Big 1: Divide and Conquer

Submitted by:

Collaboration: You are encouraged to collaborate with up to 5 other students, but all work submitted must be Whiteboard only and write-ups must be composed completely independently. List the names of all of your collaborators. Do not seek published solutions for any assignments. If you use any published resources (other than those provided by the course) when completing this assignment, be sure to cite them. Do not submit a solution that you are unable to explain orally to a member of the course staff.

Problem 1: Solving Recurrences

Solve, i.e. provide a tight Θ (big-Theta) bound for, the following recurrences using the indicated method. You may use any base cases you'd like.

1. $T(n) = 37T(n/23) + n$ (using Master Theorem)
2. $T(n) = T(n/2) + T(n/4) + T(n/8) + n$ (using Guess and Check)
3. $T(n) = T(n/2) + 2$ (using Master Theorem)

Problem 2: An Imposter Among Us

You have been called in to solve the space crime of the century!. There is a spaceship with n crewmates, but exactly one among them is an impostor. While everyone is out performing their tasks, the impostor has been stealthily sabotaging their efforts. They have now called a meeting to identify the impostor, and have requested your help. You will do so by privately asking crewmates whether they have "seen" another crewmate performing a task or not. All crewmates will answer this question honestly, but note that while a crewmate A might have seen another crewmate B perform a task, this does not necessarily mean that B saw A completing a task. You know that *none* of the crewmates has ever seen the impostor complete a task, and that the impostor will claim to have seen each of the crewmates complete a task. Design a **divide-and-conquer** algorithm to identify the impostor by asking $O(n)$ questions to the crewmates.

Divide Step. Describe the divide step of your algorithm here, making sure to mention what subproblems are produced.

Combine Step. Describe the combine step of your algorithm here.

Algorithm. Describe your whole algorithm here. You do not need to repeat the procedures for divide and combine, you're welcome to simply reference them here.

Correctness. Prove the correctness of your algorithm. Namely, show that for all groups of size n with exactly 1 impostor, your algorithm successfully identifies the impostor.

Number of Questions. Express the number of questions your algorithm asks as a recurrence relation. Describe, using the master theorem, how you know that the solution of this recurrence is $O(n)$.

Problem 3: Fast Exponentiation Returns

Computer graphics software typically represents points in n dimensions as $n+1$-dimensional vectors. To make transformations of the graphics (e.g. rotating the modelled figure, zooming in, or making the figure appear as though seen through a fish-eye lens) we have a $(n+1)\times(n+1)$ matrix which defines the transformation, and then we multiply each point by this matrix to transform the point. Let's say we are developing software for very high dimension graphics (n dimensions), and we have a transformation (defined by a $(n+1)\times(n+1)$ matrix) that we would like to apply to a particular point n times. Develop an algorithm which can apply this transformation n times in $o(n^3)$ (little-oh of n^3) time.

In other words, if T is a $(n+1)\times(n+1)$ matrix, give an algorithm which calculates T^n in $o(n^3)$ (little-oh of n^3) time.

Divide Step. Describe the divide step of your algorithm here, making sure to mention what subproblems are produced. Include the total number of operations required for the divide step.

Combine Step. Describe the combine step of your algorithm here. Include the total number of operations required for this combine step

Algorithm. Describe your whole algorithm here. You do not need to repeat the procedures for divide and combine, you're welcome to simply reference them here.

Correctness. Prove the correctness of your algorithm. Namely, show that your algorithm produces the correct answer.

Running time. Express the number of operations your algorithm requires as a recurrence relation. Describe, using the master theorem, how you know that this recurrence is $o(n^3)$

Problem 4: Flights

You have been hired to plan the flights for Professor Floryan's brand new passenger air company, "Receding Airlines". You are going to provide service to n cities. This airline will only service the United States, and will only fly you East.

You recognize that in order to enable all your passengers to travel from any city to any other city (to the East) with a single flight requires $\Omega(n^2)$ different routes. Prof. Floryan says that the airline cannot be profitable when supporting so many routes. Another option would be to order the cities in a list (from West to East), and have flights that go from the city at index i, to the city at index $i+1$. This, however, would mean some passengers would require $\Omega(n)$ connections to get to their destination. Devise a compromise set of routes which requires no passenger have more than a single connection (i.e. must take at most two flights), and requires no more than $O(n \log n)$ routes. Prove that it satisfies these requirements.

Divide Step. Describe the divide step of your algorithm here, making sure to mention what subproblems are produced. Include the total number of routes created during the divide step.

Combine Step. Describe the combine step of your algorithm here. Include the total number of routes created during this combine step.

Algorithm. Describe your whole algorithm here. You do not need to repeat the procedures for divide and combine, you're welcome to simply reference them here.

Correctness. Prove the correctness of your algorithm. Namely, show that the selected routes have the property that one can travel from any city to an eastward city with at most a single connection.

Number of Routes. Express the number of routes your algorithm selects as a recurrence relation. Describe, using the master theorem, how you know that this recurrence is $O(n \log n)$.

Source Declarations

I Collaborated with the following people when completing this assignment:

I Consulted the following resources when completing this assignment: