Data Structures and Algorithms : CS210A

Practice-sheet 1

Time complexity and Efficient Algorithms

1. What is the time complexity of each of the following subroutines as a function of N ? Give proper reasons.

   (a)
   ```
   int sum = 0;
   for (int i = 1; i < N; i *= 2)
      for(int j = 0; j < i; j++)
         sum++;
   ```

   (b)
   ```
   int sum = 0;
   for (int i = 1; i < N; i *= 2)
      for (int j = 0; j < N; j++)
         sum++;
   ```

2. There is an array A of size n storing a sequence of 0's followed by all 1's. Design an efficient algorithm to compute the smallest index in A containing a 1.

3. You are given 2 sorted arrays A and B each of size n. Design an $O(n)$ algorithm to merge them to make another sorted array C of size $2n$.

4. Primality problem is to determine whether a given positive integer is prime. Consider the following algorithm for Primality problem. Assume that the input integer n is greater than 2.

   ```
   IsPrime(n)
   {   x <- 2;
       isPrime <- true;
       while(x*x <=n and IsPrime)
       {     If(n mod x = 0) IsPrime <- false;
             x <- x+1;
       }
       If (IsPrime) print (n is a prime number)
       Else print (n is NOT a prime number)
   }
   ```

   What is its time complexity ?

   Optional: Do you know about the connection between IIT Kanpur and the Primality problem ? If not, find it out from the web.

5. For each of the following cases determine whether $f(n) = O(g(n))$ or $f(n) \neq O(g(n))$. You must provide formal arguments in support of your answer.

   (a) $f(n) = 10 \log_2 n$ and $g(n) = \log_{10} n$

   (b) $f(n) = 5^{n/2}$ and $g(n) = 2^n$

   (c) $f(n) = 2^{\sqrt{\log n}}$ and $g(n) = \sqrt{n}$

   (d) $f(n) = 100 \sum_{i=1}^{n} \frac{1}{i}$ and $g(n) = \log_{100} n$

6. For each of the following questions, answer Yes or No appropriately. Provide suitable arguments in support of your answer.

   (a) Let $f(n) = \lceil \log n \rceil !$ and $g(n) = n^{10}$. Is $f(n) = O(g(n))$?

   (b) Let $f(n) = O(g(n))$. Can we conclude that $2^{f(n)} = O(2^{g(n)})$?

   (c) For any two increasing functions $f(n)$ and $g(n)$, is it true that either $f(n) = O(g(n))$ or $g(n) = O(f(n))$?

7. Give an algorithm to find the smallest and the second smallest elements from a list of N items using the minimum number of comparisons. Note that here we are interested in the exact number and not an upper bound in terms of big "O" notation.

8. You are given an array A storing n numbers and another number x. You need to determine the subarray of A whose sum is closest to x. Design the fastest possible algorithm for this problem. Your algorithm is allowed to use $O(n)$ extra space.

9. Given two arrays storing n numbers, design an algorithm to print out all elements that appear in both arrays. The output should be in sorted order.

10. Given a positive integer n and a list containing $n - 1$ distinct integers in the range $[1, n]$, design an $O(n)$ time algorithm to find the missing number. You are not allowed to modify the list even temporarily. Your algorithm is allowed to use only $O(1)$ extra space. However, you may assume that every arithmetic operation takes $O(1)$ time.

11. Analyze the following algorithm called Euclid's algorithm for GCD of two numbers. Mention its time complexity as a function of input size (number of bits of `a` and number of bits in `b`).

```
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
}
```