f(5)

f(4)

f(3)

f(3)

f(2)

f(2)?

f(3)

f(1)

CS Dojo

f(5)

100000

Memoization.

HashMap

Bottom.

f(5)

$f(.)$ — $f(s)$

$f(1)$

Recursion

$O(n)$

n-ary

$O(n)$

$f(0)$

$\times$ $\times$

$O(1)^{n-1} + O(1)^{n-2}$

$\times n$

100

log W

size

2S

2

K

2^{size}

10^W

nW

ton

exp

$W_{max} = 11$

$W = \{1, 2, 5, 6, 7\}$

$P = \{1, 6, 18, 22, 28\}$

$w \longrightarrow$

P_i	W_i	$i \backslash w$	0	1	2	3	4	5	6	7	8	9	10	11
		0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	0	1	1	1	1	1	1	1	1	1	1	1
6	2	2	0	1	6	7	7	7	7	7	7	7	7	7
18	5	3	0	1	6	7	7	18	19	24	25	25	25	25
22	6	4	0	1	6	7	7	18	22	24	28	29	29	40
28	7	5	0	1	6	7	7	18	22	28	29	34	35	$\boxed{40}$

$$m[i,w] = \max\left(m[i-1, w],\ m[i-1, \overset{6-5}{w - W_i}] + P_i\right)$$

$m[2, 1]$

$1 \quad , \quad 0 + 6$

$1 + 18$

$\cancel{40 - 22}$

18

$$W = \{ \overset{1}{1},\ \overset{2}{2},\ \overset{3}{5},\ \overset{4}{6},\ \overset{5}{7} \}$$

