

Kruskal's pseudocode

Wednesday, October 26, 2022 1:11 PM

Notation: (u, v) to denote the edge between u and v

Inputs: E (the collection of edges),
 V (the collection of vertices)

sortedE = E as a sorted list

retTree = empty Graph

while $|retTree.E| < |V| - 1$: <-- what should this end condition be?

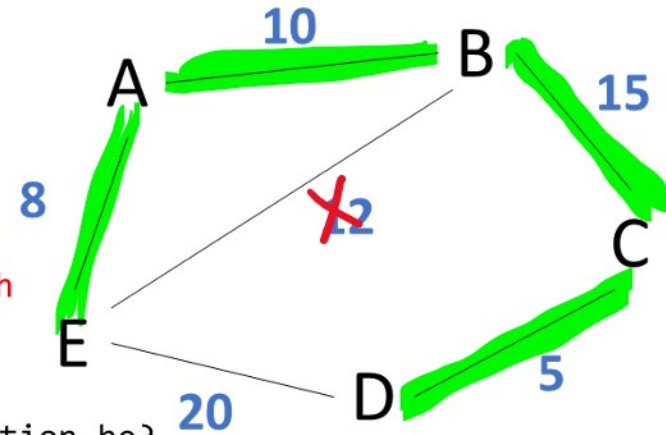
$(u, v) = \text{sortedE.removeFirst}()$

if **! pathExists(v, u)**: *(if there's a path from v to u , and we add the edge (u, v) to the spanning tree, then we have built a path from v back to itself, aka a cycle)*

add (u, v) to retTree

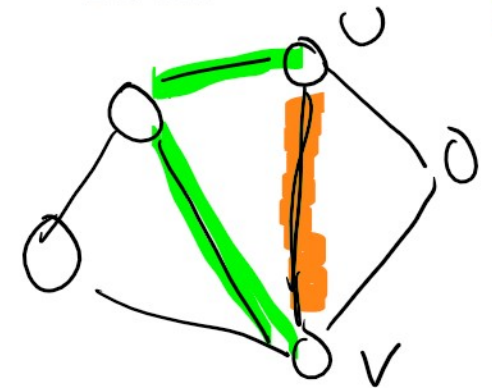
return retTree

A spanning tree on a graph
will always have $V - 1$
edges



What is a cycle?

A non-empty path
(sequence of edges)
from a vertex back to
itself



Kruskal's initial runtime

Friday, October 28, 2022 2:11 PM

Inputs: E (the collection of edges),
 V (the collection of vertices)

sortedE = E as a sorted list $O(|E| \cdot \log |E|)$

retTree = empty Graph $O(1)$

while $|\text{retTree}.E| < |V| - 1$: \leftarrow how many times through loop?
 $|E|$ times

$(u, v) = \text{sortedE.removeFirst()}$ $O(1)$
 $(O(1) \text{ to ask question})$

if ! pathExists(v, u): $O(|E| + |V|) \leftarrow$ BFS or DFS

add (u, v) to retTree

return retTree

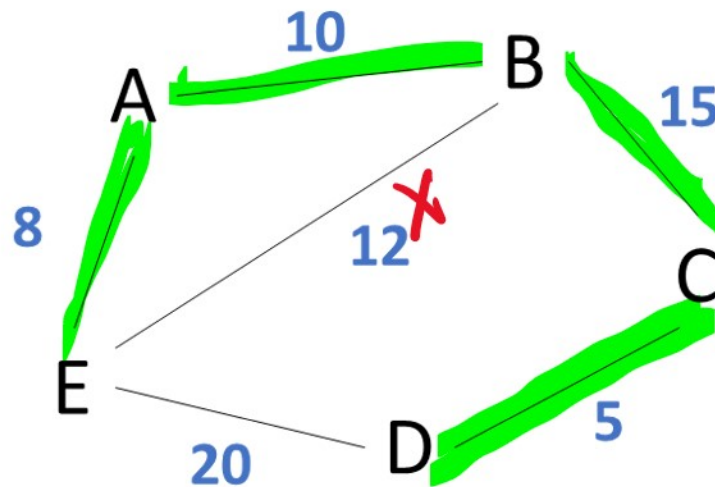
$O(1)$

$O(|E| \log |E| + |E| \cdot (|E| + |V|))$

$= O(|E|^2 + |E| \cdot |V|)$

Optimizing cycle finding

Friday, October 28, 2022 1:48 PM



See notes for detailed, written-out example of optimizing the union/find operations and using them on this graph

A "connected group" of vertices is a collection of edges and vertices that are connected (i.e. all have paths to each other)

(u, v) introduces a cycle into the graph if and only if u and v are in the same connected group

```
Initialize every vertex to be in its own  
unique connected group  
while |retTree.E| < |V| - 1:
```

```
    (u,v) = sortedE.removeFirst()
```

```
    if find(u) != find(v):  
        add (u, v) to retTree  
        union(u, v)
```

```
return retTree
```

union(u, v): combines the connected group that u is in and that v is in into one connected group

find(u): gives back the name of the connected group that u is in