ **do**

$d[u] \leftarrow \infty$

$\pi[u] \leftarrow NIL$

$d[s] \leftarrow 0$

$S \leftarrow \phi$ // $V \setminus S = V$

while $S \neq V$ **do** // assume that
// all vertices are
// reachable from S

$u \leftarrow$ vertex in $V \setminus S$ with the smallest $d[\cdot]$

$S \leftarrow S \cup \{u\}$

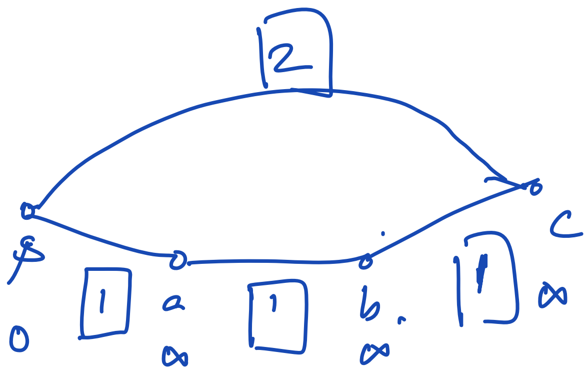
for each $v \in N(u) \cap V \setminus S$ **do**

$d[v] > w_{uv}$ **if** $d[v] > d[u] + w_{uv}$ **then**

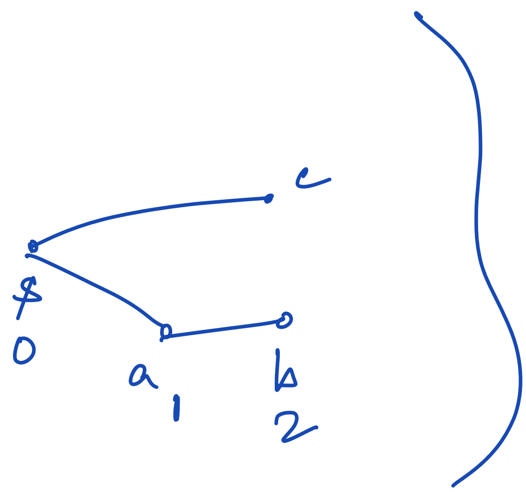
$d[v] \leftarrow w_{uv}$ $d[v] \leftarrow d[u] + w_{uv}$

$\pi[v] \leftarrow u$

Example



S



Data structure to implement Dijkstra's alg efficiently : Heaps.

Proof of Correctness

Theorem: Dijkstra's alg. will yield correct shortest paths from s to every vertex in G.

Proof: Induction on $|S|$.

IH: Let $k \geq 1$ be an arbitrary but particular integer. Assume that the claim holds when $n = k$.

That is, when $|S| = k$, all

vertices in S have their shortest paths computed correctly.

BC: $|S| = 1$.

$s \in S$.

Our alg. sets $d[s] = 0$, which is the correct answer.

IS: We want to prove the claim when $|S| = k+1$. Let v be the $(k+1)^{th}$ vertex brought into S.

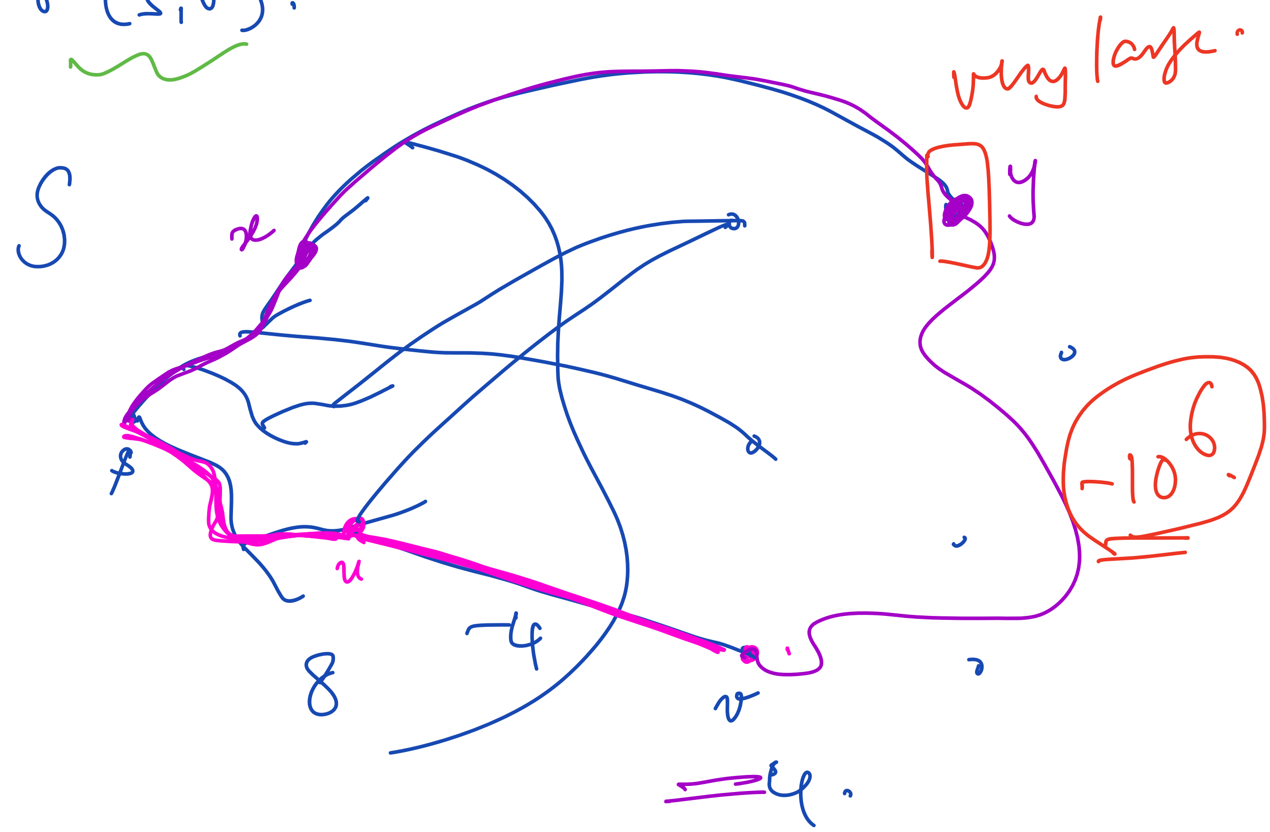
Let $\pi(v) = u$.

