

1. Using Java's ForkJoin framework, write a program to verify that an array of integers is a valid binary min-heap.

To make things easier, start with the given boilerplate code and fill in the compute function.

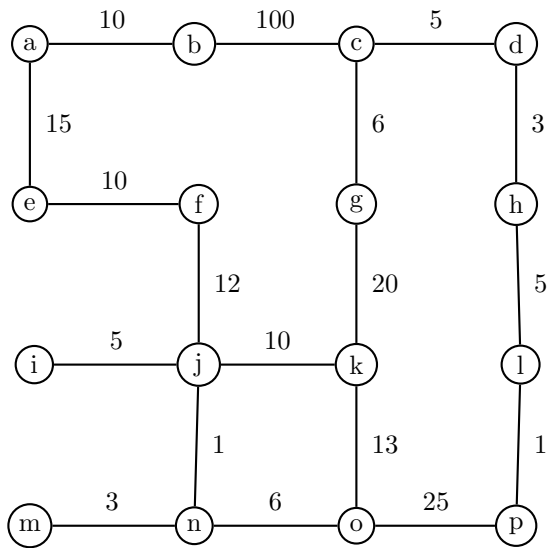
```
import java.util.concurrent.RecursiveTask;

class HeapVerifyTask extends RecursiveTask<Boolean>
{
    final int[] heap;
    final int root;

    HeapVerifyTask(int[] heap, int root)
    {
        this.heap = heap;
        this.root = root;
    }

    public Boolean compute()
    {
        // Fill this in!
        // Return true if the subtree rooted at heap[root] is in fact a heap
    }
}
```

2. Use Kruskal's algorithm to find a minimum spanning tree on the following graph.

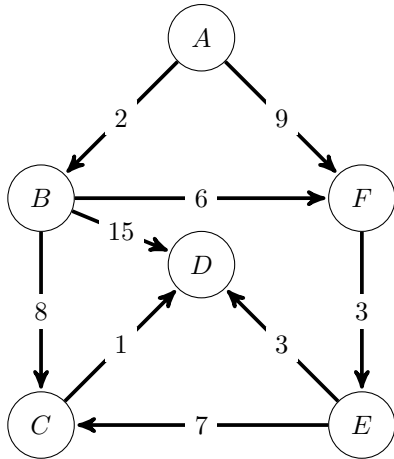


○

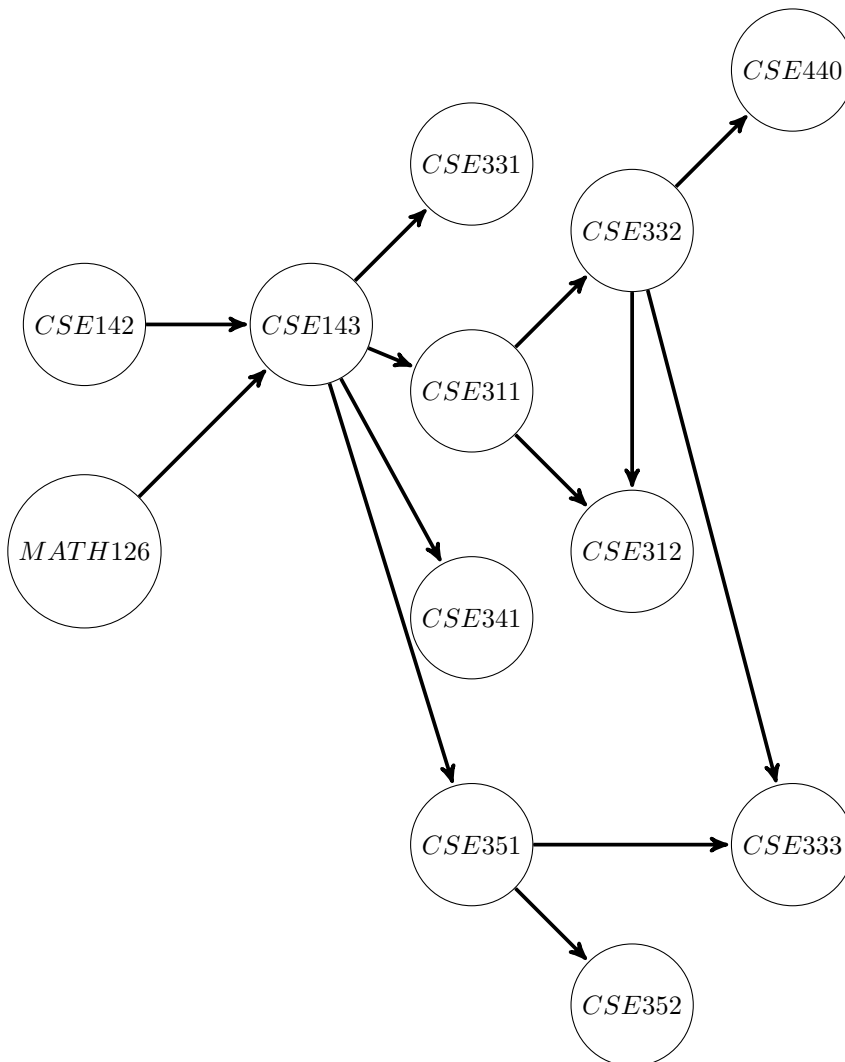
- Starting with a disjoint sets structure containing $1, \dots, 8$, each in their own set, perform the following sequence of union operations using both path-compression and union-by-rank. Show the result the after each step.

$\text{union}(5, 2), \text{union}(6, 3), \text{union}(5, 4), \text{union}(8, 7), \text{union}(7, 5), \text{union}(1, 7)$

4. Find the shortest path from A to every other node.



5. Find a potential ordering of the following classes.



6. Build a heap using Floyd's method containing the values: 4, 9, 15, 3, 10, 2, 23, 100, 1, 21
 Delete min 3 times.
 Insert the values: 92, 40, 3, 6
 Show your work!