CS218 HW 2 Challenge

due Thursday, April 25, 23:59 PM

Problem A // ID: 256063713
To simulate the counting process, and to get the minimum counting needed, use a min heap to pull the smallest two bags at a time. When combining, add double the total candies in these bags and add that combined bag back to the heap. Then, the algorithm terminates when all the bags have been merged.
Runtime: $O(n * log(n))$ due to loop running $n - 1$ times and adding and removing elements from a heap takes $O(log(n))$ time; Space Complexity: $O(n)$, for all n bags in the heap.

Problem B // ID: 256073268
First, sort the professor free days and student ending free days in order ascending. Then, work through the student free days to find the earliest professor free day to assign to it. After going through all student days, the maximum number of students the professor can assist is found.
Runtime: $O(n * log(n) + m * log(m) + m * log(n))$ due to the professor and student free days being sorted in $O(n * log(n))$ and $O(m * log(m))$ time respectively and the loop running m times with the bisect left method using binary search in $O(log(n))$ time to find free day matches for $O(m * log(n))$; Space Complexity: $O(1)$, for constant space used.

Problem C // ID: 256210654
First, fill a dictionary indexed by row keys with set values of each furniture that exists in that row. Then, enter a loop that only breaks when each row has $\leq k$ furniture in it. If there is at least one row that has $> k$ furniture, choose the furniture that appears the most in all the rows. Then, remove the most common furniture from all sets, and continue the loop.
Runtime: $O(L^2 + n * L)$ due to looping through the whole grid $O(L^2)$ and the most common value loop through n furniture and L rows in $O(n * L)$ time; Space Complexity: $O(L^2)$, for storing all the furniture in the dictionary.

Problem D // ID: 256593261
Keep a set for the trains destroyed and list of integers size m representing the number of trains destroyed for each attack. Loop through each attack and train in a nested loop, then determine which trains are destroyed based on the given formula: i not in trains destroyed and $|x_i - y_j| \leq b_j$ and $D_i \leq A_j$.
Runtime: $O(m * n)$ due nested looping m attacks and n trains; Space Complexity: $O(n + m)$, for the $O(n)$ trains destroyed and $O(m)$ attacks.