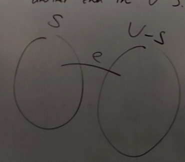
Cut property: assuming all edges have distinct costs. Let S be any subset of vertices that is neither empty nor the entire set, and let $\underline{e = \{v, w\}}$ being the (greedy) minimum-cost edge with one end in S and another end in $V-S$. Then every minimum spanning tree contains the edge e.



" e, picked by greedy, is <u>correct</u>."

" because kruskal's algorithm terminates when there exists only one connected component, all vertices are covered $\Rightarrow$ the edges that have been selected are <u>sufficient</u>"

(relax

Propositio

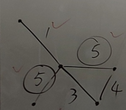
Proof.

(relax the distinct-cost constrain)

Proposition: Kruskal's algorithm is correct on graphs with equally-weighted costs.

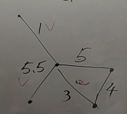
Proof: We introduce small weight perturbations to all equal-weighted edges in G, and transform G into G' where all edges are distinct-weighted. We can run the Kruskal's algorithm on G' and obtain a solution.

G



$W(T) = 1+5+3+4 = 13$

G'



$W(T') = 1+5.5+3+4 = 13.5$

The maximum weight purterbation we can introduce is $\varepsilon \sum_{i=1}^{|E|} i$ (if the unit purterbation is ε)

Let T' be the MST identified in G', and T be the MST identified in G. (through mapping T' in G)

$W(T')$: weight associated with the MST T'.

$W(T') < W(T) + \varepsilon \sum_{i=1}^{|E|} i$, or otherwise we can obtain a lower-cost spanning tree by removing all weight perturbations applied in G'.

$$W(T') < W(T) + \varepsilon \sum_{i=1}^{|E|} i$$

because ε is arbitrarily small, we can make

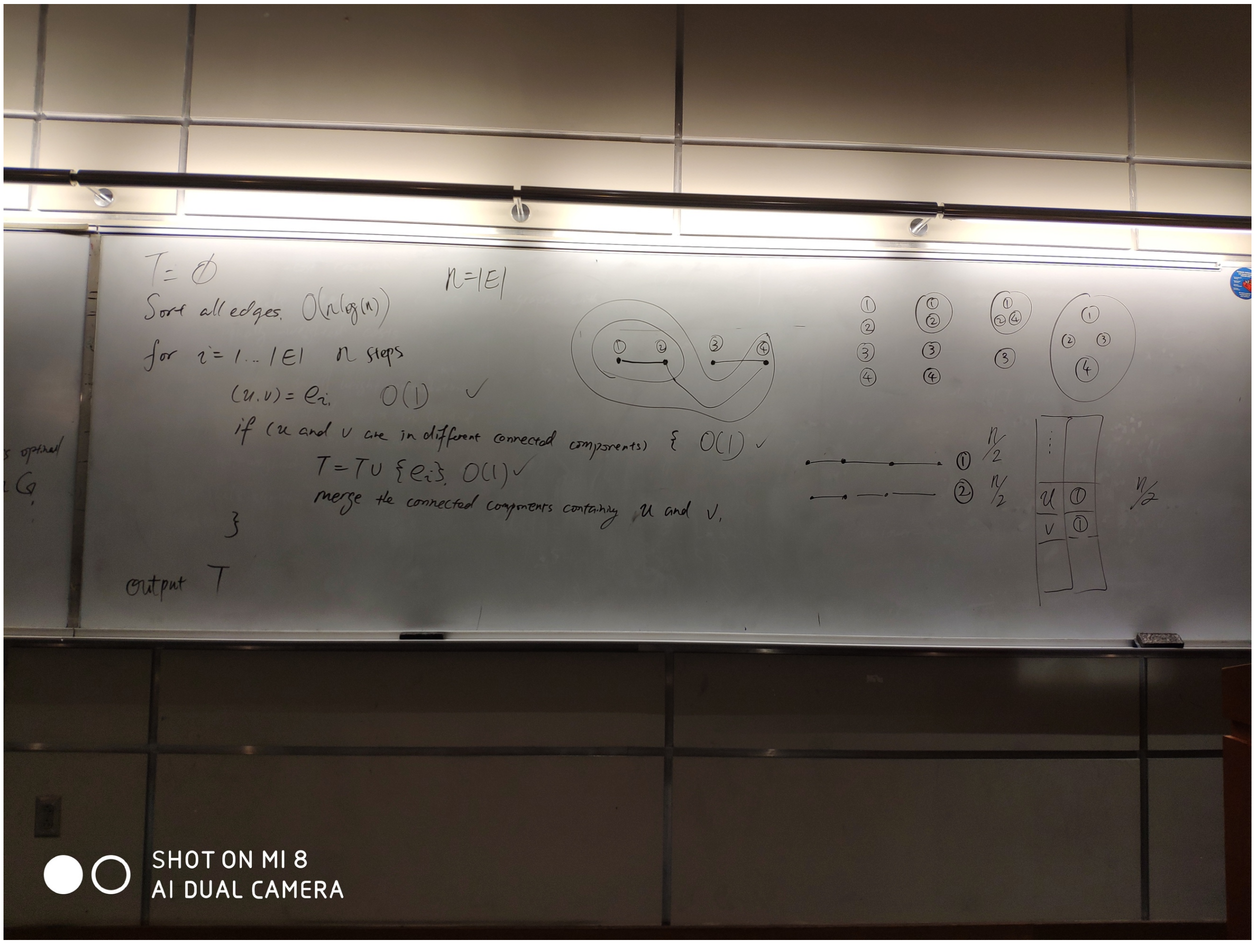
$\varepsilon \sum_{i=1}^{|E|} i < d$, where d is minimum weight ($d > 0$)

difference between any two distinct-weighted edges in G.

if there is any.

$$W(T') - W(T) < \underbrace{\varepsilon \sum_{i=1}^{|E|} i}_{} < d \Rightarrow \text{the weight difference of the trees is smaller than the minimum weight difference} \Rightarrow \underline{W(T') = W(T)}$$

T is optimal in G


T = ∅
n = |E|
Sort all edges. O(n log(n))
for i = 1 ... |E| n steps
(u, v) = e_i O(1)
if (u and v are in different connected components) { O(1)
T = T ∪ {e_i}, O(1)
merge the connected components containing u and v.
}
output T
SHOT ON MI 8
AI DUAL CAMERA