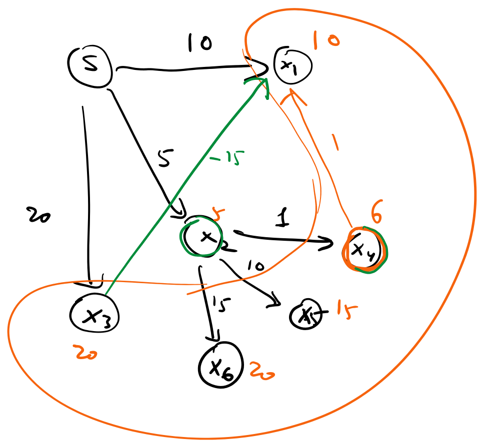
Dijkstra's shortest Path

$G = (V, E)$ is weighted graph

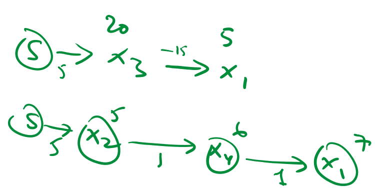
$w(u, v)$



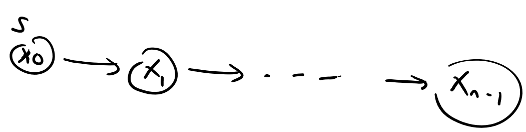
always pick the node with the shortest distance from the source for further expansion

Negative Edges cause violation of the above principle

$\Rightarrow$ Dijkstra's fails!



$|V|$ node



I) longest path from (s) to any other vertex has at most $\underline{\underline{|V|-1}}$ edges

II) update the $dist(x)$ at each step

One complete traversal over the edges

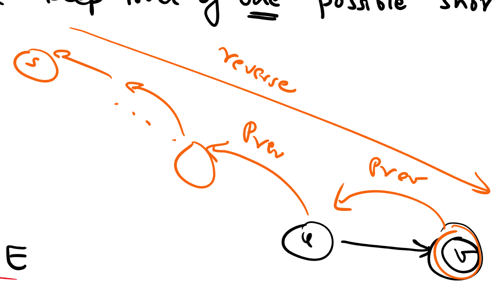
z

Bellman-Ford (G, s):

for all $u \in V$:

$O(|V|)$
dist(u) = ∞
prev(u) = [] ← list ← keep track of one possible shortest path

$\underbrace{\text{prev(u)} = [\,]}_{\text{traceback}}$



dist(s) = 0

for iter = 1 ... $\underline{|V|-1}$

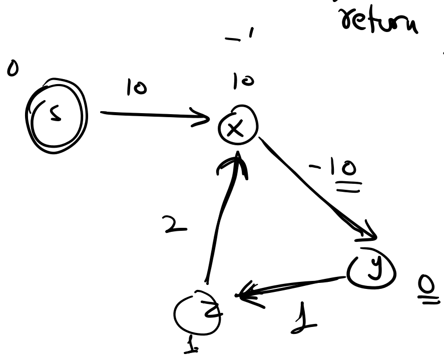
$O(|V| \cdot |E|)$ $|V|-1$ times; $O(|E|)$ for all $e = (u, v) \in E$

update (e) →

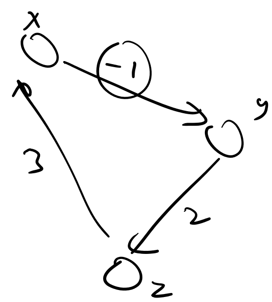
$O(|V|\cdot|E|)$ — $|V|-1$ times: — $O(|E|)$ for all $e=(u,v) \in E$
update(e) ⟶

