Competitive Programming

Submitted by:

Niwedita

17CO227

Acknowledgement

I would like to express my deepest appreciation to all IET club members, who provided us the opportunity of pursuing Summer Online course. A special gratitude I give to our mentors who helped and encouraged us for solving competitive programming problems.

Furthermore, I would also like to acknowledge with much appreciation the crucial role of IET club convenor, who gave us the permission to access support from our mentors. A special thanks goes to my batchmates, who were there with me during sessions as most of my doubts got cleared in their conversation with mentors.

Last but not least, many thanks go to my parents for cooperating during sessions.

Abstract

Competitive programming is a very vast topic. In this summer course, we got basic knowledge of all the topics of competitive programming.

It is a mind sport held over the Internet or a local network, involving participants trying to program according to provided specifications.

To get in-depth knowledge of competitive programming, participant should practice as many problems as he can.

The aim of Competitive Programming is to write a source code of computer programs which are able to solve given problems. A vast majority of problems appearing in programming contests are mathematical or logical in nature.

Content

1. Introduction

Competitive Programming is a mind sport held over the Internet or a local network, involving participants trying to program according to provided specifications. Contestants are referred to as sport programmers.

A programming competition generally involves the host presenting a set of logical or mathematical problems to the contestants and contestants are required to write computer programs capable of solving each problem. Judging is based mostly upon number of problems solved and time spent for writing successful solutions, but may also include other factors (quality of output produced, execution time, program size, etc.).

The aim of Competitive Programming is to write a source code of computer programs which are able to solve given problems. A vast majority of problems appearing in programming contests are mathematical or logical in nature.

Irrespective of the problem category, the process of solving a problem can be divided into two broad steps: constructing an efficient algorithm and implementing the algorithm in a suitable programming language.

In most contests, the judging is done automatically by host machines, commonly known as judges. Every solution submitted by a contestant is run on the judge against a set of test cases. Normally, contest problems have an all-or-none marking system, means that a solution is "Accepted" only if it produces satisfactory results on all test cases run by the judges, and rejected otherwise. However, some contest problems may allow for partial scoring, depending on the number of test cases passed, the quality of the results, or some other specified criteria.

2. Overview

Overall, all the sessions were so interactive and useful for us. We learnt a lot of new stuffs. Initially, I was totally new with this topic. But I managed to be with everyone during sessions and solving problems. Many of the Data structure topics are covered in the sessions, it's definitely going to help us in our college course.

It was really a very nice experience to be with you people.

The only problem I faced during this summer course is the timing. It's difficult to be active during whole session as that include dinner time. But still, that we can manage.

Once again, I appreciate your efforts. It was beyond my expectation. We were totally engaged in this whole summer vacation.

3. Assignment

3.1 Assignment 1

3.1.1

Link: //https://www.codechef.com/COOK85/problems/GAMSTICK/

Code:

```
#include <iostream>

using namespace std;
int main()
{
int t;
cin>>t;
while(t--)
{
    int x1,x2,x=0;
    long long y1,y2,n,y3;
    cin>>n>>x1>>y1>>x2>>y2;
    if(y1==y2)
    {
            cout<<"Draw"<<endl;
            continue;
    }
    if(y1>y2)
    {
            y1=n+1-y1;
```

```
        y2=n+1-y2;
}
y3=y2-y1;
if(x1==x2)
{
        if(y3%2==0)
        y1++;
        y3--;
        y3/=2;
        y1+=y3;
        y2-=y3;
        if(n%2==0)
        {
                if(y1<(n/2))
                cout<<"Slava"<<endl;
                else if(y1==(n/2))
                cout<<"Draw"<<endl;
                else
                cout<<"Miron";
        }
        else
        {
                if(y1<=(n/2))
                cout<<"Slava"<<endl;
                else
                cout<<"Miron"<<endl;
        }
}
```

```
else
{
        if(y3==1)
        {
                if(y1<=(n/2))
                cout<<"Draw"<<endl;
                else
                cout<<"Miron"<<endl;
                continue;
        }
        if(y3==0)
        {
                cout<<"Draw"<<endl;
                continue;
        }
        if(y3%2==0)
        {
        x=1;
        y1++;
        }
        y3--;
        y3/=2;
        y1+=y3;
        y2-=y3;
        if(n%2==0)
        {
                if(y1<(n/2))
                {
```

```
                if(x==1)
                cout<<"Slava"<<endl;
                else
                {
                        if((1+y1)==(n/2))
                        cout<<"Draw"<<endl;
                        else
                        cout<<"Slava"<<endl;
                }
        }
        else if(y1==(n/2))
        cout<<"Draw"<<endl;
        else
        {
                if(x==0)
                cout<<"Miron"<<endl;
                else
                {
                        if((y1-1)==(n/2))
                        cout<<"Draw"<<endl;
                        else
                        cout<<"Miron"<<endl;
                }
        }
}
else
{
        if(y1<=(n/2))
```

```
                {
                        if(x==1)
                        cout<<"Slava"<<endl;
                        else
                        {
                                if(y1==(n/2))
                                cout<<"Draw"<<endl;
                                else
                                cout<<"Slava"<<endl;
                        }
                }
                else
                {
                        if(x==0)
                        cout<<"Miron"<<endl;
                        else
                        {
                                if((y1-1)==(n/2))
                                cout<<"Draw"<<endl;
                                else
                                cout<<"Miron"<<endl;
                        }
                }
        }
    }
}
}
```