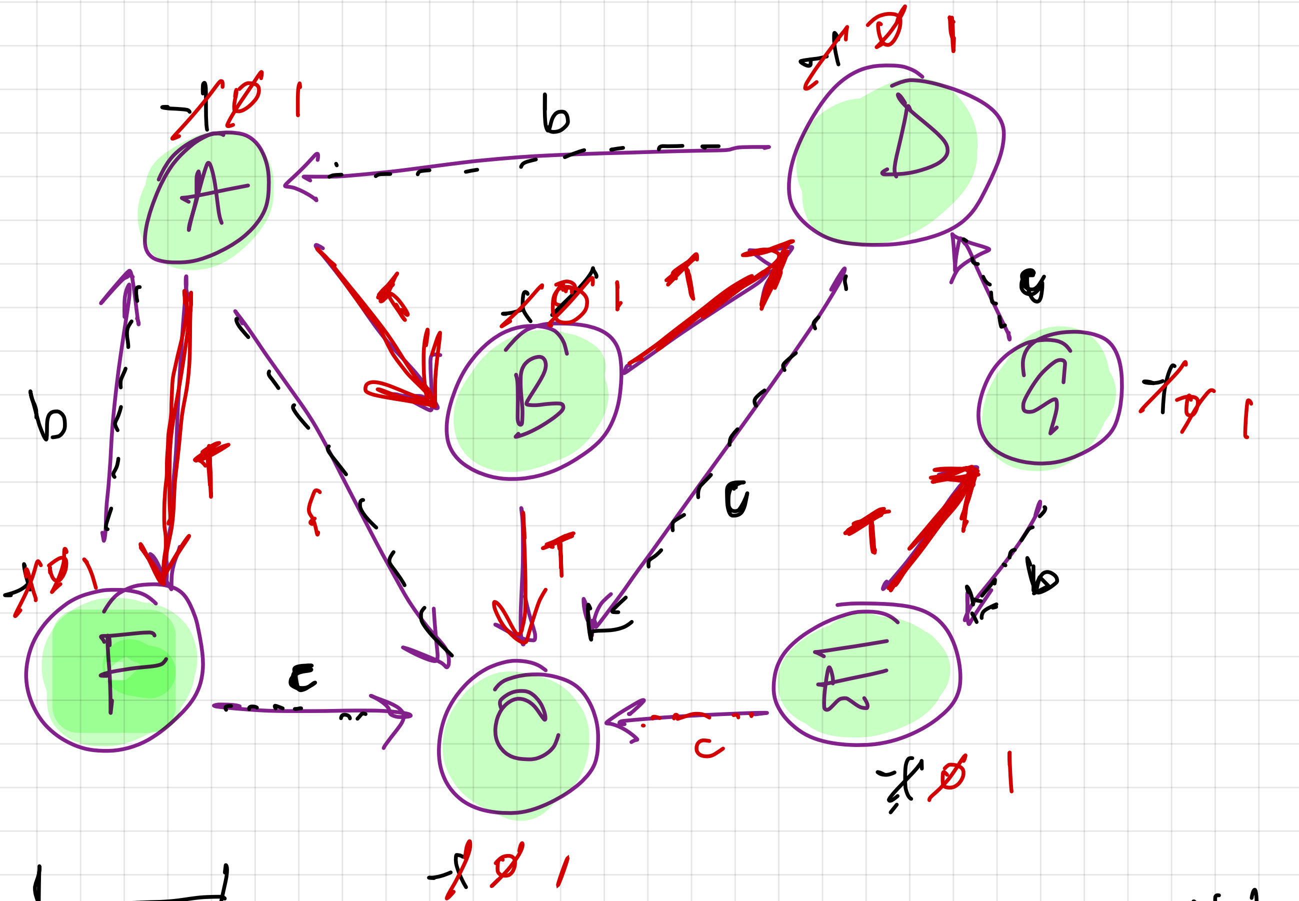
$18 - 18 = 0$

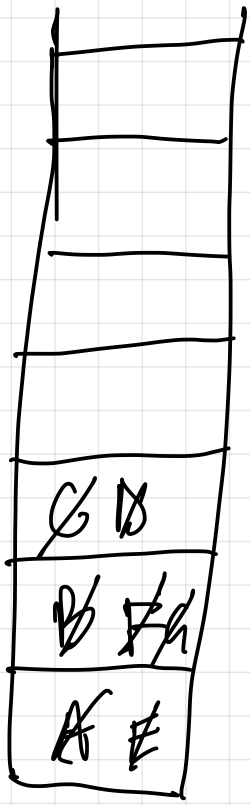
$$\text{Selected} \{ 0,\ 0,\ 1,\ 1,\ 0 \}$$

Su Tree

Cur

S.



