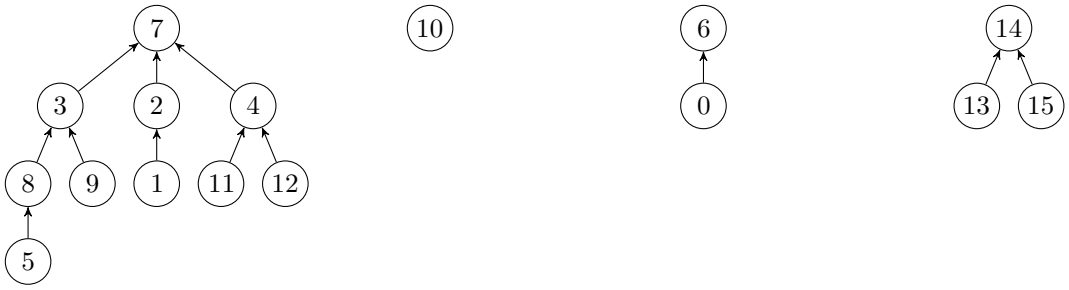
Name & Student Number:

1. Update the following Kruskal's pseudocode with `union` and `find` operations, and do a runtime analysis of the updated code.

```
1:  function Kruskal(Graph G)
2:      initialize disjoint set; call maketSet() on each vertex
3:      sort all edges by weight
4:      for each edge (u, v) in sorted order do
5:          if findSet(u) ≠ findSet(v) then
6:              add edge (u,v) to the MST
7:              union(u,v)
8:          end if
9:      end for
10: end function
```

2. Consider the following disjoint set. Assume that (from left) the first tree has rank 3, the second has rank 0, the third has rank 1, and the last tree has rank 1.



Write the array representation of this disjoint set in the array below.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
6	**2**	**7**	**7**	**7**	**8**	**-2**	**-4**	**3**	**3**	**-1**	**4**	**4**	**14**	**-2**	**14**

3. (3 points) Suppose you are given a connected graph G. Describe how you would figure out if the graph has a cycle. (Answer in at most 3-4 sentences.)

Solution: Run DFS but keep track of vertices in the stack. If we hit a vertex that is already in the stack, then there is a cycle in the graph.
(Another solution is using topological sort, which is a similar idea, but it works only for directed graphs.)