Problem 1. Given the runtime of insertion sort algorithm to be $2n^2$ and that of Merge sort algorithm to be $50nlg_2\, n$ running on the same machine. Show the following:

1. On a single plot show the growth curves for the two algorithms as n grows from 1 to 1000.
2. On your plot, show the exact value for n after which a runtime crossover occurs.
3. For which values of n would you prefer each algorithm?

Problem 2. In class we saw Kadane's (dynamic programming) algorithm to find the maximum subarray sum. Write pseudo code to modify it to return the maximum subarray indices and the maximum sum.

Problem 3. Consider the following "comparison" model. There are n red and n blue water jugs for $n \geq 1$. All jugs are of different size and hold different amounts of water. For every red jug, there is a blue jug that holds the same amount of water, and vice versa.

Your task is to find a grouping of the jugs into pairs of red and blue jugs that hold the same amount of water. To do so, you may perform the following operation: pick a pair of jugs in which one is red and one is blue, fill the red jug with water, and then pour the water into the blue jug. This operation will tell you whether the red or blue jug can hold more water, or that they have the same volume. Assume that such a comparison takes one time unit. Your job is to find an algorithm that makes a minimum number of comparisons to determine the grouping. Remember that you may not directly compare two red jugs or two blue jugs.

- Write pseudo code for an $O(n^2)$ brute force algorithm that sorts all the n pairs. Show the exact worst-case analysis of the runtime for this algorithm.
- Describe an efficient algorithm in English that finds the largest pair of jugs. How many exact comparisons would be needed in the worst case?