Honors Analysis of Algorithms
Practice Exam

Solve all three problems. This is a closed book exam.

Problem 1 (5 points)
Given $k$ lists of integers, each of them sorted and having length $n$, design an efficient algorithm to merge all the $k$ lists into a single sorted list. What is the running time of your algorithm?

Problem 2 (5 points)
Design a simple algorithm, using a max-heap, that finds the $k^{th}$ smallest integer from a given list of $n$ integers. The algorithm should run in time $O(n \cdot \log k)$.

Note: (1) Of course, we discussed in class (and in Homework 1) a recursive algorithm that solves the problem in $O(n)$ time. You are not allowed to use that algorithm (which is too complicated).

(2) Max-heap is similar to the min-heap that we discussed in class, except that a node holds a value that is larger than the values held by its children.

Problem 3 (10 points)
Recall the greedy algorithm for the Minimum Spanning Tree problem. Given a graph $G(V, E)$ with costs on the edges, the algorithm starts with a graph $G'$ with the same vertex set $V$ and no edges. It considers the edges of $G(V, E)$ in increasing order of their costs and at each step, decides whether to add an edge $e = (u, v)$ to $G'$. Let $T_1, T_2, \ldots, T_s$ denote the connected components of $G'$ just before considering the edge $e = (u, v)$. The algorithm needs to perform two operations:

- **Check-Components**: Check whether $u, v$ belong to the same connected component of $G'$. If they do, ignore the edge $e = (u, v)$.

- **Merge-Components**: If $u, v$ belong to different connected components, merge the two connected components into a single connected component and add the edge $e = (u, v)$ to $G'$.

When the algorithm terminates, $G'$ is a minimum spanning tree of $G(V, E)$. We propose a data-structure that allows efficient implementation of the two operations above. Each connected component $T$ is maintained as a rooted tree (denoted again as $T$). Initially, there are $n$ singleton trees, one for each vertex of $G(V, E)$. To merge two trees $T_1$ and $T_2$, supposing that $|T_1| \geq |T_2|$, we make the root of $T_2$ a child of the root of $T_1$, and the root of $T_1$ becomes the root of the merged tree.

- **Show that the height of any tree $T$ is at most $\log |T|$**.

- **Give an upper bound on the time required to implement either of the two operations above. If $G(V, E)$ has $n$ vertices and $m$ edges, what is the total running time to find a minimum spanning tree?**