

## Problem Set 2 - Data Structures & Paradigms

### Problem A

Classrooms, if I remember correctly, was relatively easy. I remember in one of my tutorials, one Daniel Khuu mentioned to look at interval stabbing from the lectures, and so I did. I think this gave me the idea to sort the classrooms by endpoints, and allocate from there.

I believe the algorithm was to sort the activities by end point - a greedy means to make sure we always choose the class that allows us the most possible allocations in the future, and then iterate through all of the classrooms.

We allocate by pushing the start time of the element we've added into a multiset. If the set is full (i.e. all classrooms were occupied, we check if the current end time is larger than the start time of any of the elements in the multiset - if they are, then they don't overlap, and we can allocate another. Everytime something is added to the set, we add one to the count of classrooms that could be allocated.

One key thing here that took me a while to get was that we didn't just de-allocate any of the classrooms off the set, we would deallocate the one that ended closest to the start time of our current one, in order to allow for more allocations in the future - another greedy principle. In the code, this was done with the upper bound function.