

End-Term Report

Coding Fiesta H9

Vipul Singh

Roll No-200020164

Mentor -Shivang Tiwari

Problem A

The packs needed for dogs from universal food will be $\max(0, x-a)$, similarly for cats it will be $\max(0, y-b)$, the sum of both should be at max c.

Code....

```
// हर हर महादेव
using namespace std;
#include <bits/stdc++.h>
typedef long long ll;

int main()
{
    int t;
    cin>>t;
    while(t--){
        int a,b,c,x,y;
        cin>>a>>b>>c>>x>>y;
        if (max(x-a,0)+max(y-b,0) <=c) cout<<"YES"<<endl;
        else cout<<"NO"<<endl;
    }
}
```

Problem B

We need to iterate from the end of array and check if $a_i \geq a_{i+1}$
Divide a_i by 2 till it becomes less than a_{i+1} and count the number

of operations and perform the same for all indexes and the total operations to get the final answer

Code.....

```
// हर हर महादेव
using namespace std;
#include <bits/stdc++.h>
typedef long long ll;
int round(int a[],int i,int j){
int num=0;
while(a[i]>=a[j]){
a[i]/=2;
num++;
}
return num;
}
int main()
{
int t;
cin>>t;
while(t--){
int n;
cin>>n;
int a[n];
for(int i=0;i<n;i++)cin>>a[i];
int ans=0;
for(int i=n-2;i>=0;i--){
if(a[i+1]>0)ans+=round(a,i,i+1);
else {ans=-1;break;}
}
cout<<ans<<endl;
}
}
```

Problem C

Find the last index(one) at which '1' is present and first index (zero) where '0' is present

Ans will be-

If no 1's are present ans will be zero+1

If no 0's are present ans will be n-one

If both 0's and 1's are absent ans will be n

If both 0's and 1's are present ans will be zero-one +1

Code....

```
// हर हर महादेव
using namespace std;
#include <bits/stdc++.h>
typedef long long ll;

int main()
{
    int t;
    cin>>t;
    while(t--){
        string s;
        cin>>s;
        int n=s.size();
        int one=-1,zero=-1;
        for(int i=0;i<n;i++){
            if(s[i]=='1'){one=i;}
        }

        for(int i=0;i<n;i++){
            if(s[i]=='0'){zero=i;break;}
        }
        int ans;
        if(one==-1 && zero==-1)ans=n;
        else if(zero==-1){ans=n-one;}
        else if(one==-1)ans=zero+1;
        else ans=zero-one+1;
        cout<<ans<<endl;
    }
}
```

Problem D

The minimum number of non-intersecting leading down paths that can cover all vertices of the tree is equal to the total leaf nodes. Using depth first search we can print the total number of nodes in each path and each node for each path.

Code

```
// हर हर महादेव
using namespace std;
#include <bits/stdc++.h>
typedef long long ll;
void dfs(vector<vector<int>>&adj, vector<int> &visited, int
node, queue<int>&st) {
    st.push(node);
    visited[node]=1;
    if(adj[node].size()==1) {
        cout<<st.size()<<endl;
        while(!st.empty()) {cout<<st.front()<<" ";st.pop();}
        cout<<endl;
    }
    for(auto x: adj[node]) {
        if(visited[x]==0) {
            dfs(adj, visited, x, st);
        }
    }
}
int main()
{
    int t;
    cin>>t;
    while(t--){
        int n;
        cin>>n;
        int p[n];
        for(int i=0;i<n;i++)cin>>p[i];
        set<int>xp;
        for(int i=0;i<n;i++){xp.insert(p[i]);}
        int num=xp.size();
        vector<vector<int>>adj(n+1);
        for(int i=0;i<n;i++){
            adj[i+1].push_back(p[i]);
            adj[p[i]].push_back(i+1);
        }
        vector<int>visited(n+1,0);
        int root;
        for(int i=0;i<n;i++){
            if(p[i]==i+1){root=i+1;break;}
        }
        if(n>1){cout<<n-num<<endl;
        queue<int>st;
        dfs(adj, visited, root, st);}
        else {cout<<1<<endl<<1<<endl<<1<<endl;}
    }
}
```

Link for the github repo...

<https://github.com/vipulconda/Coding-Fiesta->