



REPUBLIKA E SHQIPËRISË
MINISTRIA E ARSIMIT
DHE SPORTIT
QENDRA E SHËRBIMEVE ARSIMORE

OLIMPIADA KOMBËTARE E INFORMATIKËS

Faza e tretë

Viti mësimor 2019 - 2020

17 Tetor 2020

- Olimpiada filloi në orën **10.00** dhe mbaroi në orën **13.00**.
- Teza përmbante 4 ushtrime.
- Për të zgjidhur secilin ushtrim nxënësi mund të përdoren gjuhët e programimit: C, C++ apo Java.
- Zgjidhjet u bënë në kompjuter.
- Çdo zgjidhje e dhënë nga nxënësit që ishte e saktë shkencërisht, u konsiderua dhe u vlerësua nga komisioni.

1. Një mënyrë zgjidhje për ushtrimin e parë është:
(10 pikë)

```
// Klasa loja
class loja
{
    public:
        int lojaKod;
        int nivelet;
        int piket;
};
// funksioni qe krahason lojrat
bool krahaso(loja *x,loja *y)//funksion boolean
{
    return x->piket>y->piket;
}

int maxpiket(loja** obj,int n){
    int max=0,piket=0;;
    //maksimum nivelet
    for(int i=0;i<n;i++){
        if(i==0){
            max=obj[i]->nivelet;
        }
        else{
            if(obj[i]->nivelet>max)
                max=obj[i]->nivelet;
        }
    }
}

sort(obj,obj+n,krahaso);

int T1[max]={0}; // T1 tab me vlera 0
bool T2[max]={false}; //T2 false
for(int i=0;i<n;i++)
```

```

    {

        for(int j=(obj[i]->nivelet)-1;j>=0;j--)
        {
            if(T2[j]==false)
            {

                piket+=obj[i]->piket;
                T1[j]=obj[i]->lojaKod;
                T2[j]=true;
                break;

            }

        }

        // afishimi i sekuenes se lojrave
        cout<<"Lojrat:"<<"\t";
        for(int i=0;i<max;i++)
        {
            if(T2[i])
                cout<<T1[i]<<" ";

        }
        return piket; //piket
    }

int main()
{
    int n,size,max,totalpiket=0;
    cout<<"Jepni numrin maksimal te lojrave ne kompjuter:";
    cin>>n;
    loja *obj[n];

    for(int i=0;i<n;i++)
    {
        obj[i]=(loja*)malloc(sizeof(struct loja));
        cout<<"Jepni kodin e lojes, nivelet dhe piket: "<<i+1<<endl;

        cin>>obj[i]->lojaKod;
        cin>>obj[i]->nivelet;
        cin>>obj[i]->piket;

    }

    totalpiket=maxpiket(obj,n); //total piket
    cout<<"\nPiket totale te fituara: "<<totalpiket<<"\n";

    return 0;
}

```

**2. Një mënyrë zgjidhje për ushtrimin e dytë është:
(10 pikë)**

```
#include <algorithm>
#include <bitset>
#include <cassert>
#include <cctype>
#include <cmath>
#include <cstdio>
#include <cstdlib>
#include <cstring>
#include <ctime>
#include <deque>
#include <functional>
#include <iomanip>
#include <iostream>
#include <list>
#include <map>
#include <numeric>
#include <queue>
#include <set>
#include <sstream>
#include <stack>
#include <string>
#include <utility>
#include <vector>
using namespace std;
const int MOD=151109;
char b[100][100];
int m,n;
const int MAX = 76867;
int rev[1<<20], tr[1<<20];
int dp[20][20][MAX];
int ALL;
bool isok(int x) {
int nv=0,prev=0;
while (x>0) {
if (x&1) nv += prev, prev = 1;
else prev = 0;

x /= 2;

}

return nv<=1;
}
const int MAXTOUCH = 8858835;

int touched[MAXTOUCH];

int touchcnt;

#define MP make_pair
```

```
int go(int x, int y, int s) {
if (y>=n) return go(x+1,0,s);
if (x>=m) return (1<<9)+0;
int& ref = dp[x][y][tr[s]];
if (ref != -1) return ref;
assert(touchcnt + 1 < MAXTOUCH);
touched[touchcnt++] = (((tr[s]<<5)+y)<<5)+x;
int maxv = 0, w = 0;
// Check if we can put a mine at (x,y)
if (b[x][y]=='1' && !(y>0 && (s&1)) && !(s&(1<<(n-1)))) {
int res = go(x,y+1,((s<<1)|1)&ALL);
maxv = 1 + (res&511);
w = res>>9;
}
{ // Try not placing a mine at (x,y)
int res = go(x,y+1,(s<<1)&ALL);
if ((res&511) > maxv) maxv = (res&511), w = 0;
if ((res&511) == maxv) {
w += (res>>9);
if (w >= MOD) w -= MOD;
}
}
return ref = (w<<9)+maxv;
}
int main() {
// Make a mapping (and its reverse) from the zeckendorf representation to its value.
int cntpat = 0;
for (int i=0;i<(1<<20);++i) if (isok(i)) {
```

```
tr[i] = cntpat;

rev[cntpat] = i;

++cntpat;

}

memset(dp,-1,sizeof dp);

int ncases;

scanf("%d", &ncases);

for (int z=1;z<=ncases;++z) {

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

dp[touched[i]&31][((touched[i]>>5)&31)[touched[i]>>10] = -1;

touchcnt = 0;
scanf("%d %d", &m, &n);

gets(b[0]);

for (int i=0;i<m;++i) gets(b[i]);

ALL = (1<<n)-1;

int res = go(0,0,0);

printf("%d %d\n", res&511, res>>9);

}

}
```

