Models for e-commerce
HW2 - MAB

Our algorithm can be split into two sections:

1. We implemented an algorithm that makes sure all the arms' thresholds are met, while also learning the ERM matrix in order to be exploited in later rounds.
   We were able to do that since we know that the distribution is Uniform with parameters 0, and $b_{i,j}$, in order to estimate the parameter $b_{i,j}$ we referred to the Maximum Likelihood Estimator for the uniform distribution by updating our local parameter:
   $$b_t = max\{b_{t-1}, reward_t\}$$
   We saved these parameters in a matrix, and we wanted to know when the Matrix's values are "stable". We did that by calculating the Frobenius norm. Due to the max's function characteristics, for some time periods the values would be stable but would change after a couple of more rounds which made it hard for us to determine whether the matrix has in some sense "converged", so we decided to use "moving average" on the norms of the matrix in previous rounds with a window of 200, and we decided that the matrix has converged if the difference between last two consecutive values is less than 10^{-5}.
   Now we have a pretty good estimation of the ERM matrix so we move to the next part of the algorithm.

2. We created a function called `find_best_arms` the function takes in the configuration of the simulation, and number of rounds (we halt the real simulation while this function is working).
   Now we iterate over all the subsets of arms in order to find the best set of arms that will only be used in the remaining rounds, it does so by performing a `FakeSimulation` exactly as the `MABSimulation` class would with 20,000 rounds in order to get an estimation of how "good" a set of arms is, and then returns the best set of arms.
   We then update the Planner (back to the first section) to **only** use the best arms (while maintaining their threshold) and use the previous information to maximize the reward as much as possible.