

Max Debt

Problem Explanation

We are given a list of friend's names and the amount of money borrowed each time. Now, for a particular friend we remember the maximum 3 amounts borrowed from him.

Now, we are asked to find from which friend we have borrowed the maximum total amount (i.e. summation of maximum 3 amounts corresponding to that particular friend). If there are multiple such friends we have to choose the lexicographically smallest name.

A name s is lexicographically smaller than t , if and only if $s_i < t_i$ for the first i where s_i and t_i differ.

Approach

For each user, map them in a hashmap and store the corresponding amount in a vector. Set a variable that keeps track of the maximum amount and initialize to -1. Now, sort the vector in descending order and for each user calculate the sum of maximum 3 amounts. Compare this to the maximum amount stored in the variable previously defined. If the sum is greater than the maximum answer then update the answer.

Note:

1. Handle the case carefully when the total size of vector for a particular user is less than 3.
2. Map stores the key in ascending order so there is no need to check for the case when the sum calculated is equal to the maximum answer.

Time Complexity: $O(M N \log N)$

Space Complexity: $O(M N)$

Where M is the number of distinct friends and N is the maximum number of times the money is borrowed from a particular friend.