

Algorithmics Screen Exam, Fall 2006.

This question concerns an algorithm of the following form:

Algorithm $R(n)$:

Step 1. Using the value of n select a method for a problem of size n and apply the selected method. (Most of the available methods will be recursive, but at least one will not be.)

Step 2. Combine the answers from the recursive calls to produce the final answer.

The running time for such an algorithm is the time for the recursive calls plus some overhead for Steps 1 and 2. For this question we have the following methods:

- A. Direct (no recursion), overhead 2^n (requires $n \geq 1$).
- B. Two recursive calls of size $n - 1$, overhead n (requires $n \geq 2$).
- C. Two recursive calls, one of size $n - 1$, one of size $n - 3$, overhead n (requires $n \geq 4$).
- D. Two recursive calls of size $n - 2$ overhead $10n$ (requires $n \geq 3$).

This general specification needs some additional details, particularly which method to use for each n . For Parts 2 and 4 below, the algorithm should be using the best (fastest) method for each n . Thus for those parts, you will need to specify for each n just which method should be used as part of working out the problem. (For partial credit, you could pick a method without claiming that it is the best methods and work out the time for the method you picked.) For Part 3, you have a simpler rule for choosing the method.

Below is a list of things to do concerning these methods. Do as many of these as you have time for. Make your reasoning clear so that the grader can tell that you know how to solve these kinds of problems.

1. It is clear from simple considerations that one or two of these methods should not be used for any large values of n . Indicate those one or two methods and explain the simple considerations. (The more methods you cover here, the less you will need to cover in Part 3.)
2. For $1 \leq n \leq 6$ give the method that the optimum algorithm will use for that value of n and the time needed. Probably the easiest and clearest way to solve this part of the question is to compute the time needed by each method and then indicate which method has the smallest time.
3. For each of methods B, C, and D consider an algorithm which uses only that method for all large n and method A for small n . Give a big O analysis for each of these three algorithms. In other words, give a big O analysis for each of methods B, C, and D.
4. Give an exact analysis of the time needed by the best method for this problem. Clearly, the optimum algorithm for this problem is an algorithm that chooses the best method for each n . It is essential that your analysis corresponds to the algorithm you specify. It is best if the algorithm you specify is the most efficient algorithm possible.