CSE - QUALIFIER EXAM- ALGORITHMS - SPRING 2016
Solve three out of the four problems.

1 Data Structure Design

Design an implementation of the Abstract Data Type "Set" such that the following operations can be performed in $O(\log n)$ time where n is the number of items in the Set:
member(x; S) Return True if and only if $x \in S$.
insert(x; S) Insert x into set S.
delete(k; S) Delete the k-th smallest element from set S.

Please provide concrete details of how the data structure will be maintained after an insert or delete.

2 FLOWS: Escape Routes

We define the Escape Problem as follows. We are given a directed graph $G = (V, E)$ (picture a network of roads). A certain collection of nodes $X \subset V$: are designated as populated nodes, and a certain other collection $S \subset V$ are designated as safe nodes. (Assume that X and S are disjoint.) In case of an emergency, we want evacuation routes from the populated nodes to the safe nodes. A set of evacuation routes is defined as a set of paths in G so that (i) each node in X is the beginning of one path, (ii) the last node on each path lies in S, and (iii) the paths do not share any edges. Such a set of paths gives a way for the occupants of the populated nodes to "escape" to S, without overly congesting any edge in G.

(a) Given G, X, and S, show how to decide in polynomial time whether such a set of evacuation routes exists.

(b) Suppose we have exactly the same problem as in (a), but we want to enforce an even stronger version of the "no congestion" condition (iii). Thus we change (iii) to say "the paths do not share any nodes." With this new condition, show how to decide in polynomial lime whether such a set of evacuation routes exists. Also, provide an example of G, X, and S, in which the answer is yes to the question in (a) but no to the question in (b).

3 APPROX: Variant of BIN-PACKING

Consider the variant of BIN-PACKING where the bin size is b (not necessarily an integer) and each item has size less than $b/3$. This version is still NP-complete, though we won't prove this here. Your problem is to give an algorithm that approximates the optimal packing into bins, and prove a bound on the quality of your approximation. In particular, prove that if your algorithm uses $3a + 1$ bins for some integer a, then the optimal algorithm uses at least $2a + 1$ bins.

4 NPc: Very Independent

Given an undirected graph G = (V,E), let us say that a set $V' \subseteq V$ of vertices is *very independent* if the distance between any two distinct vertices in V' is at least 3. Define the Very Independent Set (VIS) problem as given undirected graph and an integer (G, k) decide whether G has a very independent set of size k. Prove that VIS is NP complete.