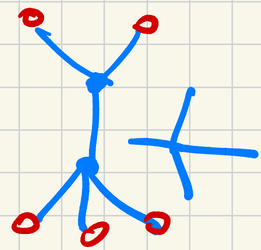
HONORS ALGORITHM | 2025-02-12

$G = (V, E)$

tree : (undirected) graph

$$\begin{cases} \text{no cycles} \\ \text{connected} \end{cases}$$

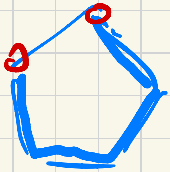


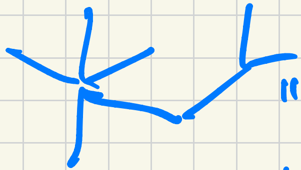
no cycles : __forest__

__EX__ tree $\Rightarrow$ $\underset{\#\text{edges}}{m} = \underset{\#\text{vertices}}{n} - 1$

__Lemma__ tree w $n \geq 2 \Rightarrow \exists$ vertex of $\deg = 1$

__Pf 1__ : endpoints of a longest path have $\deg = 1$





"free tree"

$\longrightarrow$ tree

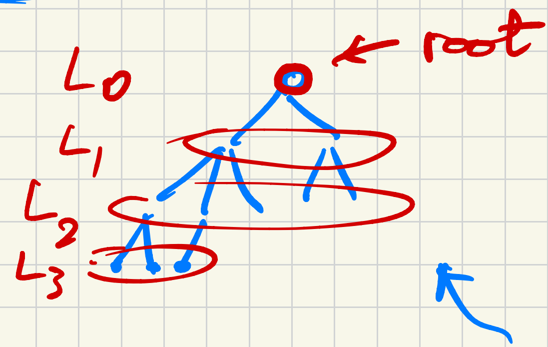
$G = (V, E)$

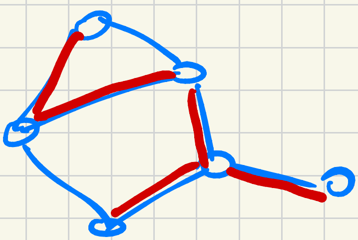
$H = (W, F)$

$H \subseteq G$ subgraph

if $W \subseteq V$

$F \subseteq E$



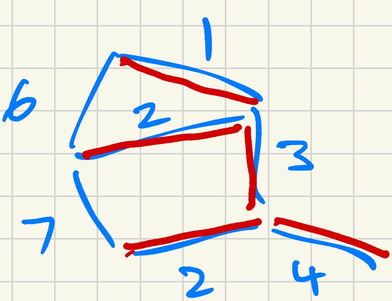


spanning tree:

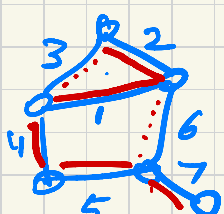
spanning subgraph

$W = V$

that is a tree



GREEDY ALGORITHM



G has a sp tree $\iff$ G is connected

$\therefore$ if G is conn then $m \geq n-1$

KRUSKAL's algorithm

$E = \{e_1 \ldots e_m\}$ $\quad w: E \to \mathbb{R}$

$w(e_1) \leq \ldots \leq w(e_m)$

init. $F = \emptyset$

for $i = 1$ **to** m

  if $F \cup e_i$ has no cycle

   then add e_i to F

return F

DO! this creates min-weight spanning tree

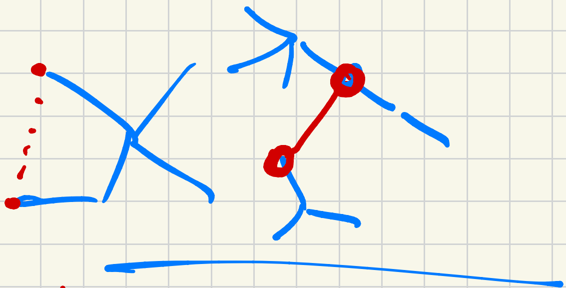
DO If all weights distinct $\Rightarrow$ min-weight sp tree is unique

requires:

(1) sorting edges by comparisons $m \log_2 m$ $= O(m \log_2 n)$

$m \leq \binom{n}{2} < n^2 \quad \therefore \log m < 2 \log n$

loop invariant: (V, F) is a forest

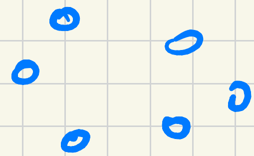


initially n components

in each round

#comp --

after n-1 rounds: #comp's 1



(2) decide: given $u, v \in V$

is v accessible from u in $\underbrace{(V, F)}_{\text{current forest}}$

3A) if BFS: $\Theta(n)$

total cost $\Theta(n^2)$

need UNION-FIND data structure

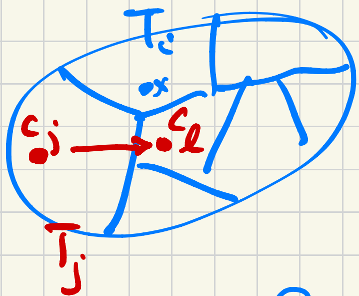
maintains a partition of the set V

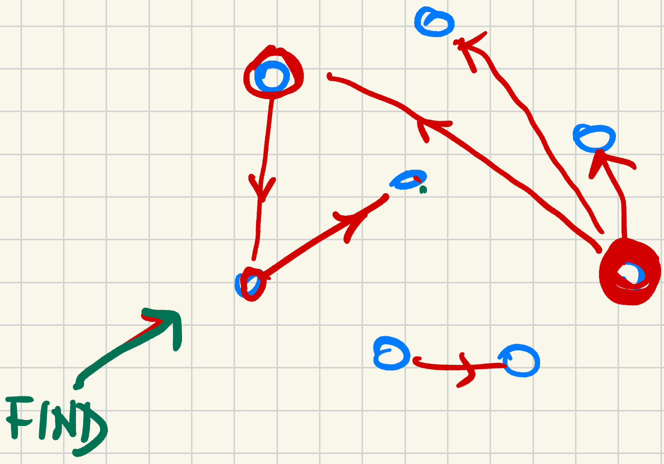
$$V = \{T_1 \ldots T_k\} \quad k \text{ variable}$$

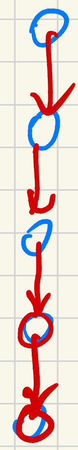
for $x \in V$

FIND(x) returns i if $x \in T_i$

UNION(x,y) :
merges (FIND(x), FIND(y))





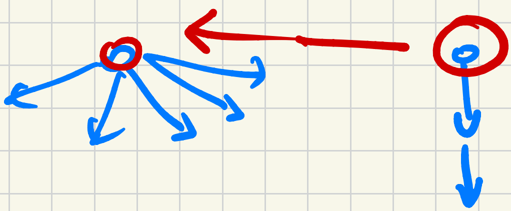


convention 1 : bigger wins

HW : prove : depth $\le \log_2 n$

convention 2 : deeper wins

HW : again : depth $\le \log_2 n$



Under either rule

cost of FIND : $O(\log n)$

cost of UNION : $O(1)$

total cost if m Finds, $n-1$ unions $\Big\}\ O(m \log n + n)$

cost of Kruskal