

Kruskal's Algorithm

Kruskal(G):

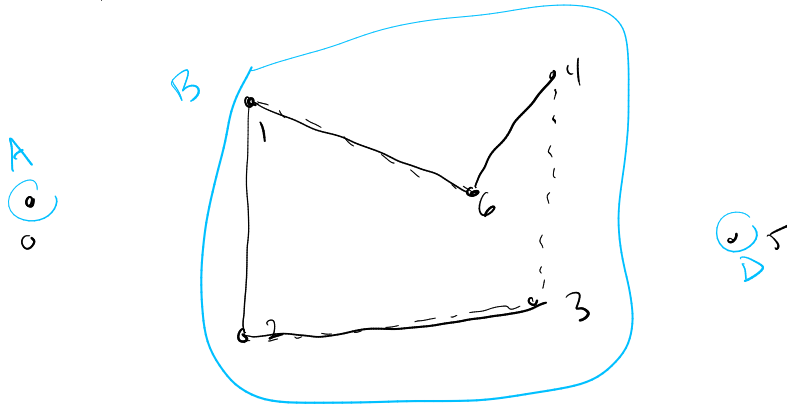
$T \leftarrow$ empty set

Sort edges E by weight

for each edge e (smallest $\rightarrow$ biggest weight):

if e does not make a cycle w/ edges in T , ~~*~~

Add e to T .



So we need a data structure supporting the following operations:

1. Create the data structure w/ every item (vertex) in its own singleton set
2. Query an item to see what set it is in (FIND)
3. Merge two sets (UNION).

This kind of data structure is called disjoint-set or union-find data structure.

Let's rewrite Kruskal in terms of a union-find structure:

Kruskal (G):

$T \leftarrow$ empty set $\Theta(1)$

$U \leftarrow$ new union-find w/ every vertex in singleton set.

Sort edges E $\Theta(E \lg V)$

for every edge $e = (x, y)$ (smallest $\rightarrow$ biggest): $\Theta(E)$ loops (worst case)

if $U.\text{find}(x) \neq U.\text{find}(y)$: $\Theta(E \cdot T_{\text{find}}(V))$ total

Add e to T $\Theta(1)$

$U.\text{union}(x, y)$ $\Theta(V \cdot T_{\text{union}}(V))$ total

Total: $\Theta(1) + \underline{T_{\text{new}}(V)} + \underline{\Theta(E \lg V)} + \underline{\Theta(E \cdot T_{\text{find}}(V))} + \underline{\Theta(V \cdot T_{\text{union}}(V))}$

$\lg V$?

How to implement union-find?

- Represent each set as an array/list of items

- Find: $\Theta(n)$

- Union: $\Theta(1)$ - if linked lists

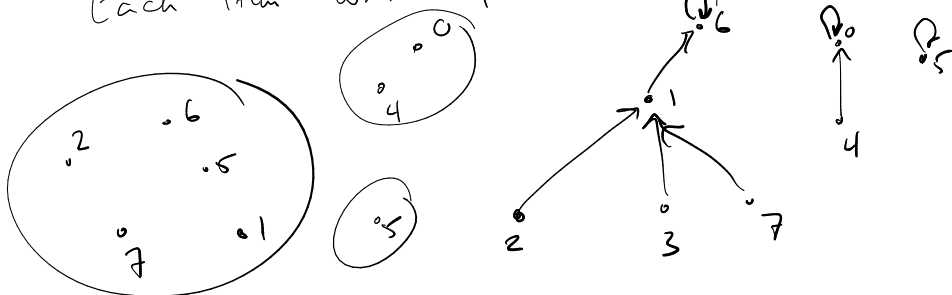
- Dictionary item $\rightarrow$ set label.

- Find: $\Theta(1)$

- Union: $\Theta(n)$

Idea: Arrange the items in each set into a tree.

Each item will map to its parent.



parent dict

0	6	1	1	0	5	6	1
0	1	2	3	4	5	6	7

height dict

1					0	2	
0	1	2	3	4	5	6	7

stores height of roots

- $\text{Find}(x)$: look up $\text{parent}[x]$, $\text{parent}[\text{parent}[x]]$, etc. until finding a root (i.e. some r with $\text{parent}[r]=r$).

Worst case $\Theta(n)$ — how do we make sure trees remain balanced?

- $\text{Union}(x, y)$:

- Find roots of $x + y$
- Make the root of the shorter tree point to the root of the taller tree.
- If heights are same, pick arbitrarily + increment height of the new root.

Claim: If we start w/ all singleton sets and repeatedly call Union , any tree of height h contains at least 2^h items.

Proof By induction on h .

- If $h=0$, the tree is just $\bullet$ which indeed has $\geq 2^0$ items.
- Suppose it's true for trees of height $h-1$. Any tree of height h was created by merging two trees of height $h-1$. By IH, these have $\geq 2^{h-1}$ items, so the merged tree has $\geq 2^{h-1} + 2^{h-1} = 2^h$.

Hence Find + Union are both $\Theta(\lg n)$ in the worst case.