

assignment 4

task 1

a) halving algorithm

with an expert that will never make any mistake, we have $M = 0$ where M is how many mistakes the best expert makes after T days. then let m^t be the amount of mistakes the algorithm makes after t days. we want to find m^T .

since experts can only be right or wrong, and we are following the majority of experts, that means if we are wrong, then the majority is wrong and removed, meaning at least half of the current set of experts is removed. if we are correct, then a minority is removed from the set, at most one half.

counting how many times we are wrong then becomes a process of counting how many times we can halve the set before we are left with only the expert that makes no mistakes. this is $\log_2 n$ of course, since we are halving the n experts.

this is the upper bound for how many mistakes we will make, since we may get lucky and remove every expert other than the super-expert, as shown by the fact that we may remove more than half if we are wrong. that means we would only make the right choice from there on out, thus only netting us a single mistake, $m = 1 \leq \log_2 n$ which is true for any $n \geq 0$, but $n = 0$ is uninteresting.

b) no superexpert

the algorithm performs halving, but readds all experts if all should be removed. this means there will be one or more best experts that make at most m^* mistakes.

when there is no such expert that never makes mistakes, we can see that the algorithm makes at most $(1 + m^*)(1 + \log_2 n) - 1$ mistakes, where $m^* = \min m_i^T$.

once again, when the algorithm makes a mistake, at least half of the experts are booted, and similarly for when we choose correctly. however, we have no guarantee this time, thus we could end up booting all experts at the first turn. as such, we need

haven't touched this since before easter so i'm not gonna bother finishing it