

## Assignment – 2

- (1) Given a sorted array (we do not know either ascending / descending) of size  $n$ . Find the maximum/minimum element in this array.
- (2) Given a sorted array of size  $n$  in ascending order. Take an element from the user as an input and find its position in the sorted array using BINARY SEARCH if present. Do the same with  $\alpha$ -SEARCH and  $\beta$ -SEARCH.  $\alpha$ -SEARCH checks for  $2/3^{rd}$  element in the sorted array unlike BINARY SEARCH where  $1/2^{th}$  element is checked, and  $\beta$ -SEARCH checks for  $1/4^{th}$  element.
- (3) Binary search divide the array in two parts and searches in one of the part. Let there is a searching approach named TERNARY search which divides the array into three parts (nearly equal) and searches in one of the part depending on the Key value (value to be search). Implement this.
- (4) In merge sort, an array of size  $n$  is recursively divided into two sub-arrays of size  $n/2$  to sort it. Implement a merge sort which divides the array of size  $n$  recursively into four sub-arrays of size  $n/4$  to sort it. You need to implement two variations of this merge sort.
  - In `Merge()` procedure, use the `Merge()` which you know to merge two sorted arrays.
  - Write a separate `Merge()` procedure which merges four sorted arrays. Do not consider merging of two sorted array as a procedure in this process.