22C:31 Exam 2

Due via ICON dropbox by 5 pm, Friday, Oct 30th

You should feel free to use your textbook and notes. However, you should not seek help on the internet or from any other person. By turning in your exam, you are confirming that you've completed your exam within these constraints. You may send me e-mail seeking clarifications; these I can provide, but I will not provide any hints. The exam is worth 150 points, 80 points for Problem 1 and 70 for Problem 2. This translates to 15% of your grade.

1. **On Dynamic Programming.** Consider the problem of neatly printing a paragraph on a printer. The input text is a sequence of n words of lengths $\ell_1, \ell_2, \ldots, \ell_n$, measured in characters. We want to print this paragraph neatly on a number of lines that hold a maximum of M characters each. Our criterion for "neatness" is as follows. If a given line contains words i through j, where $i \leq j$, and we leave exactly one space between words, the extra number of space characters at the end of the line is $M - (j - i) - \sum_{k=i}^{j} \ell_k$, which must be nonnegative so that the words fit on the line. We wish to minimize the sum over all lines, except the last, of the cubes of the numbers of extra space characters at the ends of lines. Give a dynamic programming algorihm to print a paragraph of n words neatly on a printer. Analyze the running time and space requirements of your algorithm.

   Example. If you have printed the paragraph in three lines, with 5 blanks at the end of line 1, 7 blanks at the end of line 2, and 54 blanks at the end line 3, then the "cost" of this solution is $5^3 + 7^3 = 468$. If somehow you could adjust the words in the lines so that 6 blanks were left at the end of the each of the first two lines, then the "cost" of your solution would be $6^3 + 6^3 = 432$. So the latter is a better solution, if we use this measure of neatness.

   How should you present your solution? Follow the step-by-step method that I described for describing a dynamic programming solution. If you do this you will not need an explicit proof of correctness. However, I do want you to explicitly discuss (i) the running time of your algorithm and (ii) the space requirements of your algorithm.

2. Problem 6 at the end of Chapter 5. This "divide-and-conquer" problem is actually quite easy given that you have thought hard and long about Problem 7!

   How should you present your solution? After your present your divide-and-conquer algorithm, you should provide (i) a proof of correctness, i.e., a proof showing that your algorithm does indeed always return a local minimum and (ii) a clear argument showing that your algorithm uses $O(\log n)$ probes.
