

## General Instructions

1. For general advice on homework, homework lateness policy, and honor code, refer to the *Course Syllabus* (on the CS 25 web page), which we discussed at the beginning of the term.
2. When describing an algorithm, you can use pseudocode or English, whichever is convenient. The important thing is that the description be both unambiguous and clear to the reader.
3. **Whenever** you are asked to give an algorithm, even if the problem does not explicitly ask for all of the following three parts, **you must supply** all of the following three parts:
  - (a) The algorithm.
  - (b) Proof of correctness of the algorithm, i.e., a clear and convincing argument that the algorithm is correct.
  - (c) Analysis of its running time.

Please note that all three components are assigned points in our grading guide and so an algorithm, even if correct, gets little credit unless the other components are also supplied.

## Special Instructions for Dynamic Programming problems

When solving a dynamic programming problem, be sure to include all of the following steps:

1. Describe clearly the optimal substructure of the problem (i.e., how a solution to a problem can be constructed from solutions to smaller subproblems).
2. Introduce appropriate notation to clearly state what the general version of the problem is. If, for instance, you introduce  $b[i, j]$ , you must clearly state the meaning of the value in  $b[i, j]$ .
3. Using the notation that you introduced, state the recurrence, including all the base cases.
4. Give the pseudocode that outputs an optimal solution.
5. Analyze the time and space requirements of your algorithm. **Note** that a less efficient algorithm gets a lot fewer points, even if correct.

- 
1. (10 points) Exercise 14.3-3
  2. (15 points) Exercise 14.3-6
  3. (20 points) Problem 15.4
  4. (10 points) Exercise 15.4-4
  5. (20 points) Problem 15.6
  6. (25 points) Problem 15.3