So far, use induction to design algorithms (induction on $n$)

- Color a graph
- Construct a tree

**Induction on steps of ALG.**

$$P(t) \rightarrow P(t+1) \quad P \text{ is a property of ALG.}$$

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

**Pf:** Induction on step of ALG.

Suppose at time $t$, for any tree there are $\geq 2^k$ nodes.

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:** $k_1 = k_2$.
  - The new tree has $2^{k_1} + 2^{k_2} = 2^{k_1} + 2^{k_1} = 2^{k_1+1}$.

In both cases the new tree has $\geq 2^{k+1}$ nodes.

- $T$ is output of Kruskal on $C_{e_1}$, $C_{e_2}, \ldots C_{e_m}$, $T^*$ is OPT.

$$C_{e_1} \leq C_{e_2} \leq \ldots \leq C_{e_m}$$

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

$$1, 000, 1, 000, 2, 000, 3, 000, 4, 000 \ldots$$
Run Kruskal on \((C'_{e_i}, \rightarrow, C'_{e_n})\) ⇒ give saw tree \(T\).

For contradiction

\[ C'(T^*_+) < C(T) \text{ then if } \varepsilon < \frac{C(T) - C(T^*_+)}{m^2} \]

\[ C'(T^*_+) < C(T) + \varepsilon m^2 < C(T) < C' (T) \]

⇒ \( C'(T^*_+) < C'(T) \Rightarrow T \text{ is not OPT for } C' \)

Dijkstra's ALG.

\[ s \quad \text{shortest path to } v \quad \text{Pv shortest path} \]
\[ u \text{ is before } v \text{ in } P_v \]
\[ P_v \quad (v, u) \text{ is shortest path to } u. \]

\[ P_w \quad C(P_w) < C \left( P_v - (v, u) \right) \]

⇒ \( C(P_w - (w, u)) < C(P_v) \) contradiction

\[ s \quad n \text{ is closest to } 3 \]