Assume for contradiction that

$\boxed{d[v] = d[u] + w_{uv}}$ is not the wt.

of the shortest path from s to v,

$\delta(s,v)$.



Instead, the actual shortest path from

s to v is given by

$P: \; s \rightsquigarrow x \longrightarrow y \rightsquigarrow v$

where x is the last vertex in P that

is in S.

$w(P) < d[v]$

$\therefore \; \underbrace{l(s,x) + w_{xy} + l(y,v)} < d[v]$

→ True only if edge wts are positive

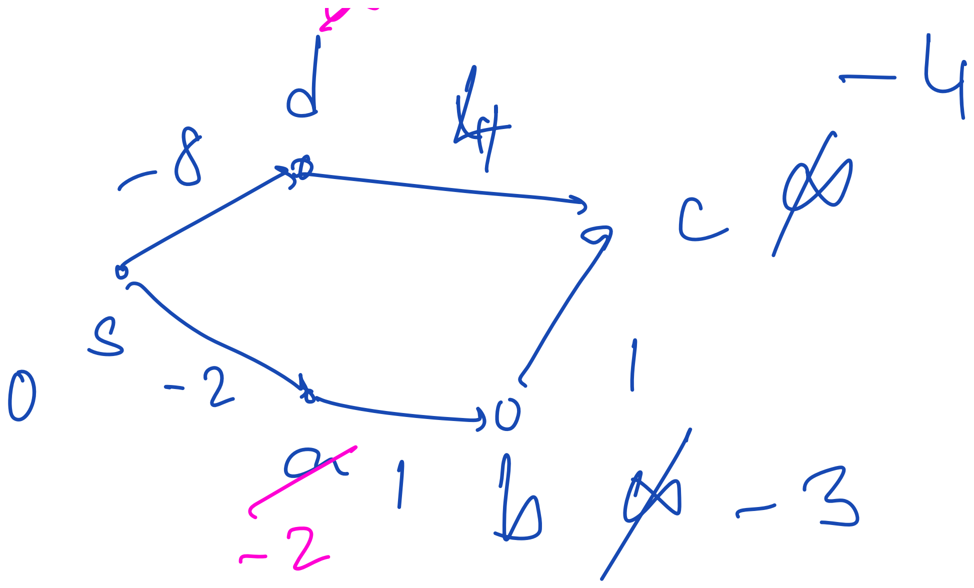
$\therefore \; l(s,x) + w_{xy} < d[v]$

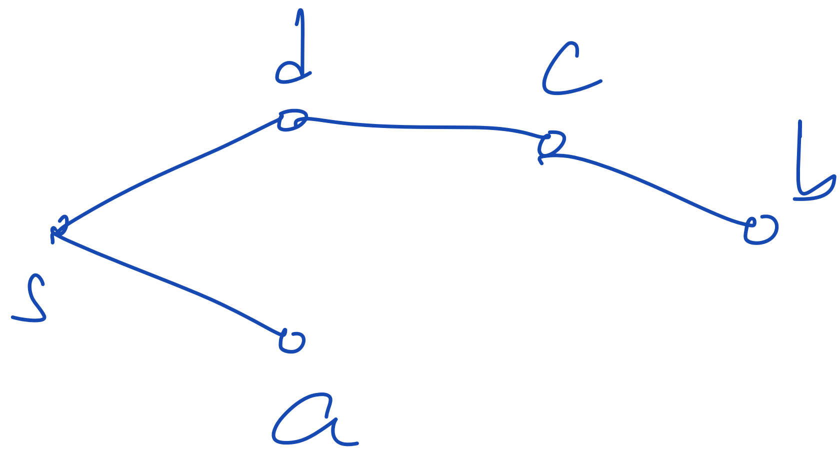
$\therefore \; \boxed{d[y]} < d[v]$

This is a contradiction because y has a smaller $d[\cdot]$ value than v & hence v cannot be the $(k+1)^{th}$ vertex brought into S.

(*) ~~Even when edge wts are negative Dijkstra works!!.~~ Bogus!

