Tutorial 2: CS21003 Algorithms I

Prof. Partha Pratim Chakrabarti and Palash Dey
Indian Institute of Technology, Kharagpur

January 21, 20201

1. Let A be a $n \times n$ two-dimensional array, in which all the rows and all the columns are sorted in ascending order from smaller to larger indices. Given a key x, your task is to find out whether x is an element of A. Develop an efficient algorithm for the same and analyse the complexity.

2. Let $X = \{x_1, x_2, ..., x_n\}$ be an array of n positive integers and a is an integer. Propose an efficient algorithm to determine whether there are two elements in X whose sum is exactly a. Derive the time-complexity of the proposed algorithm.

3. Analyze the complexity of the algorithm:

```
func(n){
  k=0, y=2
  while (kin){
    y=y*y, k++
  }
  return y
}
```

4. Let A be an array of n integers. In this exercise, we will show that the expected number of comparisons made by the randomized quick sort algorithm to sort A is $\mathcal{O}(n \log n)$.

   (a) Write a pseudo-code for randomized quick sort.

   (b) Write a pseudo-code for the partition algorithm where the pivot element is involved in every comparisons.

   (c) For every $1 \leqslant i < j \leqslant n$, define an indicator random variable $Z_{i,j}$ for the event that $A[i]$ and $A[j]$ are compared with each other by the algorithm.

   (d) Compute $\mathbb{E}[Z_{i,j}]$

   (e) Define a random variable Z to be the number of pairs of elements compared by the randomized quick sort algorithm to sort A. Write Z in terms of $Z_{i,j}, 1 \leqslant i < j \leqslant n$

   (f) Compute $\mathbb{E}[Z]$