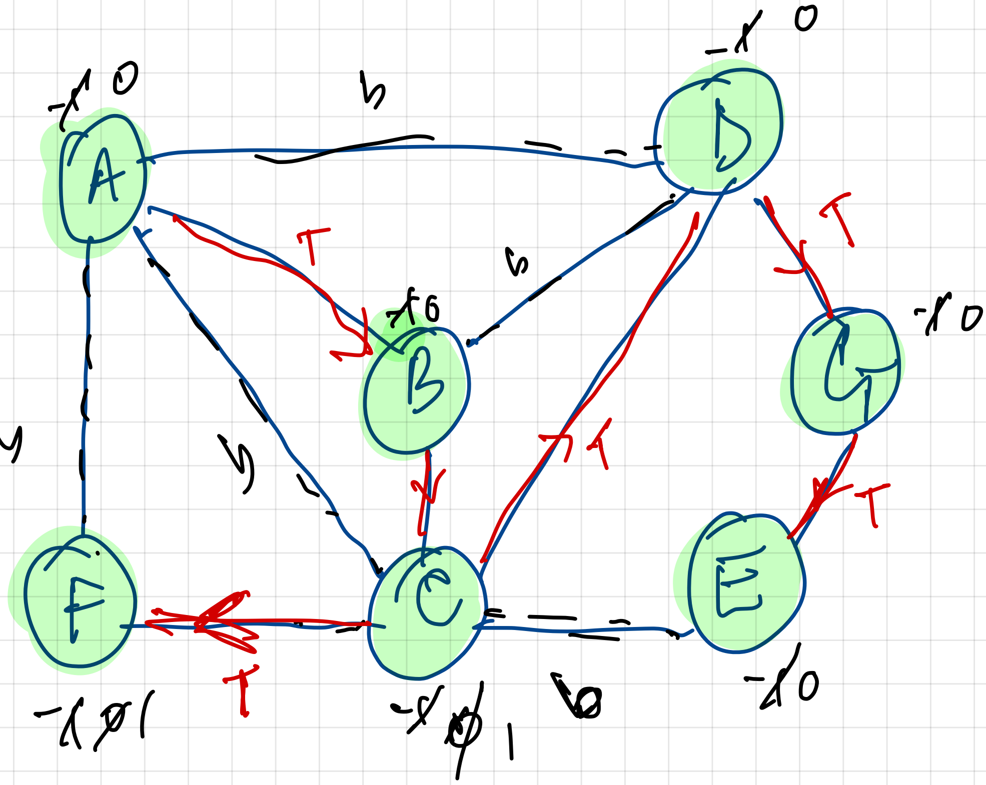


-1 → Not

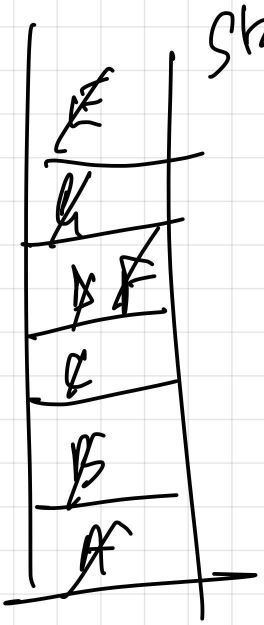
0 → visi

1 → Explore

C, D, B, F, A, G, E

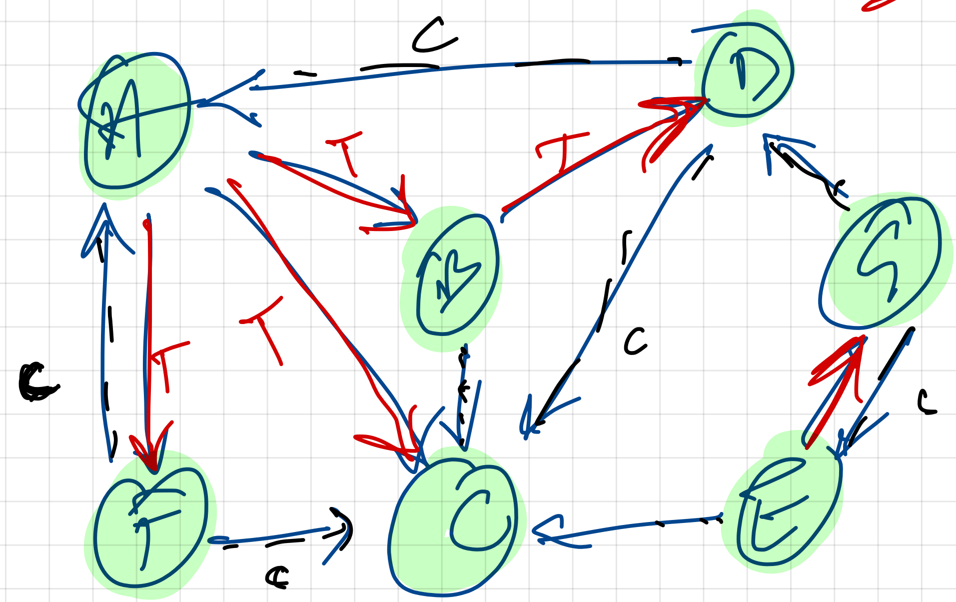


Stack



E, G, D, F, C, B, A

B.F.S



Queue : ~~A~~, ~~B~~, ~~C~~, ~~F~~, ~~D~~, ~~E~~, ~~G~~,

Time: O(|E|lg|E|) given fast FIND-SET, UNION

Invariant: Minimum weight spanning forest

```
MST-KRUSKAL(G, w)
A ← ∅
for each vertex v ∈ V[G]
      do MAKE-SET(v)
sort the edges of E by nondecreasing weight w
for each edge (u, v) ∈ E, in order by nondecreasing weight
      do if FIND-SET(u) ≠ FIND-SET(v)
            then A ← A ∪ {(u, v)}
                  UNION(u, v)
return A
```

$O(V)$

Quick Sort.

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

$|E|$

Becomes single tree at end

ASK Professor).

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

Sort.

