CPSC 413: Assignment 1

The assignment is due on Oct. 28th. You make work with others in the class, but you must write your solution up individually.

1. (a) Show that if $f = O(g)$, then $g = \Omega(f)$.

   (b) Recall that $n! = n(n-1)(n-2)\dots(2)(1)$. Show that $n! = \Omega(2^n)$.

   (c) How does $n!$ compare to n^n? Which is Big-O of the either? Prove your claim.

   For each of the remaining questions, after descibing your algorithm, prove that the algorithm does return a correct output, and show that its running time is within the desired bound.

2. A toy company has n different toys $t_1, \dots, t_n$ it could make, each with an associated profit P_i. The company has m machines, and each toy has an associated machine M_i on which it needs to be made. Their next shipment can hold $k \leq n$ toys. Each machine can only make one toy before they must be shipped out. Give an $O(n \log n)$ algorithm that determines which toys the company should make so as to maximize their total profit.

3. Over the past several weeks, a bank has noticed that someone is attempting to hack into their computer system. They have n different times when they believe someone was trying to change their data: times t_i. However, this time is approximate, so for each time t_i, they only know that the incident occured somewhere between $t_i - e_i$ and $t_i + e_i$. Now, the company has found someone they believe is responsible, and have a record of when they were accessing the system, at times $x_1, x_2, \dots x_n$. There are many thousands of these incidents, and the company needs to find out if each access x_i from this individual matches up with a unique incident t_j. So, they need to match each x_i to some unique span of time $[t_j - e_j, t_j + e_j]$. Give an algorithm that can determine if such a match exists in $O(n^2)$ time.

4. A number of developers are hoping to develop the land at the edge of a circular lake. Each has requested to develop some portion of the edge of the lake. Design an algorithm that runs in polynomial time and returns the largest set of development requests so that no two overlap.

5. A biologist looks at a sample of n animals, and wants to quickly test whether at least half of them have a common ancestor. He can test whether any two have a common ancestor fairly quickly, but testing all pairs of animals will take $O(n^2)$, and be too slow. Design an algorithm that takes as input a set of animals, and determines if at least half of the group has a common ancestor with a running time of $O(n \log n)$.

6. The police are looking for houses which have particularly large electricity consumption. To simplify the problem, imagine that they are investigating houses which are laid out on an $n \times n$ grid. Each house on the grid has some electricity consumption, $e(i, j)$. The police consider the house suspicious if it has electricity consumption equal to or greater than each of its vertical and horizontal neighbours. Design an algorithm that runs in $O(n)$ time and returns the location of a suspicious house.