

Amazing Test - Hackerearth

Wednesday, March 25, 2020 8:30 PM

The Problem:

Akshara is a Maths teacher at Dynamic Public School. One day she decided to take an unusual test of all her students. She took all her students to a fair. There she took them to a candy room. The room had 2 doors and behind each door was unlimited supply of candies. The excitement of the students could not be measured. Each student wanted a specific amount of candies. Now here was the test.

At a time 1 student could enter 1 door. After he exits another will enter. No time is lost when there is change of students. For processing 1 candy it took 1 second. The teacher gave exactly X amount of time for all the students to get their specific amount of candies. So the students ask you for help in determining whether it is possible for all the students to get their candies or not in X amount of time.

Input:

The first line contains T denoting the number of test cases. Each test case consists of two lines containing n and x denoting the number of students and the time given by the teacher. The next line contains n space separated numbers denoting the amount of candy each student wants.

Output:

For each test case print YES if it is possible or NO.

Constraints:

$1 \leq T \leq 25$
 $1 \leq N \leq 100$
 $1 \leq x \leq 100000$
 $0 \leq A[i] \leq 100$

SAMPLE INPUT

```
2
3 4
2 4 2
3 3
3 3 3
```

SAMPLE OUTPUT

```
YES
NO
```

Explanation

In the first test case:

At t=0 Student 3 enters door 1 and student 2 enters door 2.

At t=2 Student 1 enters door 1 and student 3 exits.

At t=4 Student 2 and 1 exits.

The Code:

```
/* I took a little help on this one. Seems pretty simple now.
 * The logic: Treat it as a binary knapsack problem. Only, there are two bags to
 * be filled now.
 * Maximize one bag. Subtract this bag's sum with the total sum. If this diff
 * < timeLimit, then YES!
 * In binary knapsack we maximize for one bag anyway. Just the difference logic is
 * extra.
 */
#include <stdio.h>
#define MAX 100001
#define max(a,b) \
({ __typeof__ (a) _a = (a); \
__typeof__ (b) _b = (b); \
_a > _b ? _a : _b; })
int main(){
    int caseCount, timeLimit, toffeeCount, toffie, i, j, prev, curr;
    int toffies[100];
    long int totalSum = 0;
    scanf("%d", &caseCount);
    long int** maxToffies = (long int**)malloc(2 * sizeof(long int*));
    maxToffies[0] = (long int*)malloc(MAX * sizeof(long int));
    maxToffies[1] = (long int*)malloc(MAX * sizeof(long int));

    while(caseCount > 0) {
        scanf("%d %d", &toffeeCount, &timeLimit);

        totalSum = 0;
        for(i=0; i<MAX; i++) {
            maxToffies[0][i] = 0;
            maxToffies[1][i] = 0;
        }

        for(i=0; i<toffeeCount; i++) {
            scanf("%d", &toffies[i]);
            totalSum += toffies[i];
        }

        for(i=0; i<toffeeCount; i++) {
            curr = i % 2;
            prev = abs(1-curr);
            toffie = toffies[i];
            for(j=toffie; j<=timeLimit; j++) {
                maxToffies[curr][j] = max( maxToffies[prev][j], maxToffies[prev]
[j-toffie] + toffie);
            }
        }
        if((totalSum - maxToffies[curr][timeLimit]) <= timeLimit ) {
            printf("YES\n");
        } else {
            printf("NO\n");
        }
        caseCount--;
    }
}
```

The Stats:

Score

30.0

Time (sec)

0.61

Memory (KiB)

64

Language

C