CodeChef

NAME:DIVYESH KHUNT **SAPID:60009210116**

divyesh_486

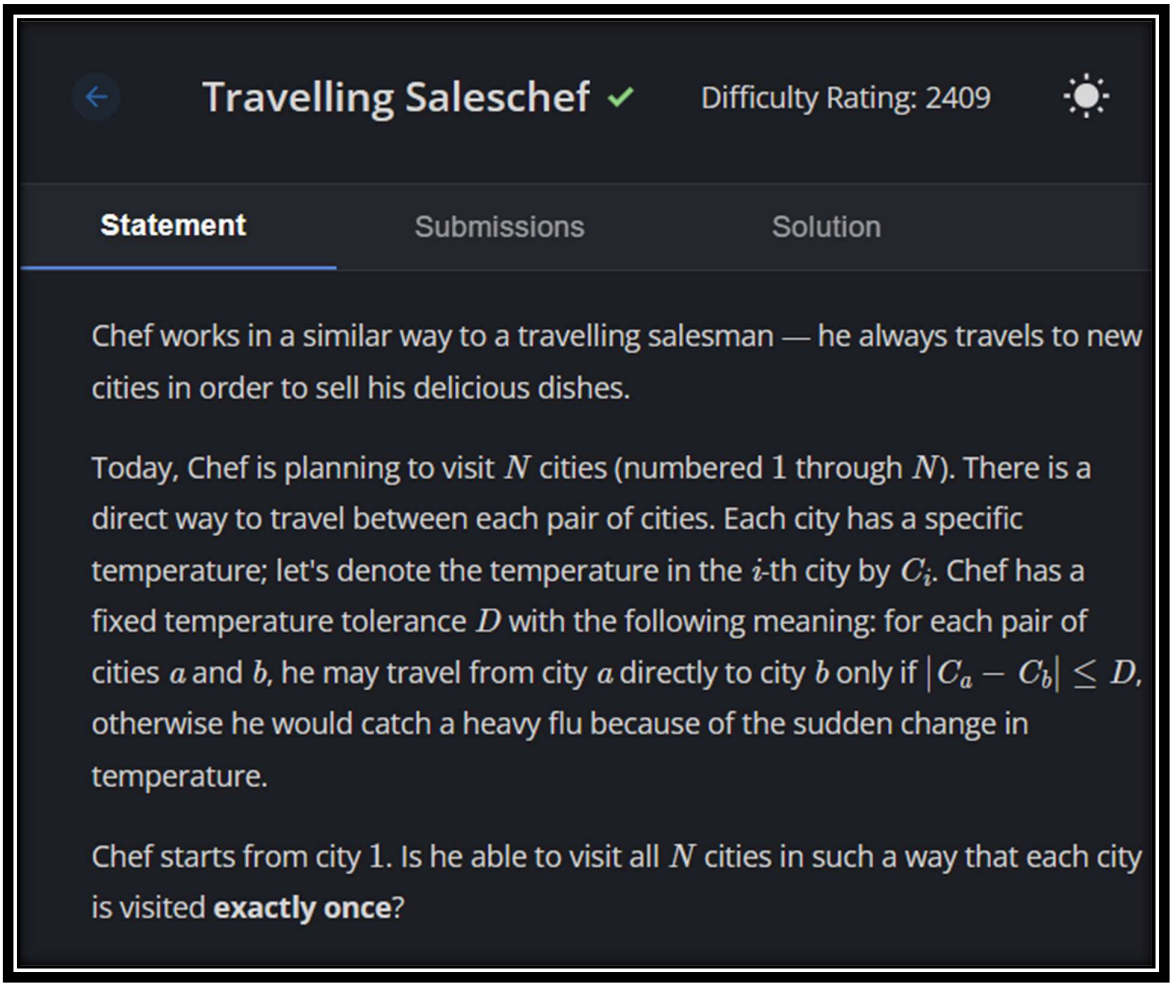
Username:	divyesh_486
Country:	India
Student/Professional:	Student
Institution:	Dwarkadas Jivanlal Sanghvi College of Engineering Vile Parle, Mumbai, Maharashtra, India

Problem Solver - Bronze Badge

107 / 250

Solve 143 more Problems to get Silver Badge

PROBLEM



Travelling Saleschef ✔ Difficulty Rating: 2409

Statement Submissions Solution

Chef works in a similar way to a travelling salesman — he always travels to new cities in order to sell his delicious dishes.

Today, Chef is planning to visit N cities (numbered 1 through N). There is a direct way to travel between each pair of cities. Each city has a specific temperature; let's denote the temperature in the i-th city by C_i. Chef has a fixed temperature tolerance D with the following meaning: for each pair of cities a and b, he may travel from city a directly to city b only if $|C_a - C_b| \leq D$, otherwise he would catch a heavy flu because of the sudden change in temperature.

Chef starts from city 1. Is he able to visit all N cities in such a way that each city is visited **exactly once**?

