CSE 3500 – Algorithms and Complexity
Homework 6

Question 1 (40 points)

Find the solution to the given recurrence. If applicable, make use of the Master Theorem. Present the process step by step.

(a) $T(n) = 16T(n/4) + n^2$
(b) $T(n) = 7T(n/2) + n^2$
(c) $T(n) = 4T(n/2) + n^2\sqrt{n}$
(d) $T(n) = T(n-1) + 1$

Question 2 (20 points)

You are given an array where the values in the array increase up to a certain position p (peak point) and then decrease from position p to the end of the array.

Example:

Input: Array A = [1, 2, 3, 4, 5, 4, 3, 2, 1]

Output: Peak point: 5

Your goal is to write a divide and conquer algorithm to identify the peak point p. Your algorithm should achieve a time complexity better than $O(n)$, where n is the number of elements in the array.

a. Write a clear divide and conquer algorithm to solve the problem.

b. Provide the recurrence relation that describes the time complexity of your algorithm.

c. Analyze the time complexity of your algorithm. If applicable, use the Master Theorem.

Question 3 (40 points)

The maximum sum subarray problem involves finding the contiguous subarray within an array of integers that has the largest sum. Given an array of integers, you need to design a divide and conquer algorithm to find the maximum sum of a subarray.

Example input:

0	1	2	3	4	5	6	7
-2	-5	6	-2	-3	1	6	-6

Output: maximum sum subarray = 8 (from index 2 to index 6)

Task: Research the algorithm and make sure that you fully understand the problem:

a. Provide a high-level explanation of how the divide and conquer approach is used to tackle this problem.
b. Write the algorithm.
c. Provide the recurrence relation that describes the time complexity of your algorithm.
d. Analyze the time complexity of your algorithm. If applicable, use the Master Theorem.

Additional Guidelines:

- You can assume that the input is a one-dimensional array of integers.
- Your algorithm should divide the problem into smaller subproblems, solve them recursively, and combine their results to find the maximum subarray.
- Pay close attention to the divide and merge steps in your algorithm.
- Explain the key components of your algorithm and how it ensures the correct maximum subarray is found.