Challenge #9 [Facebook] - Pikachu's Alternate Map

You reached the front of the line as fast as you could. And now, you are about to buy the ticket to reach your hometown.

Oggy at the ticket counter gives you a schedule about the various ships. However, you also find Pikachu over there who gives a different map of secret routes that connect all the islands by road. And the distance between some islands is so small that you might as well walk it, instead of going by ship.



Now you are confused, how can you reach home really quickly?

You decide that you will take a mix of both routes. Some islands you will reach by walking, others by ship.

In Pikachu's map, the time taken between any island i and j is the absolute difference between i and j. So if there are are 5 islands, the walking time between them is:

* From 1 to 2, time is 1 mins, to 3 the time is 2 mins and so on...
* Similarly, the distance between 3 and 5 is 2 mins and so on...

Additionally, below is Oggy's map (via ships). It contains the time by ship between any 2  i and j

|  |  |
| --- | --- |
|  | 1 2 10 |
|  | 1 5 1 |

That is, there are 2 ships available. The ship between 1 and 2 takes 10 mins. But between 1 and 5 the time is 1 mins.

Thus, to reach 6, the best thing to do is to take a ship to 5 and then walk to 6

Taking your start point as 1 and end point as D, and given the schedule of N ships, you need to find the minimum time taken to reach home.

*Note: You can only go forward, not backward.*

**Input format**

1. The first line of input contains T, the number of test cases
2. The first line of each test case is D -  the destination
3. This is followed by N - the number of ship routes
4. After that, N lines of 3 numbers which signify start position, end position and time taken between the positions

**Output format**

1. For each test case, print the minimum time taken to reach the destination

**Example Input**

|  |  |
| --- | --- |
|  | 2 |
|  | 6 |
|  | 3 |
|  | 2 6 3 |
|  | 1 4 2 |
|  | 2 5 1 |
|  | 10 |
|  | 5 |
|  | 3 7 10 |
|  | 3 5 2 |
|  | 6 10 1 |
|  | 5 7 3 |
|  | 2 4 2 |

**Example Output**

|  |  |
| --- | --- |
|  | 3 |
|  | 6 |

**Explanation**

1. In the first case, the shortest path is to:  
   walk from 1 to 2: 1 minutes  
   ship from 2 to 5: 1 minutes  
   walk from 5 to 6: 1 minutes  
   Total: 3 minutes
2. In the second case, the best path is to:  
   walk from 1 to 3: 2 minutes  
   ship from 3 to 5: 2 minutes  
   walk from 5 to 6: 1 minutes  
   ship from 6 to 10: 1 minutes  
   Total: 6 minutes

Ans-

#include <bits/stdc++.h>

using namespace std;

#define N 3000

int main()

{

int T,D,n;

cin>>T;

for(int t = 0; t < T; t++)

{

//cout<<"\nEnter dest:";

cin>>D;

//cout<<"\nEnter n";

cin>>n;

int a[N][N];

for(int i = 0 ;i < n; i++)

{

for(int j = 0; j < 3; j++)

{

//cout<<i<<", "<<j<<endl;

cin>>a[i][j];

}

}

int p[D+1][D+1];

for(int i = 0 ; i < D+1; i++) //initialize with 0

{

for(int j = 0; j < D+1; j++)

{

p[i][j] = 0;

}

}

for(int i= 0; i < n; i++)

{

p[a[i][0]][a[i][1]] = a[i][2];

}

for(int i = 0; i < D+1; i++)//1st row

{

p[0][i] = 0;

}

for(int i = 1; i <= D-1; i++) //if Difference is one then just walk or 1 minute

p[i][i+1] = 1;

//processing

for(int diff = 2; diff <= D-1; diff++)

{

for(int i = D - diff; i >= 1; i--)

{

int j = i + diff;

int min = p[i][j];

if(min == 0)

min = INT\_MAX;

for(int k = i+1; k <= j - 1; k++)

{

int d = p[i][k] + p[k][j];

if(d < min)

min = d;

}

p[i][j] = min;

}

}

cout<<p[1][D]<<endl;

}

return 0;

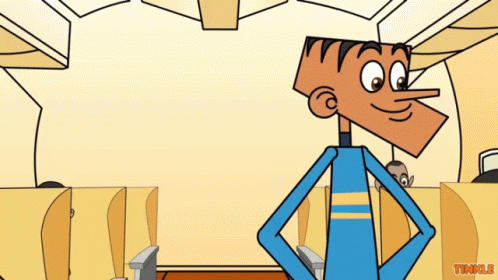
}

Challenge #10 [Google] - Suppandi does it again!