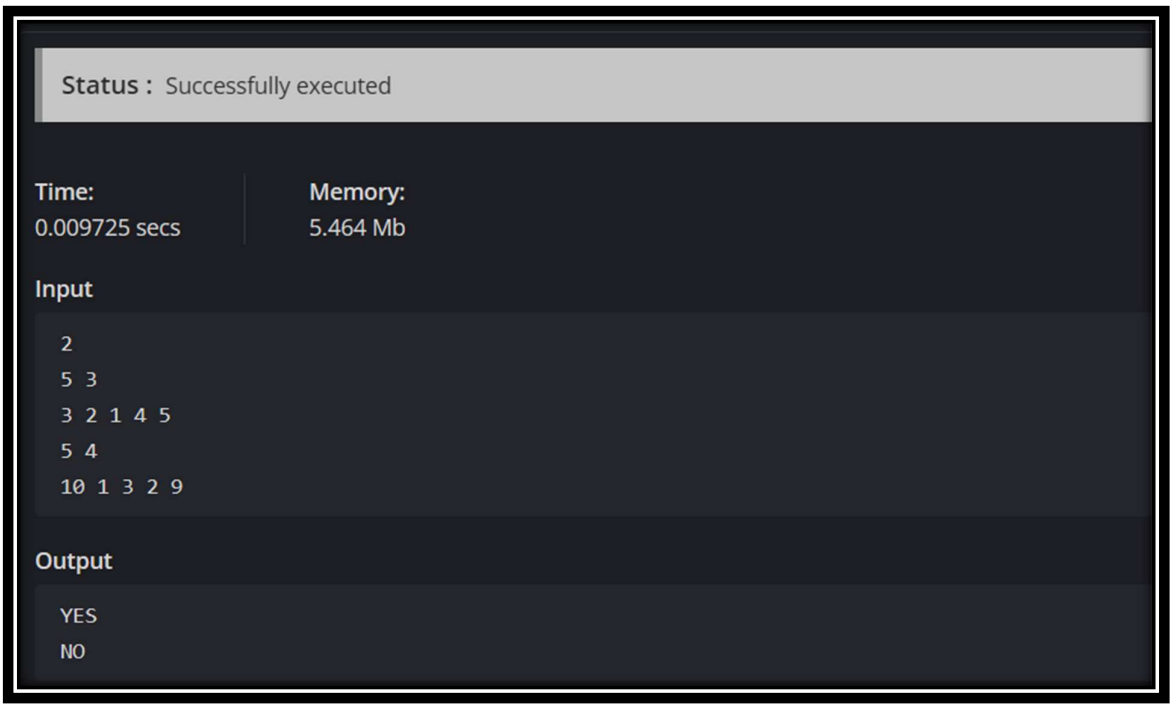
CODE:

C++14

```
#include<algorithm>
#include <iostream>
using namespace std;
int main()
{int T;
cin>>T;
int A[T][100000],C[T][100000],N[T], D[T],F[T];
for(int x=0;x<=T-1;x++){
    cin>>N[x]>>D[x];
    for(int i=0;i<N[x];i++){
    cin>>A[x][i];
    }
    F[x]=A[x][0];
    int* p;
    p=&A[x][0];
    sort(p,p+N[x]);
}

    for(int x=0;x<=T-1;x++){
    int z=0;
   for(int m=0;m<N[x]-1;m++){
    if(A[x][m+1]-A[x][m]>D[x]){
    cout<<"NO" ;goto X;}
   }
    for(int q=1;q<N[x]-1;q++){
    if(A[x][q+1]-A[x][q-1]>D[x]){
    C[x][z]=A[x][q];
    z++;}
    }
    if(z==0){
        cout<<"YES";}
    else {
    if(F[x]<C[x][0]){
        cout<<"YES";}
    else if(F[x]>C[x][z-1]){
        cout<<"YES";}
    else {
        cout<<"NO";}
    }
 X: cout<<endl;
  }
  return 0;
}
```

OUTPUT:



HackerRank

PROBLEM:

Problem

Submissions

Leaderboard

ussions

Given a number of dollars and an array of denominations of coins, determine how many ways you can make change. For example, making change for $n = 12$ dollars from coin denominations $coins = [1, 2, 5, 10]$, there are 15 ways to make change:

```
1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 2
1 1 1 1 1 1 1 1 2 2
1 1 1 1 1 1 2 2 2
1 1 1 1 2 2 2 2
1 1 2 2 2 2 2
2 2 2 2 2 2
1 1 5 5
2 5 5
1 1 10
2 10
2 2 2 1 5
2 2 1 1 1 5
2 1 1 1 1 1 5
1 1 1 1 1 1 1 5
```

Hints:

- You can solve this problem recursively, but you must optimize your solution to eliminate overlapping subproblems using Dynamic Programming if you wish to pass all test cases. More specifically, think of ways to store the checked solutions and use the stored values to avoid repeatedly calculating the same values.
- Think about the degenerate cases:
  - How many ways can you make change for 0 dollars?
  - How many ways can you make change for less than 0 dollars if you have no coins?
- If you are having trouble defining the storage for your precomputed values, then think about it in terms of the base case $(n = 0)$.

