

# CS170 GargSack WriteUp

Jason Shan, Dhruv Malik

May 2, 2017

The objective of this algorithm is to give an approximation to a set of problems that may or may not have optimal solutions. Due to the size of these files, we want to run our solution in such a way so that we do not break the constraints of memory or risk timing out. Thus, our proposal will to run a few different greedy strategies sorted by different heuristics, and see what our maximum value will be.

To make things non-deterministic, we will introduce randomness to a few of our cases. Our cases are relatively simple. They are:

- Randomize classes, and then pick our classes in such a way so that we do not break runtime constraints. We will run this many times to try to, by chance, get the most-optimal set we can. Once we get this set, we know that nothing in this set will conflict in terms of classes, so we can run a simple greedy on this.
- Run greedy algorithm based on our multiple sorting heuristics, that involve some subset of weight, cost, resell, class, etc.

Our reasoning is this. Primarily, we will use the first case, and we will run that randomized case between 20-200 times. The purpose of this is that we want to, by chance, be able to select some assortment of classes such that we will find some optimal set. But there could exist a case where we fail to do so, and that is where our simple-random solution comes into play.