DATA STRUCTURES AND ALGORITHM DESIGN

NPTEL WEEK - 3

1) Which of the following is always true for an unsorted array with distinct elements?

- ○ There exists at least one local minimum
- ○ The first element is always a local minimum
- ○ There can be no local minima
- ○ The middle element is always a local minimum

2) In Euclid's algorithm for computing $\gcd(a, b)$, what operation is repeatedly applied?

- ○ Replace (a, b) with $(b, a \bmod b)$
- ○ Replace (a, b) with $(a - b, b)$
- ○ Replace (a, b) with $(a + b, a)$
- ○ Replace (a, b) with $(a, a \bmod b)$

3) Analyze the time complexity of the following algorithm called Euclid's algorithm for GCD of two numbers. You may assume that each instruction of this algorithm takes $O(1)$ time for execution.

```
GCD(a,b)    // here a is greater than or equal to b.
{
  while b <> 0
  {
      t = b
      b = a mod b
      a = t
  }
  return a
}
```

- ○ $O(a)$
- ○ $O(a \log b)$
- ○ $O(\log a + \log b)$
- ○ $O(\log a + 1)$

4) What is the time complexity of finding a local minimum in a 1D array using divide and conquer?

- ○ $O(\log n)$
- ○ $O(n)$
- ○ $O(n \log n)$
- ○ $O(\sqrt{n})$

5) Given the recurrence $T(n) = 2T(n/2) + n$, what is the asymptotic time complexity?

- ○ $O(n \log n)$
- ○ $O(n^2)$
- ○ $O(n)$
- ○ $O(\log n)$

6) Consider the recurrence $T(n) = 2T(n/2) + \sqrt{n}$. What is the time complexity?

- $O(n)$
- $O(n \log n)$
- $O(n^{1.5})$
- $O(\sqrt{n} \log n)$

7) A matrix algorithm divides an $n \times n$ matrix into 4 equal submatrices and processes each recursively. It performs $O(n^2)$ additional work. What is the recurrence?

- $T(n) = 4T(n/2) + O(n^2)$
- $T(n) = 2T(n/2) + O(n \log n)$
- $T(n) = T(n/2) + O(n^2)$
- $T(n) = 4T(n/2) + O(n)$

8) In the divide-and-conquer algorithm for finding a local minimum in an $n \times n$ matrix, the algorithm scans the middle column to find its minimum and then recurses on one half of the matrix. What is the recurrence relation for its time complexity?

- ○ $T(n) = T(n/2) + O(n)$
- ○ $T(n) = 2T(n/2) + O(n)$
- ○ $T(n) = T(n-1) + O(1)$
- ○ $T(n) = T(n/2) + O(\log n)$

9) The divide-and-conquer algorithm for finding a local minimum in an $n \times n$ matrix runs in $O(n)$ time. Which of the following best explains this?

- ◯ The total number of elements processed across all levels shrinks geometrically and sums to $O(n)$
- ◯ The algorithm reduces both dimensions simultaneously at each step, which leads to $O(n)$ time by halving both rows and columns
- ◯ Even though each level takes $O(n)$ time and there are $O(\log n)$ levels, the matrix structure forces the total to be linear
- ◯ Since only one column and one row are scanned at each step, total cost is just $O(\log n)$

10) Which of the following problems is most naturally solved using divide-and-conquer?

- ○ Finding the closest pair of points in a plane
- ○ Finding the shortest path in a graph
- ○ Selecting activities to maximize profit
- ○ Computing the nth Fibonacci number with memoization