- Always try to give algorithm with best possible running time. The points that you obtain will depend on the running time of your algorithm. For example, a student who gives an $O(n)$ algorithm will receive more points than a student who gives an $O(n^2)$ algorithm.
- You are required to give proofs of correctness whenever needed. For example, if you give a greedy algorithm for some problem, then you should also give a proof why this algorithm outputs optimal solution.
- **Use of unfair means will be severely penalized.**

There are 3 questions for a total of 50 points.

---

(10) 1. There are n jobs and one machine. Each job i has a processing time of $p(i)$, that denotes the time that the machine has to spend in doing job i. A schedule is an order in which these jobs are performed. The waiting time of a job as per a given schedule is the time at which this job is completed by the machine.

Design an algorithm to output a schedule that minimizes the sum of waiting time of all jobs.

(20) 2. You are given three sorted lists A, B and C of numbers. A contains l numbers, B contains m numbers, and C contains n numbers. You have to find the k^{th} smallest number in the union of these three lists. Design an algorithm and discuss its running time.

(20) 3. An array containing n elements is said to have a majority element if more than half (i.e., $> \lfloor n/2 \rfloor$) of the elements of the array are the same. You are given an array of some objects. Suppose these objects do not have a relative order defined on them but checking if two objects are the same is simple and can be done in constant time. Design a divide-and-conquer algorithm that outputs the majority element, in case there is one. Discuss running time of your algorithm.