Question1 :

Merge sort:   
Merge Sort is a popular sorting algorithm, follows the divide-and-conquer paradigm:

1. **Divide**: Divide the unsorted array into two halves until each sub-array contains only one element.
2. **Conquer**: Recursively sort each sub-array.
3. **Combine**: Merge the sorted sub-arrays to produce the final sorted array.
4. **Base Case**: If the array has fewer than two elements, it is already sorted.
5. **Divide**: Split the array into two roughly equal halves.
6. **Conquer**: Recursively apply Merge Sort to each half.
7. **Combine (Merge)**: Merge the two sorted halves into a single sorted array.

(DAC)(Problem):

If (small(problem):

{solve(p)}

Else :

Divide p into p1 ,p2,p3,p4

# recurive

Apply DAC(P1), DAC (p2), DAC (p3)

Combine ( (DAC(P1)), DAC(p2)), DAC (P3)))

}

}

Merge sort is a divide-and-conquer algorithm that recursively breaks down a list into smaller sublists until each sublist consists of only one element. It then merges these sublists in a sorted manner.

Maximum and Minumum:

1. **Base Case**: If the array has only one element, it is both the maximum and minimum.
2. **Divide**: Split the array into two roughly equal halves.
3. **Conquer**: Recursively find the maximum and minimum in each half.
4. **Combine**: Compare the maximum and minimum of the two halves to determine the overall maximum and minimum