Extract min → $(\log V)$

loop is E

Time:
$O(|E|\lg|V|)$
=
$O(|E|\lg|E|)$
slightly faster with fast priority queue

```
MST-PRIM(G, w, r)
    Q ← V[G]
    for each u ∈ Q
        do key[u] ← ∞
    key[r] ← 0
    π[r] ← NIL
    while Q ≠ ∅                                  → Runs V times
        do u ← EXTRACT-MIN(Q)                    → log|V|
            for each v ∈ Adj[u]                  → Runs in δE *
                do if v ∈ Q and w(u, v) < key[v]
                    then π[v] ← u
                         key[v] ← w(u, v)        log|V|
```

Invariant: Minimum weight tree

Spans all vertices at end

min Heap which is a Max-Heap

implementation is assumed to be a heap.

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

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

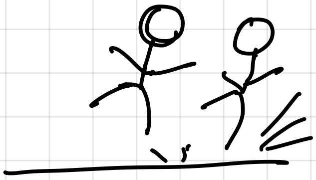
but $V << E \Rightarrow O(E \log V)$

* $V \times \delta E$ (has been shown to be) is close to E

for adjacent nodes

$\rightarrow \sum \delta E \Rightarrow V \cdot \delta E \approx 2E = O(E)$

Badguys



Super Woman.

for (nonnegative) weighted, directed graph G=(V,E)

```
DIJKSTRA(G, w, s)
  INITIALIZE-SINGLE-SOURCE(G, s)
  S ← ∅
  Q ← V[G]
  while Q ≠ ∅
      do u ← EXTRACT-MIN(Q)
         S ← S ∪ {u}
         for each vertex v ∈ Adj[u]
             do RELAX(u, v, w)
```

→ Runs $|V|$ times.

