

太戈编程
etiger.vip

信奥算法

位运算

bitwise operation

```
cout<<(1<<31)<<endl;  
cout<<(1LL<<31)<<endl;
```

```
-2147483648  
2147483648
```

```
cout<<(1<<31)-1<<endl;  
cout<<(1LL<<31)-1<<endl;
```

```
2147483647  
2147483647
```

```
cout<<(1LL<<63)<<endl;  
cout<<(1LL<<63)-1<<endl;
```

```
-9223372036854775808  
9223372036854775807
```

高频
错误

位运算移位溢出
建议使用1LL

bitset
位序列

```
1 #include<iostream>
2 #include<bitset> ←
3 using namespace std;
4 int main(){
5     const int N=8; ←
6     bitset<N> x;
7     x[0]=x[2]=x[7]=1;
8     cout<<x<<endl;
9     cout<<x.count()<<endl;
10    x.flip();
11    cout<<x<<endl;
12    x.set();
13    cout<<x<<endl;
14    x.reset();
15    cout<<x<<endl;
16    return 0;
17 }
```

很多题目需要
N=100000


10000101
3

01111010

11111111

00000000

```
1 #include<iostream>
2 #include<bitset>
3 #include<string>
4 using namespace std;
5 int main(){
6     const int N=8;
7     bitset<N> x(15);
8     bitset<N> y(string("11111100"));
9     cout<<x<<endl;
10    cout<<y<<endl;
11
12    cout<<(x&y)<<endl;
13    cout<<(x|y)<<endl;
14    cout<<(x^y)<<endl;
15    cout<<(~y)<<endl;
16    return 0;
}
```



00001111

11111100

00001100

11111111

11110011

00000011

bitset加速优化

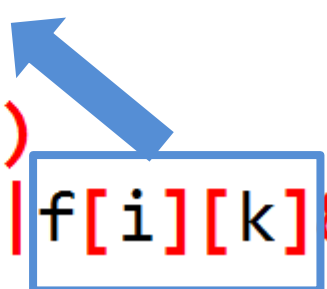
01数组批量运算

Floyd-Warshall加速优化

$f[i][j]$ 最终值代表i号点能否到j号点

枚举中转站k, 起点i, 终点j

```
for(int k=1;k<=n;k++)  
    for(int i=1;i<=n;i++)  
        for(int j=1;j<=n;j++)  
            f[i][j]=f[i][j]||f[i][k]&&f[k][j];
```



```
for(int k=1;k<=n;k++)  
    for(int i=1;i<=n;i++) if(f[i][k])  
        for(int j=1;j<=n;j++)  
            f[i][j]=f[i][j]||f[k][j];
```


Floyd-Warshall加速优化

```
for(int k=1;k<=n;k++)  
  for(int i=1;i<=n;i++)if(f[i][k])  
    for(int j=1;j<=n;j++)  
      f[i][j]=f[i][j]||f[k][j];
```

f[i][]这行 整体或 f[k][]这行

```
4  const int N=109;
5  bitset<N> f[N];
6  int n,x;
7  cin>>n;
8  for(int i=1;i<=n;i++)
9      for(int j=1;j<=n;j++){
10         cin>>x;
11         f[i][j]=x;
12     }
13  for(int k=1;k<=n;k++)
14     for(int i=1;i<=n;i++)if(f[i][k])
15         f[i]|=f[k];
16  for(int i=1;i<=n;i++,cout<<endl)
17     for(int j=1;j<=n;j++)cout<<f[i][j]<<" ";
```

太戈编程2192

按位贪心

识别出二进制每一位计算相对独立

从二进制最高位到最低位,依次分析每一位
找该位为**1**的数字

若找到数字个数至少两个

该位可以取**1 and 1**得**1**
之后集合里只考虑该位为**1**的数
该位为**0**的数不再被考虑

```
13 void solve(){
14     vector<ll> ID[2];
15     ll now=0;
16     for(ll i=1;i<=n;++i)ID[0].push_back(i);
17     ll ans=0;
18     for(ll p=30;p>=0;--p){
19         ll nxt=(now^1);
20         ID[nxt].clear();
21         for(ll i=0;i<ID[now].size();++i){
22             ll id=ID[now][i];
23             if((1LL<<p)&x[id])
24                 ID[nxt].push_back(id);
25         }
26         if(ID[nxt].size()<2)continue;
27         ans^=(1LL<<p);
28         now=nxt;
29     }
30     cout<<ans<<endl;
31 }
```

ID[now]储存待考虑数字编号

太戈编程2193

按位贪心

识别出二进制每一位计算相对独立

从二进制最高位到最低位,依次分析每一位
枚举初始值0或1,根据最终值确定取值

若初始值0可以推导出最终值1
则该位初始值取0

否则,若初始值1可以推导出最终值1
需要初始值在m以内
该位初始值才可以取1

```
26  ll ans=0;
27  for(ll p=30;p>=0;--p){
28      if((1LL<<p)&res0)
29          ans+=(1LL<<p);
30      else if(((1LL<<p)&res1) && ) {
31          ans+=(1LL<<p);
32          
33      }
34  }
35  cout<<ans<<endl;
```


太戈编程2461

连续段xor

m块的块内xor值需要全部为0
最终or值才会取0

块内xor, 即连续段xor
可以转换为两个前缀xor值的xor值

pxor记录前缀异或值
pxor[i]表示 $a[1] \text{ xor } a[2] \text{ xor } \dots \text{ xor } a[i]$

$a[i] \text{ xor } a[i+1] \text{ xor } \dots \text{ xor } a[j]$
等于 $\text{pxor}[i-1] \text{ xor } \text{pxor}[j]$

按位贪心

识别出二进制每一位计算相对独立

从二进制最高位到最低位,依次分析每一位
判断该位结果能否为0

在`pxor[1..n]`对应的二进制位里
有没有至少`m`个0

```
10     ll n,m;
11     scanf("%lld %lld",&n,&m);
12     for(ll i=1;i<=n;++i){
13         scanf("%lld",&a[i]);
14         pxor[i]=pxor[i-1]^a[i];
15     }
```

pxor记录前缀异或值
pxor[i]表示 $a[1] \text{ xor } a[2] \text{ xor } \dots \text{ xor } a[i]$

```

16 ll ans=0;
17 for(ll p=62;p>=0;--p){
18     ll c0=0;
19     if((1LL<<p)&pxor[n]){
20         ans+=(1LL<<p);
21         continue;
22     }
23     for(ll i=1;i<=n;++i){
24         if(DEL[i])continue;
25         if((1LL<<p)&pxor[i])continue;
26         ++c0;
27     }
28     if(c0<m){
29         ans+=(1LL<<p);
30         continue;
31     }
32     for(ll i=1;i<=n;++i){
33         if(DEL[i])continue;
34         if((1LL<<p)&pxor[i])
35             
36     }
37 }

```

太戈编程

2192

2193

2461