

Session

A very straightforward greedy problem. Consider if you proceed the string from left to right, you find an 'o' and the rest time is long enough to take this session, if you don't take it then:

- There is no 'o' afterward or,
- When you take the next 'o', your rest time will last longer with the same number of taken sessions.

Then the answer is to take the session as long as it is feasible.

Respond

Binary search the maximum confusion. Consider that you find some value mid and you try to check if the maximum confusion is larger than mid or not. For each student i , we can find the latest time to answer her question if the maximum confusion is at most mid , that is $(mid - x_i)/y_i$. Then we sort the latest time for all students we can easily check if there is any contradiction: reject value mid if there is more than t time smaller than t for any $t \in \{0, \dots, n - 1\}$

Subset

You may try to solve the question like this: design the state as $dp[i][j][k]$ means the answer for the first i item and j is the sum of numbers in the first subset and k is the sum of the second subset. The problem with this solution is **there too much redundant state**. We don't need to mind the sum of the first subset and the second subset, we only care about **the difference between the bigger subset sum and the smaller subset sum**.

We design the state as $dp[i][j]$ means the maximum sum of the bigger subset of the first i item and the difference of the bigger sum and the smaller sum is j . Transition equation is:

$$\begin{aligned} dp[i][j] = \max(& dp[i-1][j], && \text{means that you could ignore the item } a_i \\ & dp[i-1][j+a_i], && \text{or add it into the smaller subset.} \\ & dp[i-1][j-a_i] + a_i, && \text{or add it into the bigger subset (if } j \geq a_i) \\ & dp[i-1][a_i-j] + j) && \text{add into bigger subset if } j < a_i \end{aligned}$$

Finally $dp[n][0]$ will be the maximum m . Then the answer is $sum - dp[n][0]$.