45 Lecture - CS301

Important Subjective

1. **What is divide and conquer paradigm?**

   Divide and conquer is a problem-solving technique in computer science that involves breaking down a problem into subproblems, solving these subproblems independently, and then combining their solutions to solve the original problem.

2. **What are the steps involved in the divide and conquer algorithm?**

   The steps involved in the divide and conquer algorithm are:
   - Divide the problem into smaller subproblems.
   - Solve each subproblem independently.
   - Combine the solutions of the subproblems to solve the original problem.

3. **How does merge sort work using the divide and conquer paradigm?**

   Merge sort works using the divide and conquer paradigm by:
   - Dividing the array to be sorted into two halves.
   - Sorting each half recursively using merge sort.
   - Merging the sorted halves into a single sorted array.

4. **What is the time complexity of quicksort algorithm?**

   The time complexity of quicksort algorithm is O(n log n) on average, and O(n^2) in the worst case.

5. **What is the base case in the divide and conquer paradigm?**

   The base case in the divide and conquer paradigm is the smallest possible input that can be solved without further recursion.

6. **What is the difference between top-down and bottom-up approaches in the divide and conquer paradigm?**

   Top-down approach starts with the full problem and divides it into smaller subproblems, while bottom-up approach starts with the smallest subproblems and combines them into a solution for the full problem.

7. **What are the advantages of using divide and conquer paradigm in algorithm design?**

   The advantages of using divide and conquer paradigm in algorithm design are:
   - It provides a clear and structured approach to solving complex problems.
   - It can lead to more efficient algorithms by reducing the problem size and avoiding redundant computations.

It allows for parallel processing of subproblems.

8. **What are the disadvantages of using divide and conquer paradigm in algorithm design?**

The disadvantages of using divide and conquer paradigm in algorithm design are:
It may lead to increased memory usage due to the recursive calls.
It may not always be the most efficient approach for certain types of problems.
It can be difficult to implement and debug.

9. **What is the divide and conquer approach for finding the maximum subarray?**

The divide and conquer approach for finding the maximum subarray involves dividing the array into two halves, finding the maximum subarrays in each half recursively, and then combining them to find the maximum subarray that crosses the middle.

10. **What is the recurrence relation for the time complexity of a typical divide and conquer algorithm?**

The recurrence relation for the time complexity of a typical divide and conquer algorithm is $T(n) = aT(n/b) + f(n)$, where a is the number of subproblems, n/b is the size of each subproblem, and f(n) is the time taken to divide the problem into subproblems and combine their solutions.