→ $\log|V|$

→ δE δE ⊛

→ $\log|V|$

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

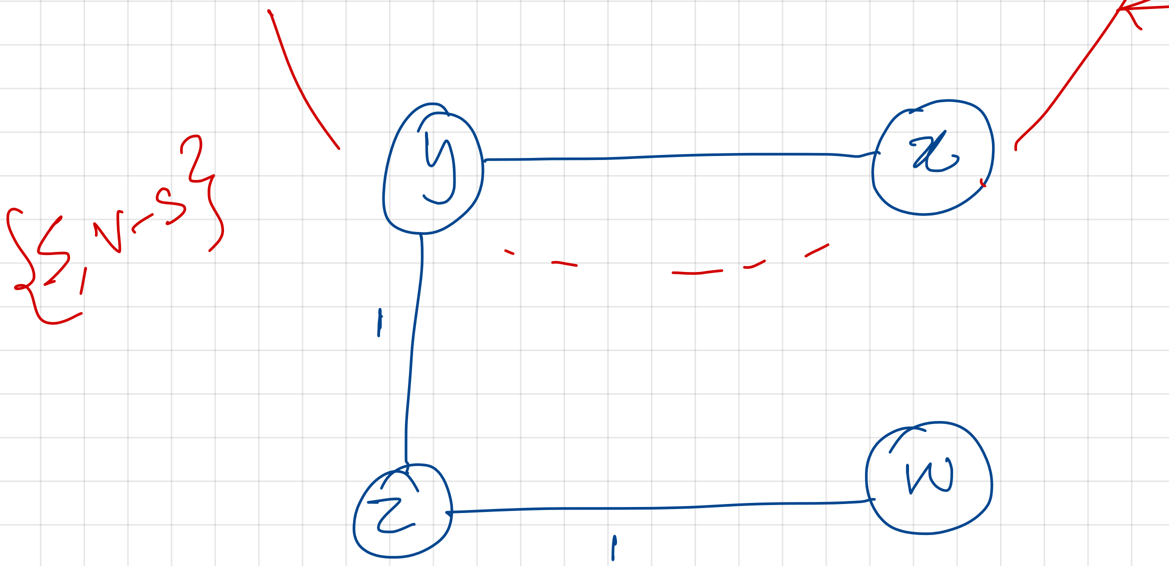
$O((V+E) \log V)$

but $V << E \Rightarrow O(E \log V)$

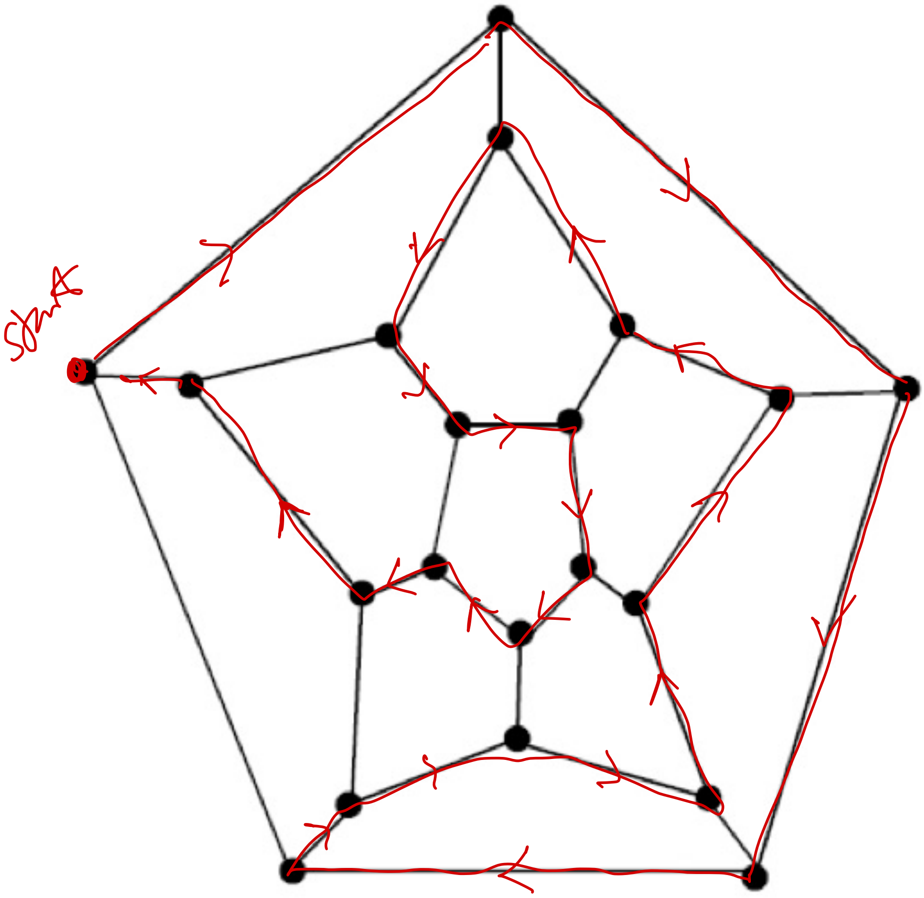
⊛ $V \times \delta E$ is (has been shown to be) close to E

for adjacent nodes

$\sum \delta E \Rightarrow V \cdot \delta E$
$\approx 2E$
$= O(E)$



If you take this edge out, then that is no longer an MST



We cannot be friends.

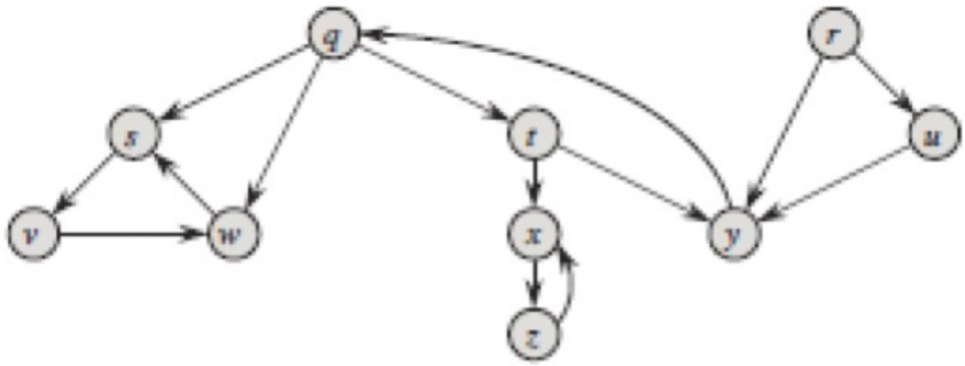


Figure 22.6 A directed graph for use in Exercises 22.3-2 and 22.5-2.