3.1.2

Link: //https://www.codechef.com/LTIME52/problems/C00K0FF/

Solution: https://www.codechef.com/status/C00K0FF,nivi2017

code:

```
#include <iostream>
#include <string>
using namespace std;

int main()
{
    // your code goes here
    long t;
    cin>>t;
    while(t--)
    {
        long n;
        int c0 = 0;
        int c1 = 0;
                int c2= 0;
        int c3= 0;
        int c4= 0;
        int c5= 0;
```

```
            int c6= 0;
        cin>>n;
      for(long i=0;i<n;i++)
          {
        string s;
        cin>>s;
        if(s == "cakewalk")
            c0++;
        else if(s == "easy")
            c1++;
        else if(s== "simple")
             c2++;
        else if(s == "easy-medium")
            c3++;
        else if(s == "medium")
            c4++;
        else if(s == "medium-hard")
            c5++;
        else if(s == "hard")
            c6++;
          }
        if((c0 > 0) && (c1 > 0) && (c2 > 0) && (c3> 0 || c4> 0) && (c5 > 0
||c6 > 0))
        cout<<"Yes"<<endl;
        else
         cout<<"No"<<endl;
    }
    return 0;
```

}

3.2 :Assignment2

3.2.1

Link: //https://www.hackerrank.com/contests/university-codesprint-3/challenges/bobs-game/

Code:

```
#include <iostream>
#include <set>
#include <vector>
using namespace std;
int main()
{
int t,i,j,n;
int g[1024][1024];
char s[1024][1024];
cin>>t;
while(t--)
{
    cin>>n;
    for(i=0;i<n;i++)
    cin>>s[i];
    for(i=0;i<n;i++)
    {
```

```
            for(j=0;j<n;j++)
            if (s[i][j]=='X')
            {
                  g[i][j]=-1;
                  continue;
            }
            set<int>w;
            if(i>0)
            w.insert(g[i-1][j]);
            if(j>0)
            w.insert(g[i][j-1]);
            if(i>0 && j>0)
            w.insert(g[i-1][j-1]);
            g[i][j]=0;
            while (w.find(g[i][j])!=w.end())
            g[i][j]++;
      }
}
int a=0;
for(i=0;i<n;i++)
      for(j=0;j<n;j++)
      if(s[i][j]=='K')
      a^=g[i][j];
if(a==0)
cout<<"lose"<<endl;
else
{
      int c=0;
```

```
    for(i=0;i<n;i++)
            for(j=0;j<n;j++)
            {
                    if(s[i][j]!='K')
                    continue;
                    vector<int> v;
                    if(i>0 && s[i-1][j]!='X')
                    v.push_back(g[i-1][j]);
                    if(j>0 && s[i][j-1]!='X')
                    v.push_back(g[i][j-1]);
                    if(i>0 && j>0 && s[i-1][j-1]!='X')
                    v.push_back(g[i-1][j-1]);
                    vector<int>::iterator itr=v.end();
                    if((a ^ *itr ^ g[i][j])==0)
                    c++;
            }
            cout<<"win"<<endl;
}
}
```

