Kick Start 2020 - Round A

Analysis: Allocation

We want to buy as many as possible houses. Intuitively, we can keep buying the cheapest house. The rationale is to save money at each step so we could buy more in the end. One way to implement this strategy is to sort all the houses by prices from low to high and then buy houses one by one until we run out of money.

The sorting part has O(**N** log **N**) time complexity and the processing part has O(**N**) time complexity. Using counting sort could reduce the sorting complexity to O(**N**) since the range of the prices is [1, 1000]. The overall time complexity is O(**N**).

Let's prove the correctness of this greedy algorithm. Let the solution produced by the greedy algorithm be **A** = {a_1, a_2, ..., a_k} and an optimal solution **O** = {o_1, o_2, ..., o_m}.

If **O** and **A** are the same, we are done with the proof. Let's assume that there is at least one element o_j in **O** that is not present in **A**. Because we always take the smallest element from the original set, we know that any element that is not in **A** is greater than or equal to any a_i in **A**. We could replace o_j with the absent element in **A** without worsening the solution, because there will always be element in **A** that is not in **O**. We then increased number of elements in common between **A** and **O**, hence we can repeat this operation only finite number of times. We could repeat this process until all the elements in **O** are elements in **A**. Therefore, **A** is as good as any optimal solution.