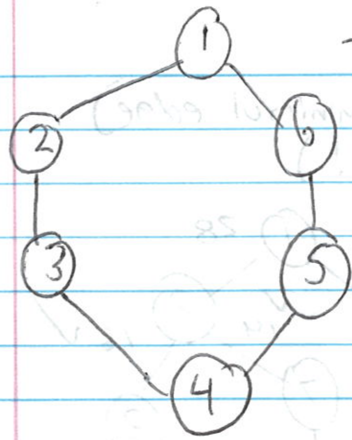
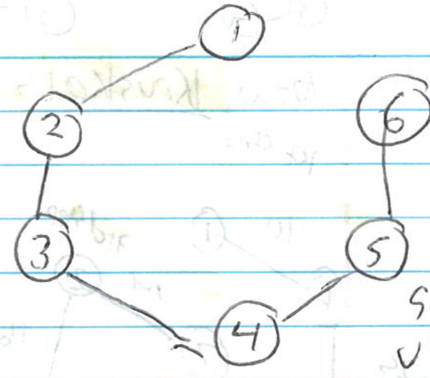
Minimum Cost Spanning Tree

no weights



→



subgraph

↓

$S \subseteq G$

$S = (V', E')$

$V' = V$

$|E'| = |V| - 1$

$G = (V, E)$

$V = \{1, 2, 3, 4, 5, 6\}$

$E = \{(1,2), (2,3)(3,4) \dots\}$

$|V| = n = 6$

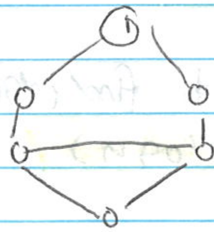
$n - 1 = 5$

so 5 edges

How many spanning trees can be created?

$|E| = 6$ $\qquad$ $6C5 = 6$ so 6

now



$7 C 5 = 6$

↑
edges

← since 2 cycles

$7C5 - 2$

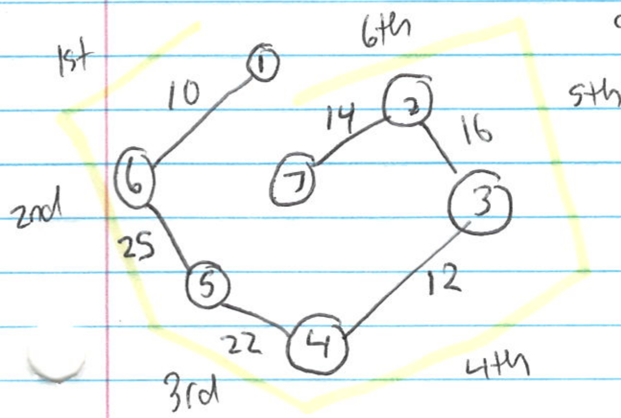
Formula: $_{|E|}C_{|V|-1}$ − number of cycles

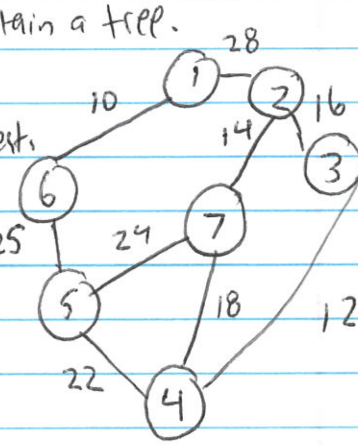
---

weights

Prim's ★ Always maintain a tree.

★ Note: don't need to choose smallest edge, can choose any vertex and pick smallest.

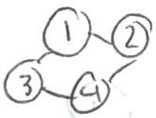




cost = 99

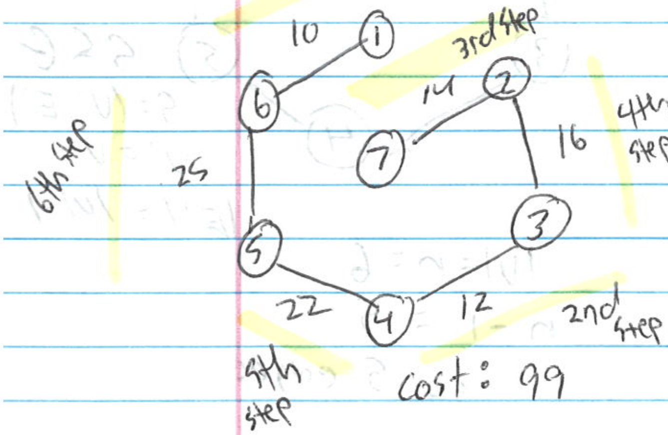
- Choose vertex, pick smallest edge
- Pick next smallest edge that continues the tree.

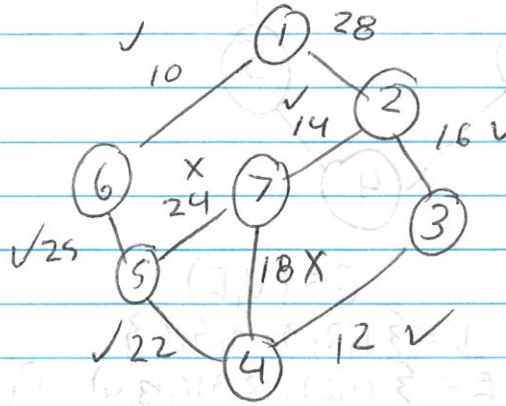
two pieces Prim Fails! No alg. can! Spanning tree must be connected



fails!

Now, **Kruskals** (Always pick minimal edge) (Don't form cycle!)





cost: 99

$|E|$ $\quad |V|-1$

time: $O(|V||E|)$

$O(n \cdot e) = O(n^2)$

if you use min heap: $\log n$ to find edge now

time with min heap: $O(n \log n)$