3.3 Assignment3

3.3.1

Link: //http://codeforces.com/problemset/problem/687/B/

Code: #include <iostream>

using namespace std;

```
int GCD(int a , int b)
{
int temp;
if (a<b)
{
temp=a;
a=b;
b=temp;
}
if (b==0)
return a;
return GCD(b,a%b);
}
int LCM(int a,int b)
{
return a*b/GCD(a,b);
}
int main()
{
int n,k;
cin >> n >>k;
int t=1;
for(int i=0;i<n;i++)
{
int c;
cin >> c;
t=GCD(k,LCM(c,t));
}
```

```
if(t==k)
cout << "yes"<< endl;
else
cout <<"no"<< endl;
return 0;
}
```

3.3.2

Link: //http://codeforces.com/problemset/problem/75/C

Solution: http://codeforces.com/contest/75/submission/38471378

Code:
```
#include <iostream>
#include <algorithm>
#include <vector>
#include <cmath>
using namespace std;
vector<int>d;
int GCD(int m,int n)
{
    int temp;
    if(m<n)
    {
            temp=m;
            m=n;
            n=temp;
    }
```

```
    if(n==0)
    return m;
    return GCD(n,m%n);
}
void all_d(int g)
{
    int rt= sqrt(g);
    for(int i=1;i<=rt;i++)
    {
        if(g%i==0)
        {
            d.push_back(i);
            d.push_back(g/i);
        }
    }
    if(d.back()==rt)
    d.pop_back();
    sort(d.begin(),d.end());
}
int main()
{
    int a,b,n,low,high;
    cin>>a>>b;
    all_d(GCD(a,b));
    cin>>n;
    for(int i=0;i<n;i++)
    {
        cin>>low>>high;
```

```
            int c;
            c = upper_bound(d.begin(),d.end(),high)-d.begin();
            cout<<((c>0 && low<=d[c-1])? d[c-1] : -1)<<endl;
    }
    return 0;
}
```

3.4 Assignment5

3.4.1

Link: //https://www.codechef.com/OCT17/problems/MARRAYS

Solution: https://www.codechef.com/viewsolution/18990575

Code:

```
#include<iostream>
#include<vector>
#include<algorithm>
using namespace std;
int main()
{
    long long t,n,i,j,k,m,p;
    cin>>t;
    while(t--)
    {
            cin>>n;
```

```
vector<long long>v[n],sum[n],last;
for(i=0;i<n-1;i++)
{
	cin>>m;
	for(j=0;j<m;j++)
	{
		cin>>p;
		v[i].push_back(p);
	}
}
cin>>m;
for (j=0;j<m;j++)
	{
		cin>>p;
		last.push_back(p);
	}
sort(last.begin(),last.end());
v[n-1].push_back(last[0]);
v[n-1].push_back(last[m-1]);
for (i=0;i<v[0].size();i++)
	{
		sum[0].push_back(0);
	}

for (i=1;i<n;i++)
{
	for (j=0;j<v[i].size();j++)
	{
```

```
                long long max =0;
                for (k = 0; k < v[i-1].size()-1; k++)
                {
                        p=abs(v[i][j]-v[i-1][k])*i + sum[i-1][k+1];
                        if(max<p)
                        max=p;
                }
                p=abs(v[i][j]-v[i-1][k])*i + sum[i-1][0];
                if(max<p)
                        max=p;
                sum[i].push_back(max);

        }
    }
    long long max =0;
    for ( i = 0; i < sum[n-1].size(); ++i)
        {
            if(sum[n-1][i]>max)
                max=sum[n-1][i];
        }
    cout<<max<<endl;

  }
  return 0;
}
```

