COMP 3711 Design and Analysis of Algorithms
Fall 2015 Midterm Exam

1. **Short-Answer Questions** (30 pts)

   1.1 (10 pts) Arrange the following functions in asymptotic ascending order (e.g., n, n^2, n^3)

   $$(a)\ n;\ (b)\ \log^9 n;\ (c)\ n^{1.1}/\log n;\ (d)\ n\log n;\ (e)\ 10^{10^{10}}.$$

   1.2 (4 pts) Give at least two scenarios in which you want to use insertion sort instead of quicksort.

   1.3 (4 pts) What do you have to do to show that an algorithm's worst-case running time is $\Omega(n \log n)$?

   1.4 (12 pts) Solve the following recurrences. A correct answer gives you full credits; otherwise, showing the steps may gain you partial credits. Please give the answer using the $\Theta()$ notation. You may assume that n is a power of a for any constant $a > 1$ for your convenience.

   (a) $T(1) = 1$; $T(n) = T(n/2) + 1$ for all $n \geq 2$.
   (b) $T(1) = 1$; $T(n) = 4T(n/2) + n$ for all $n \geq 2$.
   (c) $T(n) = 1$ for $n \leq 3$; $T(n) = T(n/4) + T(3n/4) + n$ for all $n \geq 4$.
   (d) $T(1) = 1$; $T(n) = 2T(n/2) + \log n$ for all $n \geq 2$.

2. **Divide-and-conquer** (20 pts)

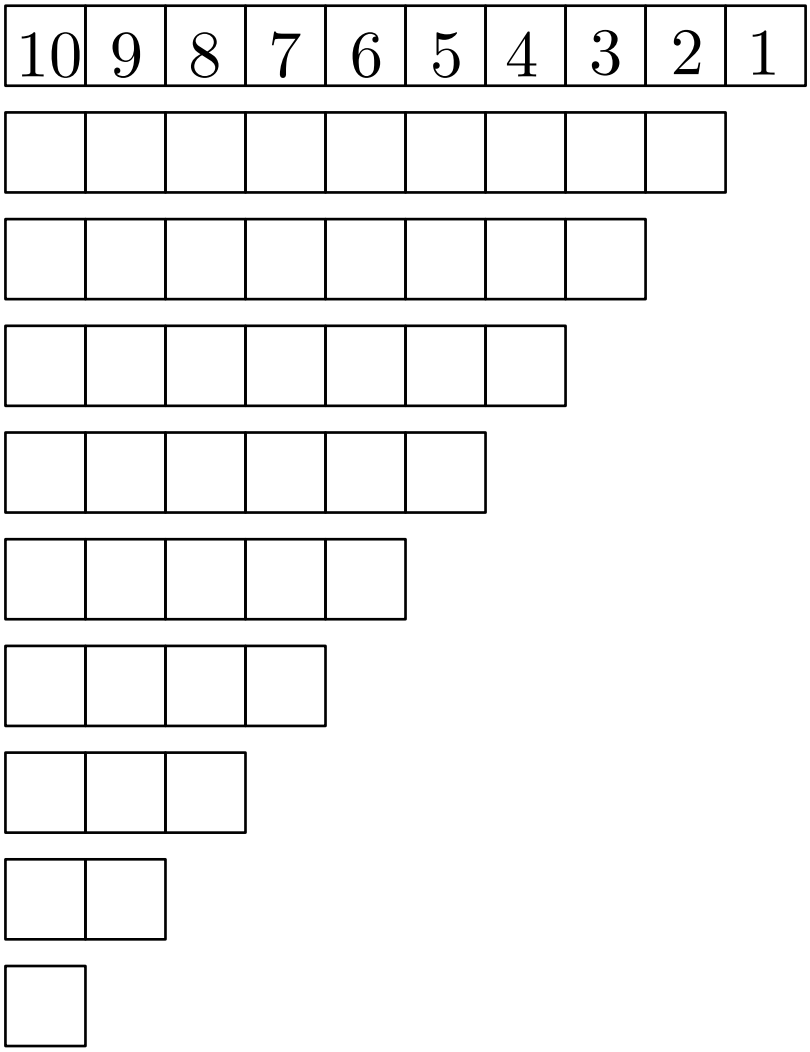
   You are given the prices of a stock over a period of n days. Suppose its price is $p(i)$ on day i, for $i = 1, \ldots, n$. Please design an $O(n \log n)$-time algorithm that finds a strategy to make as much money as possible, i.e., find i, j with $i < j$ such that $p(j) - p(i)$ is maximized. If there is no way to make money, your algorithm should return "no way".

3. **The hat-check problem** (15 pts)

   Recall the hat-check problem in the tutorial: Each of n customers gives a hat to a hat-check person at a restaurant. The hat-check person gives the hats back to the customers in a random order. In the tutorial, you computed the expected number of customers who get back their own hat. Now suppose each hat has a different quality. What is the expected number of customers who get back their own hat or a better one?

4. **Max-heap** (15 pts)

   For the given max-heap stored as an array as follows, show the content of the array after we have run 9 `Extract-Max` operations on it.



5. **Greedy algorithm** (20 pts)

Suppose you and your friends are going on a hiking trip over multiple days. For safety reasons you can only hike during daytime. You can travel at most d kilometers per day, and there are n camping site along the hiking trail where you can make stops at night. Assuming the starting point of the trail is at position $x_0 = 0$, the camping sites are at locations $x_1, \ldots, x_n$, with $x_1 < x_2 < \cdots < x_n$, where x_n is the final destination. Please design an $O(n)$-time algorithm to find a plan that uses the minimum number of days to finish the trip. For example, if $n = 5, d = 5$, and $(x_1, \ldots, x_n) = (4, 5, 6, 9, 12)$, an optimal plan would take 3 days, making stops at camping sites x_1 and x_4. Note that there may be more than one optimal plan; your algorithm just needs to find any one of them. You can assume that $x_{i+1} - x_i \leq d$ for all i (otherwise there is no solution). Remember to prove the correctness of your algorithm.