- 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.

---

(20) 1. You are a Tavel Agent and you have information regarding all flights for a day. The information is in the form of a 4-tuple. The i^{th} flight information is denoted by (s_i, d_i, p_i, q_i) with the following meaning:
- s_i denotes the source city.
- d_i denotes the destination city.
- p_i denotes the time at which the flight leaves source city s_i.
- q_i denotes the time at which the flight lands in the destination city d_i.

You need to figure out a travel plan from a given source city to a destination city such that as per the plan you will reach the destination city as early as possible. The additional constraint you have to maintain is that you should have a time gap of at least 1 hour between connecting flights at any intermediate city.

(20) 2. Recall the Fractional Knapsack problem discussed in the class. Suppose we remove the "fractional" assumption from the problem. That is, for every item, you can either take the item or not take it (recall in the fractional version, you were allowed to any any arbitrary fraction of the item). Consider the following greedy algorithm for the problem.
GreedySteal

- Let $\{i_1, ..., i_n\}$ be the items sorted in decreasing order of value per unit volume (i.e., $V(i)/W(i)$).
- Let T be the item with maximum value.
- Let k be such that $\sum_{i=1}^{k} W(i) \leq W$ and $\sum_{i=1}^{k+1} W(i) > W$.
- Either steal items $\{i_1, ..., i_k\}$ or item T whichever maximizes the total value.

Let G be the value of the items chosen by the greedy algorithm and let OPT be the value of the optimal solution for the above problem. Show that $G \geq (1/2) \cdot OPT$.

(10) 3. Let S and T be sorted arrays each containing n elements. Give an algorithm to find the n^{th} smallest element out of the $2n$ elements in S and T.