4. Contest

4.1 Contest1: https://www.hackerrank.com/cp-summer-mentorship-test-1

4.1.1: Pop count

Code:

```
#include<iostream>
using namespace std;
int main()
{
    int n,q;
    long long a[200000];
    cin>>n;
    for(int i=1;i<=n;i++)
    {
        cin>>a[i];
    }
    cin>>q;
    while(q--)
    {
        int l,r;
        long long min,max,c;
        cin>>l>>r;
        if(r>l)
        {
            min=a[l];
            max=a[l];
            for(int i=l;i<=r;i++)
```

```
        {
                if(a[i]>=max)
                        max=a[i];
                else if(a[i]<=min)
                        min=a[i];
        }
}
else
{
        min=a[l];
        max=a[l];
        for(int i=l;i<=n;i++)
        {
                if(a[i]>=max)
                        max=a[i];
                else if(a[i]<=min)
                        min=a[i];
        }
        for(int i=1;i<=r;i++)
        {
                if(a[i]>=max)
                        max=a[i];
                else if(a[i]<=min)
                        min=a[i];
        }
}
c= max-min;
cout<<c<<endl;
```

```
    }
    return 0;
}
```

4.1.2 : Lets play odd even

Code:

```
#include <iostream>
#include <algorithm>
#include <cmath>
using namespace std;
int multiset(int a, int p, int q, int r)
{
    int add=0,c;
    for(int i=1;i<=sqrt(a);i++)
    {
    if(a%i==0)
    {
            if(((a/i)%2==0 && (i%2!=0)) || ((a/i)%2!=0 && (i%2==0)))
             c= (r>add)?r:add;
            else if((a/i)%2!=0 && (i%2!=0))
             c= (p>add)?p:add;
            else if((a/i)%2==0 && (i%2==0))
            c=(q>add)?q:add;
            add=c;
    }
    }
    return add;
}
```

```
int func(int N,int p,int q, int r)
{
    int a[100];
    int sum=0;
    for(int i=0;i<N;i++)
    {
            cin>>a[i];
            sum = sum + multiset(a[i],p,q,r);
    }
    return sum;
}

int main()
{
    int N,t,p,q,r;
    cin>>t;
    for(int i=0;i<t;i++)
    {
            cin>>N>>p>>q>>r;
            cout<<func(N,p,q,r)<<endl;
    }
    return 0;
}
```

4.1.3 : Summation Series

Code:

```
#include<bits/stdc++.h>
#define ll long long
```

```
#define m 1000000007
using namespace std;
ll power(ll b,ll p)
{
    if(p==0)
        return 1;
    if(p==1)
        return b;
    else
    {
        ll temp=power(b,p/2);
        if(p%2==0)
            return (temp%m * temp%m)%m;
        else
            return ((temp%m * temp%m)%m*b%m)%m;
    }
}
int main()
{
    int n;
    cin>>n;
    ll a[1000000];
    ll prod=1;
    ll g=0;
    for(int i=0;i<n;++i)
    {
        cin>>a[i];
        prod=(prod%m*a[i]%m)%m;
```

```
        g=__gcd(g,a[i]);
    }
    ll ans=0;
    for(int i=0;i<n;++i)
    {
        ans = (ans%m+ ((prod%m * power(a[i],m-2)%m)%m))%m;
    }
    cout<<ans%m<<endl;

}
```

4.1.4 : Van Helsing Snares Dracula

Code:

```
#include<iostream>
using namespace std;
int main()
{
    long long n,v=0;
    cin>>n;
    while(n)
    {
            if(n%4>1)
            {
                    v=-1;
                    break;
            }
            else
```

```
        {
            v+=n%4;
            n/=4;
        }
    }
    cout<<v<<endl;
    return 0;
}
```

4.1.5 : Shipment of toys

Code:

```
#include<iostream>
using namespace std;
int main()
{
    int n,w[100000],temp,count=0,k=0;
    cin>>n;
    for(int i=0;i<n;i++)
        cin>>w[i];
    for(int i=0;i<n-1;i++)
    {
        for(int j=0;j<(n-i-1);j++)
        {
            if(w[j+1]<w[j])
            {
                temp=w[j];
                w[j]=w[j+1];
                w[j+1]=temp;
```

```
                }
            }
        }
        int c=0;
        while(c<n)
        {
            if(w[k]>w[c]+4)
            {
                c=k;
                count++;
            }
            k++;
        }
        cout<<count<<endl;
    }
```