3. Një mënyrë zgjidhje për ushtrimin e tretë është: (15 pikë)

```

#include <bits/stdc++.h>
using namespace std;

#define MAXN 100010
int ft[MAXN];

void update(int idx)
{
    for(;idx<MAXN;idx+=idx&-idx)
        ++ft[idx];
}

int query(int idx)
{
    int sum = 0;
    for(;idx;idx-=idx&-idx)
        sum += ft[idx];
    return sum;
}

int main()
{
    int n;
    printf("Jepni numrin e nxenesve:");
    scanf("%d",&n);
    vector<pair<int,int> > v(n);
    vector<int> a(n);
    printf("Afishoni vendosjen fillestare te nxenesve:");
    for(int i=0;i<n;++i)
    {
        scanf("%d",&v[i].first);
        v[i].second = i;
    }
    printf("Afishoni si duhet te ishin vendosur nxenesit ne salle:");
    for(int i=0;i<n;++i)
        scanf("%d",&a[i]);
    sort(v.begin(),v.end());
    for(int i=0;i<a.size();++i)
    {
        vector<pair<int,int> >::iterator it;
        it = lower_bound(v.begin(),v.end(),make_pair(a[i],0));
        a[i] = it->second+1;
    }
    long long inv = 0;
    for(int i=n-1;i>=0;--i)
    {
        inv += query(a[i]-1);
        update(a[i]);
    }
    printf("Numri i levizjeve qe duhen kryer eshte: %lld\n",inv);
}

```

4. Një mënyrë zgjidhje për ushtrimin e katërt është: (15 pikë)

```

#include <bits/stdc++.h>
using namespace std;
#define MAXN 100010
int ft[MAXN];
void update(int idx)
{
    for(;idx<MAXN;idx+=idx&-idx)
        ++ft[idx];
}

int query(int idx)
{
    int sum = 0;
    for(;idx;idx-=idx&-idx)
        sum += ft[idx];
    return sum;
}

int main()
{
    int n;
    printf("Jepni numrin e nxenesve:");
    scanf("%d",&n);
    vector<pair<int,int> > v(n);
    vector<int> a(n);
    printf("Afishoni vendosjen fillestare te nxenesve:");
    for(int i=0;i<n;++i)
    {
        scanf("%d",&v[i].first);
        v[i].second = i;
    }
    printf("Afishoni si duhet te ishin vendosur nxenesit ne salle:");
    for(int i=0;i<n;++i)
        scanf("%d",&a[i]);
    sort(v.begin(),v.end());
    for(int i=0;i<a.size();++i)
    {
        vector<pair<int,int> >::iterator it;
        it = lower_bound(v.begin(),v.end(),make_pair(a[i],0));
        a[i] = it->second+1;
    }
    long long inv = 0;
    for(int i=n-1;i>=0;--i)
    {
        inv += query(a[i]-1);
        update(a[i]);
    }
    printf("Numri i levizjeve qe duhen kryer eshte: %lld\n",inv);
}

```