So far Use induction to design algorithms (induction on n)

↙ Color a graph

↘ construct a tree

Induction on steps of ALG.

$$\underset{\text{hold}}{P(t)} \longrightarrow \underset{\text{holds}}{P(t+1)}$$

P is a property of ALG.

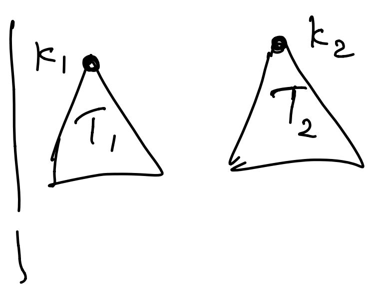
Claim: At any time in ALG for every tree if root has label $k \Rightarrow$ there are at least 2^k nodes.

Pf: Induction on step of ALG.

Suppose at time t, for any tree there are $\geq 2^{\text{root-label}}$ nodes in tree.

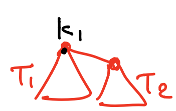
Say at time $t+1$ we merge two trees T_1, T_2

By IH: $|T_1| \geq 2^{k_1}$, $|T_2| \geq 2^{k_2}$.



Case 1: $k_1 > k_2$.

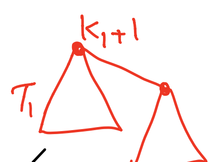
The new tree has $2^{k_1} + 2^{k_2} \geq 2^{k_1}$ ✓



Case 2: $k_1 = k_2$

new tree has $\geq 2^{k_1} + 2^{k_2} = 2 \cdot 2^{k_1}$

$= 2^{k_1+1}$ ✓



In both cases the new tree has $\geq 2^{\text{label-root}}$ many nodes

---

T is output of kruskal on $c_{e_1} - c_{e_m}$, T^* is OPT. (c)

$$c_{e_1} \leq c_{e_2} \leq \quad \cdots \leq c_{e_m}$$

$$1 \leq 1 \leq 2 \leq 2 \leq 2 \leq 3 \leq 3 \leq 3 \leq 3 \leq 4$$

$$1 \quad 1.0001 \quad 2.0002 \quad 2.0003 \quad \cdots$$

$$c'_{e_i} = c_{e_i} + i \cdot \varepsilon$$

$\varepsilon = 0.0001$

Run kruskal on $(c'_{e_1}, -, c'_{e_n}) \Rightarrow$ give same tree T.

B/C kruskal only uses the ordr not the cost

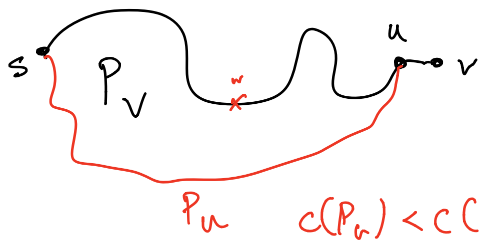
For contradictio

$c(T^*) < c(T)$ then if $\varepsilon < \dfrac{c(T) - c(T^*)}{m^2}$

$$c'(T^*) \leq c(T^*) + m^2\varepsilon < c(T) \leq c'(T)$$

$\Rightarrow c'(T^*) < c'(T) \Rightarrow T$ is not OPT for c'

Dikstra's ALG.



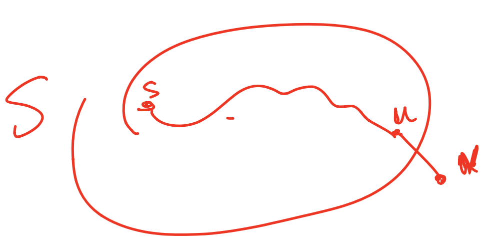
shortest path to v P_v shortest path

u is before v in P_v

$P_v - (v,u)$ is shortest path to u.

$c(P_u) < c(P_v - (v,u))$

$\Rightarrow c(P_u + (u,v)) < c(P_v)$ contradiction



is closest to s