1.

Algorithm sum(arr)

 input: array to add elements from

 output: sum of the elements of the array

 return sumHelper(arr, 0)

Algorithm sumHelper(arr, index)

  input: array and index the initial index to start sum

  output: sum of each element of the array

  if index = arr.length then return 0

  return arr[index] +sumHelper(arr , index+1)

O(n)

2.

Algorithm findMax(arr)

  input: array to check max value From

  output: max value

  return maxhelper(arr,0);

Algorithm findMaxhelper(arr, index)

  input: array and index to see which has maximum value

  output: the maximum value in the array

 toCheck <- findMaxhelper(arr, index+1)

  if index =  arr.length then return 0;

  if arr[index]  > toCheck then return arr[index]

  else return toCheck

O(n)

3.

Algorithm sum(arr)

  input: sorted arr with 0 and 1 - elements

  output: sum up with O(log n)

  i<-arr.length-1

  sum<-0

  while arr[i]>0 do

    sum=sum+arr[i]

    i=i-1;

  return sum