if $dist(v) > dist(u) + w(u,v)$
then
$dist(v) = dist(u) + w(u,v)$
$prev(v) = u$

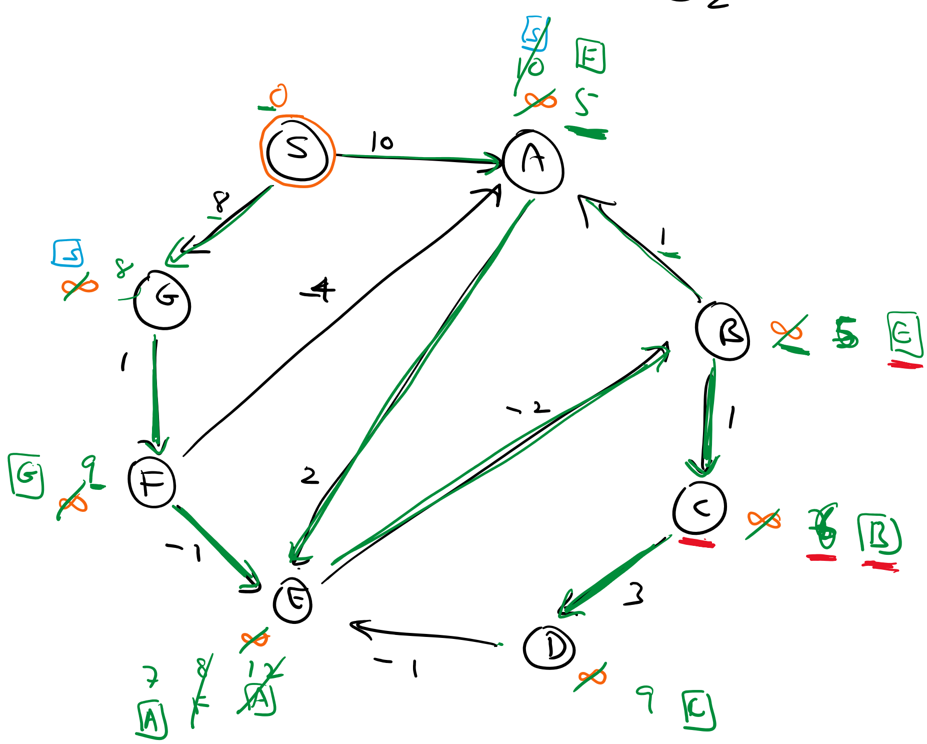
$O(|E|)$ do one more round over $e \in E$
if any dist decreases
return FAIL/NEG CYCLE



negative cycle : a cycle whose sum of edge weights is negative



$O(|E|) = O(|V|)$ sparse
⇒ $O(|V|^2)$ dense



C → dist = 6
↳ prev: C B E A F G S
reverse: S G F A E B C ⟶

DAGs : shortest paths
↳ topological sort
↳ then update the distances in that order

works for both +ve and -ve edge

Greedy Algorithms

Pick $\boxed{\text{the "best" option}}$ at each step
$\underbrace{\qquad}_{\text{local best}}$

Minimum Spanning Tree (MST) $\qquad G=(V,E)$

Given a weighted undirected graph, find a spanning tree with the least sum of edge weights

Spanning $\equiv$ must include all the vertices

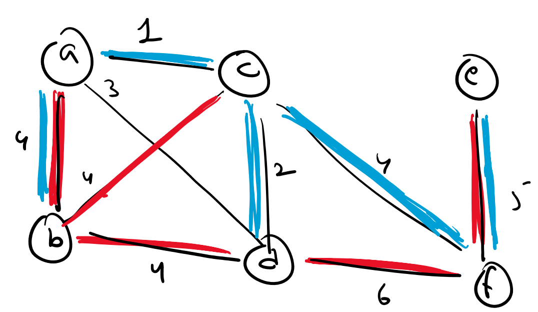
Spanning Tree: $\quad T=(V', E')$

$V' = V \quad$ (spans all vertices)

$E' \subseteq E \quad$ (E' is a subset)

Tree $\Rightarrow$ $|V|-1$ edge and connected

G



Spanning tree

$T_1: \{ ab, bc, bd, df, fe \}$

$Cost(T_1): \quad 23$

$4+4+4+6+5$

$T_2: \{ ab, ac, cd, cf, fe \}$

$Cost(T_2): \quad 4+1+2+4+5$

$= 16$

Minimum spanning tree

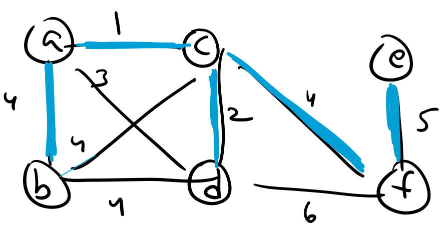
Initial idea: (greedy)

1) sort the edge in increasing order

(greedy)

Prim's

1) sort the edge in increasing order
2) pick the next best edge for T (exclude an edge if cycle)
   greedy
3) keep T connected (grow T by adding one more edge from the existing node in T)



global
1) ^ sort

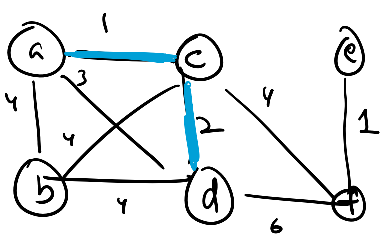
$O(|V| \log |V|)$

ac — 1 ←
cd — 2 ←
ad — 3 ← ? ✗
ab — 4 ←
bc — 4 ← ✗
bd — 4 ← ✗
cf — 4 ←
ef — 5 ←
df — 6 ← ✗

$T = \phi$

works!

when can this fail



⟹ Sort:

ac — 1 ←
ef — 1 ✗
cd — 2 ←
⋮

pick the next best option among the neighbors of existing node in T
greedy

Connected constraint

Prim's Algorithm(G):

for $u \in V$:
 cost(u) = ∞
 prev(u) = ϕ
Pick any source node $s \in V$
cost(s) = 0
PQ = [s] ← priority queue
while PQ not empty:

PQ = [s] ← priority Queue

while PQ not empty:

u = deletemin(PQ) ← picks the lowest cost vertex | greedy
min

for all $(u, v) \in E$

if $cost(v) > \underline{\underline{w(u,v)}}$ Consider current edge weight

$cost(v) = w(u,v)$

$prev(v) = u$

decreasekey(PQ, v, cost(v))

$O(|E|)$ decrease key

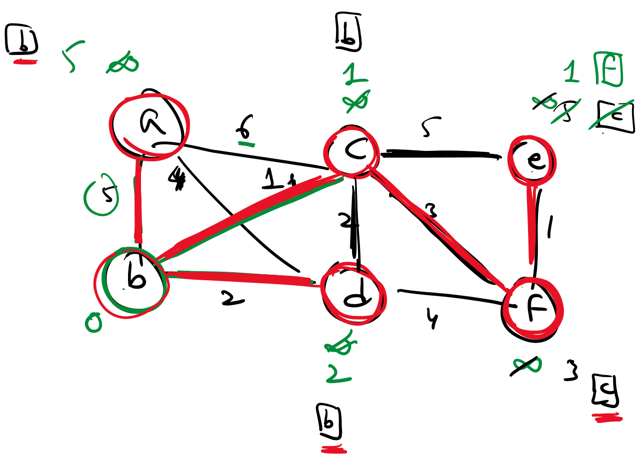
$O(|V|)$ deletemin

$O((\underline{\underline{|E| + |V|}})\ \underline{\underline{\log(|V|)}})$ via binary heaps

time per operation

⇓

$O(|E| \log |V|)$



T = { red edge }

⇓

look at prev for each node

e → f	1
f → c	3
c → b	1
d → b	2
a → b	5
	12

Correctness?

Cut property

cut in a graph is simply a partition of the vertex set V into two parts

$\underline{\underline{S}}, \underline{\underline{V-S}}$

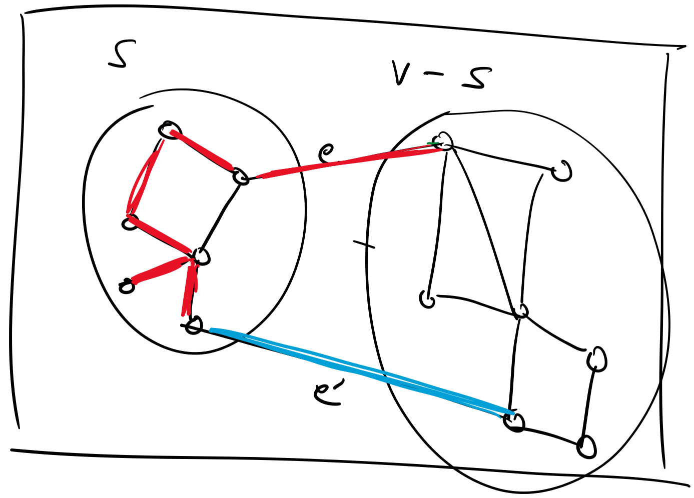
$S \subseteq V$

$S, \bar{S}$

S ↑ ↑ Complement of S

$\bar{S} = V - S$

complement of S



given the current $T = (V', E')$

we need to grow by one more edge

let $S = V'$

next node to be chosen must come from $V - S$

1) If there is only one edge between S & $V-S$, then we must include it

2) there is more than 1 edge between S and $V-S$

e is min cost/weight edge

$$w(e) \le w(e')$$

Show that choosing the least cost edge is guaranteed to give us an MST

Assume $\exists$ some MST T' that does not include e, then it must include e' (e' is some other edge that crosses the cut)

Cost of ~~that~~ MST

$$Cost(T') = \underbrace{w(S) + w(V-S)} + w(e') \ge \underbrace{w(S) + w(V-S)} + w(e)$$

we can construct a new MST T'', that includes e instead of e'

true for ANY cut in G that respects T

no current edge in T crosses from S to $V-S$

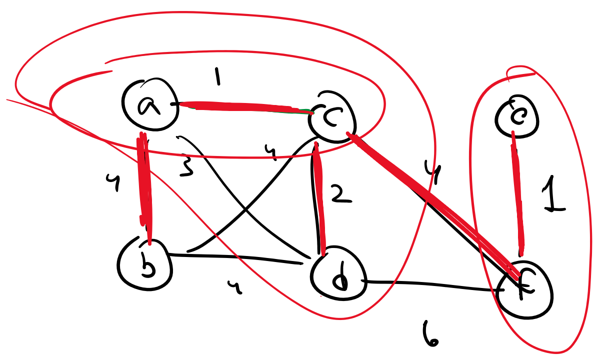
$\Rightarrow$ greedy choice (...

S to V − S

⇒ greedy choice (edge with the least weight) must be part of some MST

guarantees correctness.

Kruskal's method : drops the connectedness constraint



$S = \{a, c, e, f\}$

$V - S = \{b, d\}$

$S = \{a\ c\ d\ e\ f\}$

$V - S = b$

? { Check for cycles
check for disconnectedness

Maintain a forest