**CS2040S: Data Structures and Algorithms**

# Exercises/Optional Problems for Week 2

*For: February 10, 2020*

**Problem 1. Kattis's Quest**

Taken from: `https://open.kattis.com/problems/gcpc`

One hundred years from now, in 2117, the International Collegiate Programming Contest (of which the NCPC is a part) has expanded significantly and it is now the Galactic Collegiate Programming Contest (GCPC).

This year there are $n$ teams in the contest. The teams are numbered $1, 2, \ldots, n$, and your favorite team has number 1.

Like today, the score of a team is a pair of integers $(a, b)$ where $a$ is the number of solved problems and $b$, is the total penalty of that team. When a team solves a problem there is some associated penalty (not necessarily calculated in the same way as in the NCPC – the precise details are not important in this problem). The total penalty of a team is the sum of the penalties for the solved problems of the team.

Consider two teams $t1$ and $t2$ whose scores are $(a1, b1)$ and $(a2, b2)$. The score of team $t1$ is better than that of $t2$ if either $a1 > a2$, or if $a1 = a2$ and $b1 < b2$. The rank of a team is $k + 1$ where $k$ is the number of teams whose score is better.

You would like to follow the performance of your favorite team. Unfortunately, the organizers of GCPC do not provide a scoreboard. Instead, they send a message immediately whenever a team solves a problem.

Your job is to is to make use of data structures to support the following function:

- `int update(int team, long newPenalty)` Given a team, increments its number of solve by one, and returns the rank of team 1, after this new event. Should be done in $O(\log n)$ time.

For example, consider the following sequence:

1. update(2, 7) (returns 2)
2. update(3, 5) (returns 3)
3. update(1, 6) (returns 2)
4. update(1, 9) (returns 1)