Pearploy Chaicharoensin 6381278

HW08

Task 1: Breadth-First Search Running Time

```
Set<Integer> nbrsExcluding(
        UndirectedGraph<Integer> G,
        Set<Integer> vtxes,
         Set<Integer> excl
){
        Set<Integer> union = new TreeSet<>(); // not HashMap //O(1)
        for (Integer src : vtxes) { //It takes n times to run so O(n)*O(logn) = O(nlogn)
                for (Integer dst : G.adj(src)) //O(logn)
                        if (!excl.contains(dst)) { union.add(dst); } // O(logn)
        }
        return union;  //O(1)
}
Set<Integer> bfs(UndirectedGraph<Integer> G, int s) {
        Set<Integer> frontier = new TreeSet<>(Arrays.asList(s)); //O(1)
        Set<Integer> visited = new TreeSet<>(Arrays.asList(s)); //O(1)
        while (!frontier.isEmpty()) {
                frontier = nbrsExcluding(G, frontier, visited); //O(nlogn)
                visited.addAll(frontier); // the i-th position is what's reached at i hops //O(nlogn)
        }
        return visited; //O(1)
}
```

- The total time it takes is O(nlogn), where n is the number of vertices.

Task 2: Mathematical Facts

i)

```
int minSoFar = Integer.MAX_VALUE;
int numUpdate = 0;
for (int i=1;i<=n;i++) {
        if (p(i) < minSoFar) {
                minSoFar = p(i);
                numUpdate++;
        }
}
```

Prove that at the end of the for-loop, $\ln(n+1) \leq E[numUpdate] \leq 1+\ln n$.

- The start is when p(1) and the end of the loop is when p(n).

(ii) LetG=(V,E,w) be an undirected connected weighted graph with distinct edge weights. Show that G has a unique minimum spanning tree.

- If each edge has a distinct edge weight, then each weight will appear in the graph only once, meaning there will be a pair of edges (V,E) and (E,V) that have the same weight (w).

Task3: HackerRank Problems

ID: pearploy_cha