∞ -8

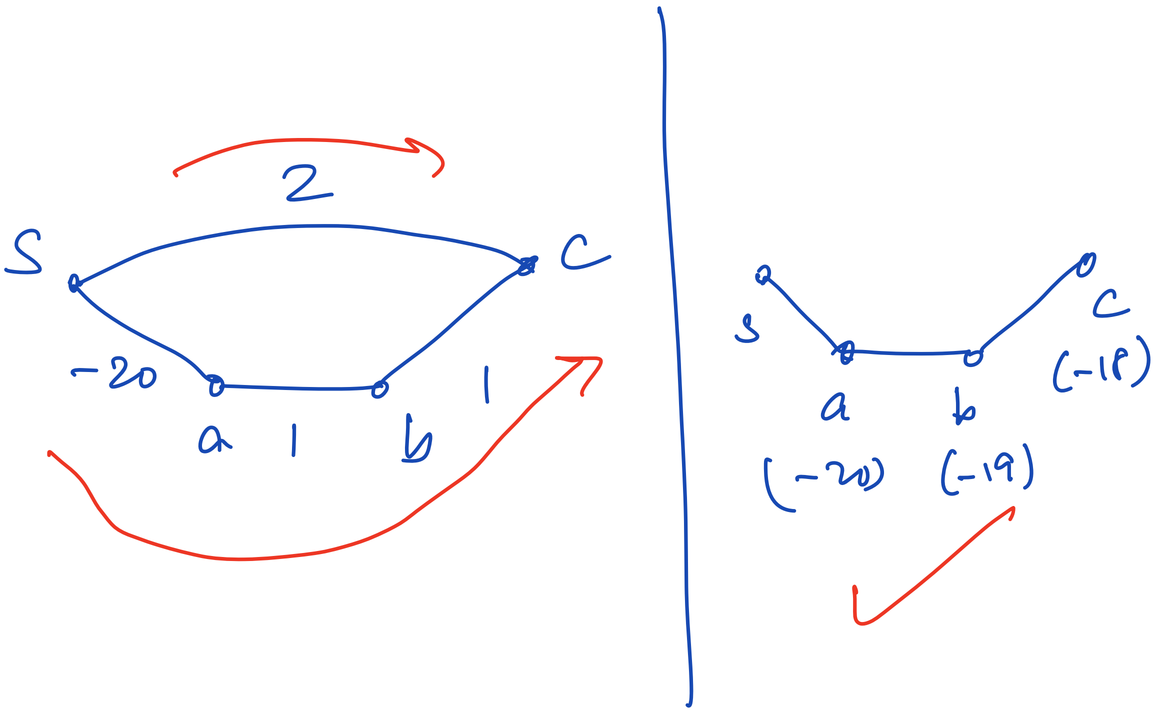


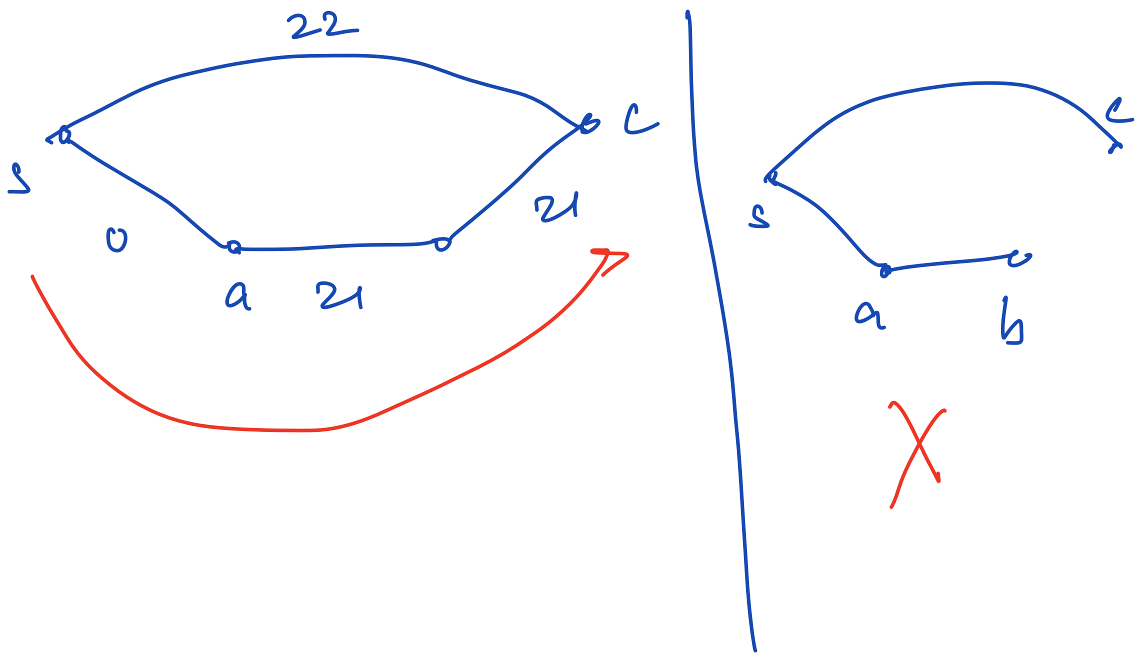


Small modification that will make Dijkstra work.

~~Add M to every edge wt, where $-M$ is the most negative edge wt in G.~~

Bogus!.



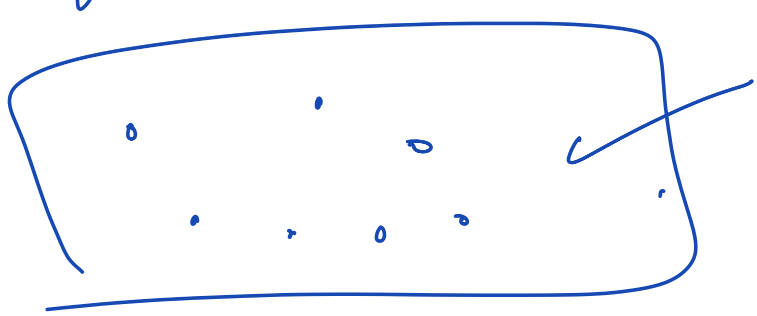


Minimum Spanning Trees

Input: Undirected, connected graph $G = (V, E)$

wts on edges (positive)

Objective: To obtain a minimum wt spanning subgraph of G, that is connected.



<u>Observation</u>: The output must always be a tree.

<u>Assumption</u>: Edge wts are distinct.

<u>Algorithms</u>

