Submitted to

**Md. Rajib Hossain**

Lecturer, Dept of CSE

Imperial College of Engineering

Submitted by

**Fahim Ahammed Firoz**

ID: 1938520126

Part-2, Odd Semester

Dept. of CSE

Imperial College of Engineering

Lab Report

Imperial College of Engineering

(Rajshahi University Affiliated)

Jessore Road, Boikali, Khulna-9000



|  |  |
| --- | --- |
| **Course Code** | **CSE2222** |
| **Course Title** | **Design and Analysis of Algorithms Lab** |

**Problem 1: Mice and maze**

#include<bits/stdc++.h>

using namespace std;

#define MX 105

#define INF 1000000000

struct node{

int val;

int cost;

};

vector < node > G[MX];

bool vis[MX];

int dist[MX];

void reset(){

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

G[i].clear();

vis[i] = 0;

dist[i] = INF;

}

}

class cmp{

public:

bool operator() (node &A, node &B){

if (A.cost > B.cost) return true;

return false;

}

};

int dijkstra(int source, int Time){

priority\_queue<node, vector<node>, cmp> PQ;

PQ.push({source, 0});

int cnt = 0;

while(!PQ.empty()){

node current = PQ.top();

PQ.pop();

int val = current.val;

int cost = current.cost;

if (vis[val] == 1) continue;

dist[val] = cost;

vis[val] = 1;

if (dist[val] <= Time){

cnt++;

}

for (int i = 0; i < G[val].size(); i++){

int nxt = G[val][i].val;

int nxtCost = G[val][i].cost;

if (vis[nxt] == 0){

PQ.push({nxt, cost + nxtCost});

}

}

}

return cnt;

}

int main()

{

//freopen("input.txt", "r", stdin);

int test;

scanf("%d", &test);

for (int cs = 1; cs <= test; cs++){

reset();

int n, e, t;

scanf("%d%d%d", &n, &e, &t);

int m;

scanf("%d", &m);

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

int u, v, w;

scanf("%d%d%d", &u, &v, &w);

G[v].push\_back({u, w});

}

int ans = dijkstra(e, t);

if (cs > 1) printf("\n");

printf("%d\n", ans);

}

return 0;

}

**Problem-2: Cheeky Cheeky:**

#include <iostream>

#include <string>

using namespace std;

int main(){

int t;

cin>>t;

while(t--){

ws(cin);

string s;

cin>>s;

int n=s.size();

cout<<"size: "<<n<<endl;

int len=n/3+1;

cout<<"len: "<<len<<endl;

if(len<1){

cout<<"len < 1 : "<<len<<endl;

len=1;

}

string ans;

while(len<=n/2){

cout<<"len <= N/2 : "<<len<<endl;

string tmp=s.substr(n-len,len);

if(tmp==s.substr(n-2\*len,len))

ans=tmp;

cout<<"ans tmp : "<<ans<<endl;

len++;

}

for(int i=0;i<8;i++)

cout<<ans[i%ans.size()];

cout<<"..."<<endl;

}

}

Problem-3: Palindrome:

#include <iostream>

using namespace std;

void solve();

int main(){

int t;

cin >> t;

while (t--){

solve();

}

return 0;

}

void solve(){

int a;

cin >> a;

int arr[a];

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

cin >> arr[i];

bool flag = false;

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

for (int j = i + 2; j < a; j++){

if (arr[i] == arr[j]){

flag = true;

}

}

}

if (a==1 || flag)cout << "YES" << endl;

else cout << "NO" << endl;

}

**Problem-4:**

#include<iostream>

using namespace std;

typedef long long ll;

vector<ll>pi;

void preFunc(string s){

ll len=s.size();

pi.resize(len+5);

pi[0]=0;

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

ll j=pi[i-1];

while(j>0&&s[i]!=s[j])j=pi[j-1];

if(s[i]==s[j])j++;

pi[i]=j;

}

}

int main(){

ll t;

cin>>t;

while(t--){

string s;

cin>>s;

ll len=s.size();

reverse(s.begin(),s.end());

preFunc(s);

ll idx;

for(ll i=len-1;i>=0;i--){

if(pi[i]\*2==(i+1)){

idx=pi[i]-1;

break;

}

}

string ans="";

for(ll i=0;i<=idx;i++)ans+=s[i];

len=ans.size();

ll rep=0;

if(len<8){

rep=(8/len);

if(8%len!=0)rep++;

}

for(ll i=0;i<rep;i++){

ans+=ans;

}

reverse(ans.begin(),ans.end());

for(ll i=0;i<8;i++)cout<<ans[i];

cout<<"...\n";

}

return 0;

}