1. Trace Kruskal's MST algorithm. Consider edges by non-decreasing weight; if there is a tie, consider the edge which is first in alphabetic order. In the UF structure, use each root vertex as component label. Each time `find(x)` is called, give the vertex `x` and the number of parent links followed to find the root. If x and y have the same rank, then `union(x,y)` sets `parent[root(x)]` to `root(y)`. List MST edges as they are picked. Update UF parent values as they change.

	B	C	D	E	F	G
A	8	8	9	13	4	11
B		0	9	12	1	18
C			3	7	15	6
D				1	13	9
E					18	6
F						5

```
find( ) calls:
links followed:
MST edges:

union-find data structure:
node   A  B  C  D  E  F  G
parent A  B  C  D  E  F  G
```

2. Repeat the above for Prim's algorithm, starting from vertex G.

```
find( ) calls:
links followed:
MST edges:

union-find data structure:
node   A  B  C  D  E  F  G
parent A  B  C  D  E  F  G
```

3. Prove by induction: in an UF data structure, with union-by-rank, for all integers $t \geq 0$, the number of nodes in a tree with depth d is at least 2^d.