This homework is due on gradescope Friday November 16th at 11:59pm. Remember to justify your work even if the problem does not explicitly say so. Writing your solutions in \LaTeX is recommended though not required.

**Question 1** (Closest Pair of Points, 35 points). [See also problem 2.32 in the textbook]

Consider the following algorithmic problem: Given \( n \) points \((x_i, y_i)\) in the plane, find the pair of them whose Euclidean distance \( \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \) is as small as possible. There is an obvious \( O(n^2) \) algorithm for this, but with a little effort, you can get it down to near linear time.

**Problem:** Give a full near-linear time algorithm for closest pair of points.

**Extended hint:** An outline of a working algorithm will look something like this:

\[
\text{CPP}(S) \\
\text{If } |S| < 5, \text{ solve problem by brute force} \\
\text{Find a vertical line } L \text{ that divides } S \text{ into near-equally sized sets } S_1 \text{ and } S_2 \\
\text{Run CPP}(S_1), \text{ CPP}(S_2) \text{ and let } d \text{ be the smallest distance found} \\
\text{Let } T \text{ be the set of points in } S \text{ close to } L \\
\text{Sort the points of } T \text{ by } y\text{-coordinate} \\
\text{Compute the distance from } T \text{ to nearby points in this sorted order} \\
\text{if any is less than } d \text{ return the best pair, otherwise return the best pair found in the recursive calls}
\]

When analyzing this problem you will want to consider the following:

(a) How do you find the line \( L \)?

(b) After the recursive calls, we only need to care about pairs of points closer than \( d \) that cross \( L \). You’ll want to show that:

- This only includes points in \( T \).
- This can only include pairs in \( T \) whose \( y\)-coordinates are close in sorted order. To prove this, you may want to first prove that any \( d \times d \) square can contain at most 4 points of \( S_1 \) or \( S_2 \).

(c) You will likely get a recurrence relation of the form

\[
T(n) = 2T(n/2 + O(1)) + O(n \log(n)).
\]

You cannot immediately use the Master Theorem here, but with a little work, you should be able to show that this has runtime \( O(n \log^2 n) \). Alternatively, with a little preprocessing, you can reduce the \( O(n \log(n)) \) in the recurrence above to just \( O(n) \).

**Question 2** (Campaign Strategy, 35 points). Bernadette is running for political office. Her campaign has raised a total of \( n \) dollars which can be spent on any of \( m \) different activities. She must spend a non-negative integer number of dollars on each activity, and the total of these expenditures cannot exceed \( n \). If she spends \( k \) dollars on the \( j \)th activity, she will get \( f_j(k) \) votes from it, for some known functions \( f_j \). These functions \( f_j \) exhibit decreasing marginal votes per dollar spent. In particular, for any \( j \) and \( k \geq 1 \), \( f_j(k+1) - f_j(k) \leq f_j(k) - f_j(k-1) \). Bernadette wishes to allocate her money spending \( k_j \) dollars on the \( j \)th opportunity so that \( \sum_{j=1}^{m} k_j \leq n \) and subject to this, so that \( \sum_{j=1}^{m} f_j(k) \) is as large as possible.
(a) Give a polynomial time greedy algorithm for this problem. [20 points]

(b) Through some further optimization, this runtime can be improved to $O(m \log^2(n))$. Show how to do this. [Hint: you will want to use some kind of binary search.] [15 points]

Question 3 (Job Selection, 30 points). Rajesh runs a freelance design business. Over the course of the next $n$ days he has $m$ job requests. The $i$th request has a deadline $d_i$ that the job must be completed by, and a profit $p_i$ that Rajesh will receive if the job is completed by the deadline (he makes nothing if he takes longer). Each job will take Rajesh exactly one day to complete, and in order to focus on individual projects, he will not work on more than one job in any given day. Give a polynomial time algorithm to compute a schedule for which jobs Rajesh should work on when in order to maximize his total profit.

Question 4 (Extra credit, 1 point). Approximately how much time did you spend on this homework?