CODE:

```c
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>

unsigned long int coinchange(int W,int n)
{
    int a[n];
    int i,w,j;
    unsigned long int K[n][W+1];
    for(i=0;i<n;i++)
        scanf("%d",&a[i]);
    for(int i=0;i<n;i++)
        for(int j=0;j<=W;j++)
            {
            K[i][j]=0;
    }
    for(i=0;i<=W;i++)
        {
        if(i%a[0]==0)
            K[0][i]=1;
    }
```

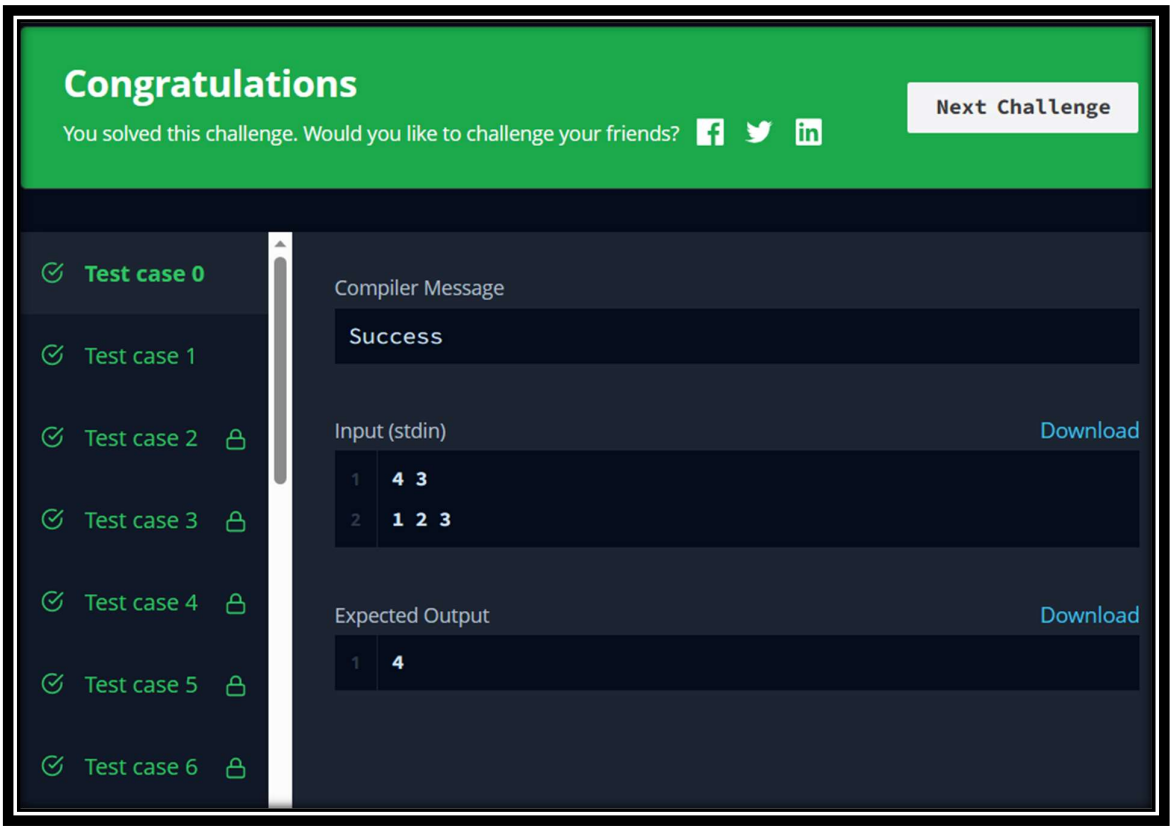
```c
    // Build table K[][] in bottom up manner
    for (i = 1; i<n; i++)
        for (w = 0; w <= W; w++)
        {
            if (w-a[i]>=0)
                K[i][w] = K[i][w-a[i]] + K[i-1][w];
            else
                K[i][w] = K[i-1][w];
        }
    return K[n-1][W];
}

int main() {

        int n,W;
        scanf("%d %d",&W,&n);
        printf("%lu",coinchange(W,n));

    /* Enter your code here. Read input from STDIN. Print output to STDOUT */
    return 0;
}
```

OUTPUT:


Congratulations
You solved this challenge. Would you like to challenge your friends?
Next Challenge
Test case 0
Test case 1
Test case 2
Test case 3
Test case 4
Test case 5
Test case 6
Compiler Message
Success
Input (stdin)
Download
1 4 3
2 1 2 3
Expected Output
Download
1 4