4.1.6: Good Knight

Code:

```
#include<bits/stdc++.h>
#define ll long long
#define m 1000000007
using namespace std;
ll power(ll b,ll p)
{
   if(p==0)
      return 1;
   if(p==1)
      return b;
   else
   {
      ll temp=power(b,p/2);
      if(p%2==0)
         return (temp%m * temp%m)%m;
```

```
        else
            return ((temp%m * temp%m)%m*b%m)%m;
    }
}
int main()
{
    int n;
    cin>>n;
    ll a[1000000];
    ll prod=1;
    ll g=0;
    for(int i=0;i<n;++i)
    {
        cin>>a[i];
        prod=(prod%m*a[i]%m)%m;
        g=__gcd(g,a[i]);
    }
    ll ans=0;
    for(int i=0;i<n;++i)
    {
        ans = (ans%m+ ((prod%m * power(a[i],m-2)%m)%m))%m;
    }
    cout<<ans%m<<endl;

}
```

4.1.7: Fill the Tank

Code:

```
#include<iostream>
#include<math.h>
using namespace std;
int main()
{
    int q;
    cin>>q;
    while(q--)
```

```
    {
        long long n;
        cin>>n;
        double s=sqrt(n);
        if(ceil(s)==floor(s))
            cout<<"B"<<endl;
        else
            cout<<"A"<<endl;
    }
    return 0;
}
```

4.1.8 : Calculate the power

Code:

```
#include <math.h>
#include <vector>
#include <iostream>
#include <algorithm>
#define ll long long
#define m 1000000007
using namespace std;

ll power(ll b,ll p)
{
   if(p==0)
      return 1;
   if(p==1)
      return b;
   else
```

```
    {
        ll temp=power(b,p/2);
        if(p%2==0)
            return (temp%m * temp%m)%m;
        else
            return ((temp%m * temp%m)%m*b%m)%m;
    }
}
int main() {
    int t;
    cin>>t;
    for(int w=1;w<=t;++w)
    {
        ll n,k,x,y,c,d,e1,e2,f;
        cin>>n>>k>>x>>y>>c>>d>>e1>>e2>>f;
        ll x1,y1;
        ll a[1000000];
        a[1]=(x+y)%f;
        for(int i=2;i<=n;++i)
        {
            x1=(c*x +d*y +e1)%f;
            y1=(d*x + c*y +e2)%f;

            a[i]=(x1+y1)%f;
            x=x1;
            y=y1;
        }
```

```
        ll ans=0,sum=k;
        ans=((a[1]%m*(n)%m)%m*sum%m)%m;
       // cout<<ans<<endl;
       // cout<<sum<<endl;
        for(int i=2;i<=n;++i)
        {
            sum=(sum%m +(i%m*((ll)power(i,k)%m-1%m+m)%m*power((i%m-
1%m+m)%m,m-2))%m)%m;
           // cout<<sum<<endl;
            ans=(ans%m + ((a[i]%m*(n-i+1)%m)%m*sum%m)%m)%m;
        }
        cout<<"Case #"<<w<<": "<<ans%m<<endl;
    }
    return 0;
}
```

4.1.9 Golf with Gray Code

Code:

```
#include <iostream>
#include <cmath>
#include <algorithm>
using namespace std;
int graycode(int a)
{
return a^(a>>1);
}
```

```
int main()
{
int n;
float a;
cin>>n;
for(int i=0;i<n;i++)
{
cin>>a;
cout<<graycode(a)<<endl;
}
return 0;
}
```

4.1.10: Game of numbers

Code:

```
#include<iostream>
#include<math.h>
using namespace std;
int main()
{
    int q;
    cin>>q;
    while(q--)
    {
            long long n;
            cin>>n;
            double s=sqrt(n);
```

```
            if(ceil(s)==floor(s))
                    cout<<"B"<<endl;
            else
                    cout<<"A"<<endl;
    }
    return 0;
}
```