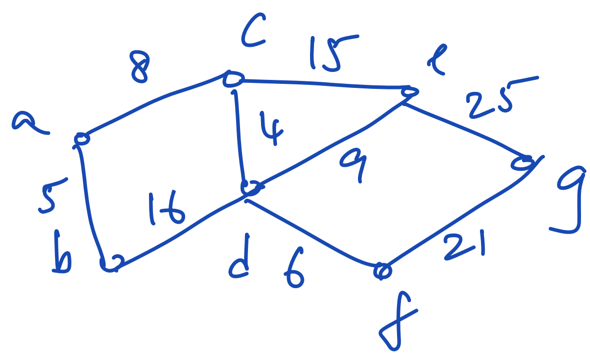
Prim's alg ("Wrong" Dijkstra's alg).

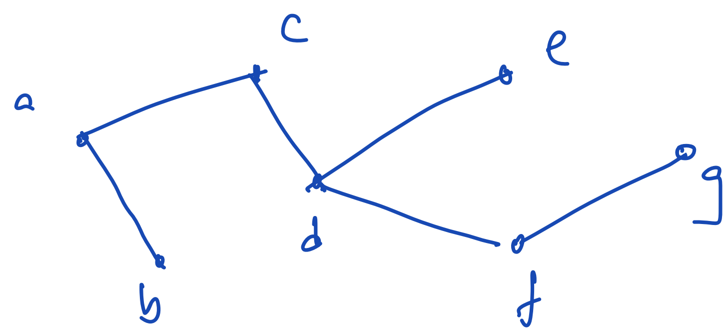
Kruskal's alg.

- Start with n vertices & no edges.
- Sort edges in ↗ order of their wts.
- Process the edges in that order:

  for each edge $e = (u,v)$:
      if adding (u,v) creates a cycle :
          do not add e to the soln
      else
          add e to the soln

Example:





Reverse Delete alg.

- Start with G
- Sort edges in ↘ order of wt.

- Process edges in the above order:

- if removing the edge does not disconnect the graph then

  remove it

  else

  keep it.