Kruskal's Algorithm.

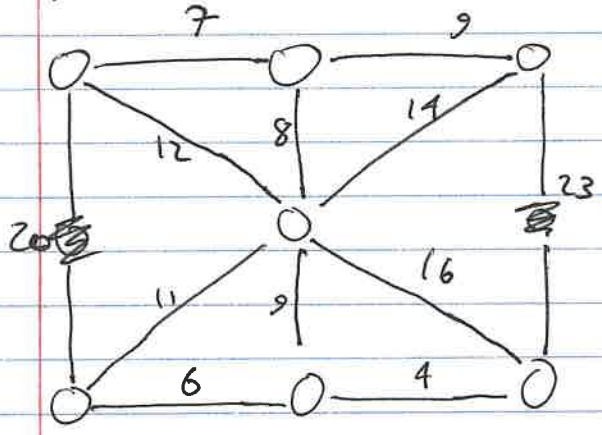
Greedy

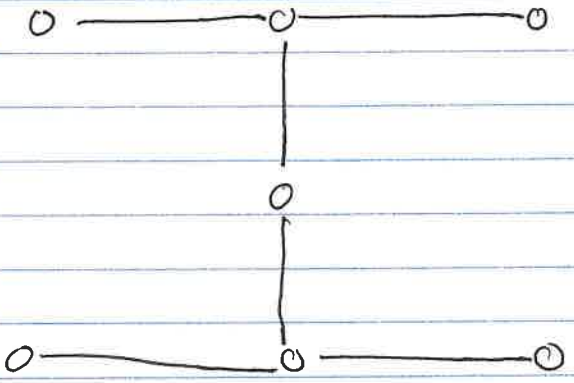
Greedy algorithms are fairly obvious. When they work, they work well, but they don't always work. Suppose you are trying to pack luggage and you have only 100 units of space. If you have 3 pieces of luggage of weight 70, 40, 50, a greedy algorithm might choose the maximum first. This with only 30 units of space left, couldn't fit any more luggage. A better algorithm would not choose the largest first and would rather pick $40+50=90>70$. to carry more luggage.

Given a weighted undirected graph of G, a minimum spanning tree is a set of $n-1$ edges T such that all of V shares an edge with T (it's spanning) and $\sum_{e \in T} w_e$ is the smallest (it's minimum).

Edge E + $O(VE)$ = $O(V^2)$

This problem actually has an efficient greedy solution! Sort E by weight, repeatedly choose the smallest edge which does create a cycle. Note that a tree T is a graph with exactly n edges and no cycles.





Unlike other algorithms, the correctness is not obvious? How do we know it is always forms a spanning tree? It certainly will be a cycle, but why spanning? and why minimal? We will spend today more on the proof of correctness.

To prove that Kruskal's algorithm does output an MST, we show that the in progress set of edges are on the way to building, and therefore outputting an MST.

cut property

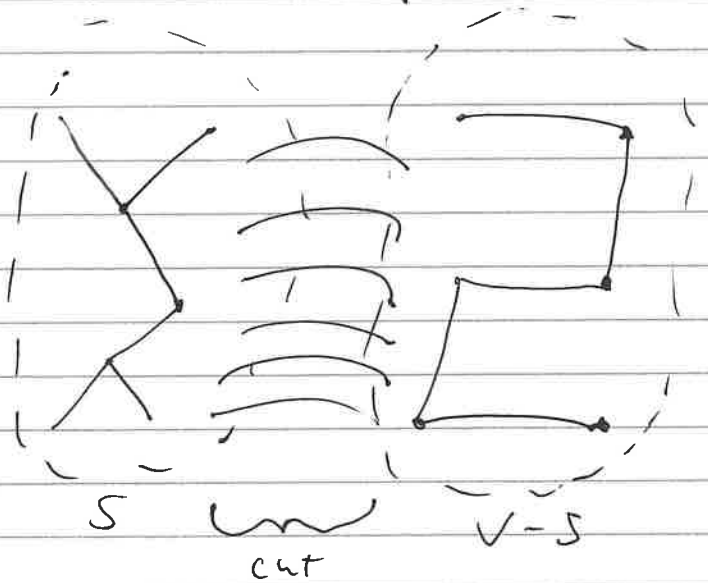
We prove the following cut property: Let X be a set of edges in some in progress MST so $X \subset T$ with T and MST of G. Let e be the next edge chosen by our algorithm. We want to prove $X + e \subset T'$, that $X+e$ is part of some MST $\& T$, so we continue to be on our way to building an MST.

cut

Let X be an in progress set of edges to be early an MST. Let $S, V-S$ partition G so no edge of X crosses S to $V-S$ or back. Let e be the lightest edge which does cross S to $V-S$. Then $X+e \subseteq T'$, $X+e$ is part of an MST. A cut is a set of edges from S to $V-S$.

facts

Note for any connected graph with $S, V-S$ non empty, every cut contains atleast one edge. Otherwise it would be disconnected. Second for each cut, one edge must be in the MST. We want to show the smallest edge in the cut is part of the MST.