4.2 Final contest: https://www.hackerrank.com/cp-summer-mentorship-final-contest

4.2.1: Lesser Primes

Code:

```
#include<bits/stdc++.h>
using namespace std;
int Nprime(int n )
{
   int count=0;
  bool prime[n+1];
  memset(prime,true,sizeof(prime));
  for(int p=2;p*p<=n;p++)
  {
   if(prime[p]==true)
   {
      for(int i=p*2;i<=n;i+=p)
         prime[i]=false;
   }
  }
```

```
  for(int p=2;p<=n;p++)
  if(prime[p] && p!=n)
  count++;
    return count;
}
int main()
{
    int n;
    cin>>n;
    cout<<Nprime(n)<<endl;
    return 0;
}
```

4.2.2: Bunny loves u

Code:

```
#include<bits/stdc++.h>
using namespace std;
int subString(char str[],int n)
{
     int count=0;
     for(int len=1;len<=n;len++)
     {
              for(int i=0;i<=n-len;i++)
              {
                       int j=i+len-1;
                       for(int k=i;k<=j;k++)
                       {
                                if(str[k]=='u')
                                {
```

```
                    count++;
                    break;
                }
            }
        }
    }
    return count;
}
int main()
{
    char str[1000];
    cin>>str;
    cout<<subString(str,strlen(str))<<endl;
    return 0;
}
```

4.2.3: Play to win

Code:

4.2.4

4.2.5: Choice of seats

Code:

```
#include<iostream>
#include<string.h>
using namespace std;
int main()
{
    char s[100000];
    int ans;
    cin>>s;
```

```
    if(s[strlen(s)-1]=='E')
            cout<<strlen(s)-1<<endl;
    else if(s[0]=='E')
            cout<<"0"<<endl;
    else
    {
            for(int i=strlen(s)-2;i>0;i--)
            {
                    if(s[i]=='E' && s[i-1]=='E')
                            ans=i;
            }
            cout<<ans<<endl;
    }
    return 0;
}
```

4.2.5: Cyclic array problem

Code:

```
#include<iostream>
using namespace std;
int main()
{
    int n,q;
    long long a[200000];
    cin>>n;
    for(int i=1;i<=n;i++)
    {
            cin>>a[i];
```

```
}
cin>>q;
while(q--)
{
	int l,r;
	long long min,max,c;
	cin>>l>>r;
	if(r>l)
	{
		min=a[l];
		max=a[l];
		for(int i=l;i<=r;i++)
		{
			if(a[i]>=max)
				max=a[i];
			else if(a[i]<=min)
				min=a[i];
		}
	}
	else
	{
		min=a[l];
		max=a[l];
		for(int i=l;i<=n;i++)
		{
			if(a[i]>=max)
				max=a[i];
			else if(a[i]<=min)
```

```
                min=a[i];
            }
            for(int i=1;i<=r;i++)
            {
                if(a[i]>=max)
                    max=a[i];
                else if(a[i]<=min)
                    min=a[i];
            }
        }
        c= max-min;
        cout<<c<<endl;
    }
    return 0;
}
```

4.2.6: Good subsequence

Code:

```
#include<iostream>
using namespace std;
int main()
{
    int n,q;
    long long a[200000];
    cin>>n;
    for(int i=1;i<=n;i++)
    {
        cin>>a[i];
```

```
}
cin>>q;
while(q--)
{
	int l,r;
	long long min,max,c;
	cin>>l>>r;
	if(r>l)
	{
		min=a[l];
		max=a[l];
		for(int i=l;i<=r;i++)
		{
			if(a[i]>=max)
				max=a[i];
			else if(a[i]<=min)
				min=a[i];
		}
	}
	else
	{
		min=a[l];
		max=a[l];
		for(int i=l;i<=n;i++)
		{
			if(a[i]>=max)
				max=a[i];
			else if(a[i]<=min)
```

```
                min=a[i];
            }
            for(int i=1;i<=r;i++)
            {
                if(a[i]>=max)
                    max=a[i];
                else if(a[i]<=min)
                    min=a[i];
            }
        }
        c= max-min;
        cout<<c<<endl;
    }
    return 0;
}
```

5. Future work

After getting hold of basics of all the competitive programming topics, I would like to in-depth my knowledge in competitive programming.

Computer programming is a mind sport which increases our ability of solving any situation. I will try to increase my Problem-solving ability. I wish to develop more impressive algorithms to solve all the problems in much easier way.

I will try to develop a programming competition community with more innovative code challenges related to our daily life.