

Practice-sheet 1

Date: 18 August, 2012.

1. What is the time complexity of each of the following subroutine 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. Given an array 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. Give an algorithm to find the smallest and 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.
4. 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.
5. 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.
6. 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 less than or equal to b.
{
    while b  0
    {
        t <- b
        b <- a mod b
        a <- t
    }
    return a
}
```

7. Write an iterative as well as recursive function in C for reversing a singly linked list.
8. Given two singly linked lists storing n elements in ascending order. Transform these lists into one list which also contains elements in ascending order.
9. You are given the head of a singly linked list. You need to determine if this list loops (a node whose next pointer points to some node appearing earlier in the list) somewhere or not. Design the most efficient algorithm for this problem.
10. Given an array A storing n numbers, design an $O(n)$ time algorithm to find the maximum value of $A[j] - A[i]$ where $j \geq i$.
11. Given a $n \times n$ matrix of 0s and 1s such that in each row no 0 comes before a 1, design an algorithm to find the row with the most 0s. The algorithm should run in $O(n)$ time.