Let $X \subset T$ with e the lightest edge in some $S, V-S$ cut. If $e \in T$, then $X+e \subset T$ so $X+e$ is part of an MST and there is nothing to do. So suppose $e \notin T$. Consider $T+e$. Since $S, V-S$ a cut, and T is an MST, there is some other edge already across the cut, say e'. So if $T+e$ spans S and $V-S$ with two edges in the cut, then $T+e$ is not a tree, and contains a cycle through e and e'. we disconnect the cycle to form $T' = T+e-e'$. We want to show that T' is also an MST. It certainly is spanning since T was spanning, so we prove it is minimal.

$$w(T') = w(T) + w(e) - w(e').$$

Since e was the lightest edge in the cut, $w(e) \leq w(e')$. Then $w(T') + w(e') = w(T) + w(e)$. Combining these, we see $w(T') \leq w(T)$. But since T is an MST $w(T)$ is minimal, so it could only be that $w(T') = w(T)$ so T' is an MST and $X+e$ is in progress to an MST T'.

Lets apply the proof of the cut property towards the justification of Kruskal's. At each step we have a partial solution X. e is the lightest of all the edges so it is the lightest in some cut. Since we check first that adding e doesn't add a cycle, no other edge of this cut has been selected before, so choosing e to get $X+e$ is on the ways to eventually outputting an MST.

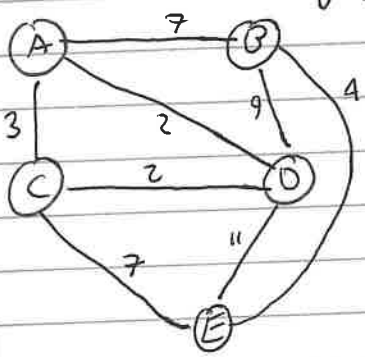
This proof can also lead us to a more efficient formulation. Before, we had to sort ($E \log E$) then check for a cycle ($V+E$) $V-1$ times (V^2). We can put each part of the graph into a "bubble" (i.e. S vs $V-S$. Then by choosing one edge of the cut, there is no need to consider any of the other edges in the cut! We do the "bubbles" with union-find.

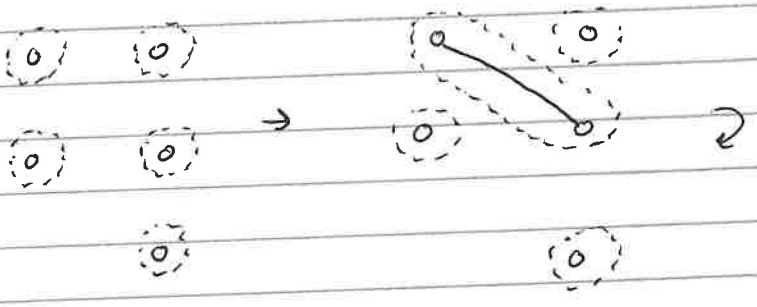
we use union-find as a blackbox with the following operations
makeset(v) $v \to \{v\}$ in $O(1)$ time
find(v) returns set containing v in $O(\log n)$ time
union(set A, set B) Combines returns the sets A, B in $O(\log n)$ time.
we can then implement Kruskals as the following

```
def kruskal(G, w):
    for v ∈ V:                          } O(V)
        makeset(v)
    X = {}
    sort E by weights w                 } E log E
    for each (u,v) ∈ E:                 E
        if find(u) ≠ find(v)            2 log V
            X += {(u,v)}
            union(find(u), find(v))     log V
```

Since $O(\log V) = O(\log E)$ then this kruskals with union find takes $E \log E$.

Consider the following graph. The sorted edges are [2, 2, 3, 4, 7, 7, 9, 11]





1 {A}, {B}, {C}, {D}, {E}
2 {A, D}, {B}, {C}, {E}
3 {A, D, C}, {B}, {E}
4 find(A) = find(C).
5 {A, D, C}, {B, E}
6 {A, D, C, B, E}
7 find(C) = find(E)
8 find(B) = find(D)
9 find(D) = find(E)