That was quite a travel you had in the past 9 days. You solved a bunch of problems for **Suppandi, Chhota Bheem and Perry the Platypus**. You rescued **Nobita**, flew with **Noddy**, played with **Shinchan** and crossed the bridge with **Dora**. You skipped a few spots in the line to by tickets from **Oggy** and followed **Pikachu**'s secret map.

And now you're back home. You would think that all your problems are over now. But there is one last adventure.

While you were away, **Suppandi**was busy trying to study. He doesn't understand words too well. In order to study, he thought he could make the sentences into one big word and learn the entire word in one go!



Now that you are back, you need to decipher your notebook. After all, you need to study to prepare for the upcoming ***Edyst April Daily Challenges***

This is what Suppandi has done:

1. He concatenated all the words into one big word, by removing all the spaces
2. He is not perfect at copying, so before concatenation he may have jumbled the original word. He jumbles as follows: he rotates a word up to 3 times to the left or up to 3 times to the right

Luckily, you have always made sure to follow the dictionary in whatever you write. You have a dictionary of all the words you have written. Given this dictionary, you need to figure out how many possible sentences could have been made from the very long word that Suppandi has made.

Example:

|  |  |
| --- | --- |
|  | Long Word: heretawsaemprobl |
|  | Dictionary: ["here", "there", "a", "was", "problem", "at", "saw"] |

Possible sentences:

|  |  |
| --- | --- |
|  | there saw a problem |
|  | there a saw problem |
|  | here at saw problem |

Thus, 3 sentences could have been our original sentence.

Given a long word, followed by a dictionary, print the number of possible sentences.

**Input Format**

* First line contains T, the number of test cases. For each test case, we have:
* First line has the long word
* The second line has N, followed by N words of the dictionary

**Output Format**

* Print the number of possible sentences that could lead to the long word

**Example Input**

|  |  |
| --- | --- |
|  | 2 |
|  | heretawsaemprobl |
|  | 7 here there a was problem at saw |
|  | ilooknowoolc |
|  | 6 i cool now know look loo |

**Output**

|  |  |
| --- | --- |
|  | 3 |
|  | 2 |

**Explanation**

1. Explained above
2. We could have had the following sentences:  
   i look now cool  
   i loo know cool

***Note: A word from the dictionary can be used more than once***

***Note: Students coding in Java, please keep your class name as Main***

Ans-

#include<bits/stdc++.h>

#define int int64\_t

using namespace std;

string rotateleft(string s)

{

if(s.size() == 1) return s;

else return s.substr(1 , s.length()-1) + s[0];

}

string rotateright(string s)

{

if(s.size() == 1) return s;

else return s[s.length()-1] + s.substr(0 , s.length()-1);

}

string func(string s , int idx , int len) {

string ans = "";

idx--;

while(len >= 1)

{

ans += s[idx];

idx--;

len--;

}

string res = "";

for(int i = ans.size()-1; i >= 0; i--) res += ans[i];

return res;

}

signed main() {

ios\_base::sync\_with\_stdio(0);

cin.tie(0);

int t , n;

cin >> t;

while(t--) {

string s;

cin >> s >> n;

vector<string> a(n);

for(int i = 0; i < n; i++) cin >> a[i];

map<string , int> map;

for(int i = 0; i < n; i++) {

int len = a[i].length();

set<string> set;

string left = a[i] , right = a[i];

set.insert(a[i]);

for(int j = 1; j <= 3; j++) {

if(j <= len) {

left = rotateleft(left);

right = rotateright(right);

set.insert(left) , set.insert(right);

}

}

for(auto str : set) map[str]++;

}

vector<int> dp(s.length()+1 , 0);

dp[0] = 1;

for(int i = 1; i <= s.length(); i++) {

for(int len = 1; len <= i; len++) {

string str = func(s , i , len);

dp[i] = dp[i] + map[str] \* dp[i-len];

}

}

cout << dp[s.length()] << "\n";

}

return 0;

}