

Algorithm and Programming (UE-101)

Assignment 4

1. Give a divide-and-conquer algorithm to find the average of all numbers in an array $A[1..n]$ of integers, where n is a power of 2. The algorithm should start by dividing the input elements into approximately two halves. How much work space is required by the algorithm? Analyze the time complexity of the algorithm.
2. Consider the following modification of Algorithm mergesort. The algorithm first divides the input array $A[low..high]$ into four parts A_1, A_2, A_3 , and A_4 instead of two. It then sorts each part recursively, and finally merges the four sorted parts to obtain the original array in sorted order. Assume for simplicity that n is a power of 4. (a) Write out the modified algorithm. (b) Analyze its running time.
3. Show that 2^n is not $\theta(3^n)$.
4. Find the time complexity of the algorithm $fun(A, 0, n - 1)$ where A is an array of size n , and the function is defined as follows:

```
fun(A,p,r){  
    if(p<r){  
        q=bar(A,p,r);  
        fun(A,p,q-1);  
        fun(A,q+1,r);  
    }  
}  
  
bar(A,p,r){  
    return r-1; }
```