省选级别试题 第二组 题解

2703 世界树的考验

解

我们将每个点的点权设为：其周围的所有边的边权的异或和。
这样原来“使所有边权为0”与现在“使所有点权为0”是等价的。
证明：考虑度数为1的点，因为其点权为0，那么其相邻的那一条边边权也必然为0，删去这条边与这个点。持续这样删边，使最终只剩一个点，得证。

这样做就可以将每次操作看为修改两个点的权值：因为对于路径上的点而言，这条路径要经过它相邻的两条边，根据异或的自反性，就可以忽略影响了。

显然，权值相同的点直接选取异或，剩下的点的权值互不相同，状压DP查找当前状态到0的操作次数。

状态s的第i位若为1，则表示图中仍存在边权为i的点，dp[s]就表示将原图的状态转变为状态s需要做的最少操作次数，转移时枚举下一步操作要将哪两个点异或即可

标准代码

C++：

```
//#include<bits/stdc++.h>
#include<iostream>
#include<cstdlib>
#include<cstdio>
#include<cstring>
#define N 100005
#define M 65540
using namespace std;
int n,s[N],S,ans,f[M],inf,sum[20];

int dfs(int S)
{
  if(!S)return 0;
  if(f[S]<inf)return f[S];
  for(int i=0;i<16;i++)if((S>>i)&1)
    for(int j=0;j<16;j++)if(i!=j&&(S>>j)&1)
    {
      int p=i^j,x=S^(1<<i)^(1<<j)^(1<<p);
      if(S>>p&1)f[S]=min(f[S],dfs(x)+2);
      else f[S]=min(f[S],dfs(x)+1);
      //枚举每次异或的两个节点i,j,i异或掉j,j异或掉j,还剩下p=i^j
      //如果p本身存在那两个p直接消掉
    }
  return f[S];
}

int main()
{
```

```
  int a,b,c;
  scanf("%d",&n);
  for(int i=1;i<n;i++)
    scanf("%d%d%d",&a,&b,&c),s[a]^=c,s[b]^=c;
  for(int i=0;i<n;i++)sum[s[i]]++;
  for(int i=1;i<16;i++)
    ans+=(sum[i]>>1),S+=(1<<i)*(sum[i]&1);
  memset(f,127,sizeof(f));inf=f[0];
  printf("%d\n",ans+dfs(S));
  return 0;
}
```

2707 蓝精灵的请求

解

将一个图分成两个子图，使得每一个子图都是完全图。
所有不相连的点不能再一个子图里，那么我们就对这些不相连点进行建边，然后二分图染色（即建立了一个原图的补图）。

在新图中，同一个连通块中且是相同颜色的点一定在同一个子图里，**而不同连通块中不同颜色的点可以放在一起。**

s[i]表示当前是否存在一种分组方式使得某组的蓝精灵数为i，枚举每个联通块s数组即可。

标准代码

C++：

```
#include<cstdio>
#include<cstring>
#include<algorithm>

using namespace std;

const int N = 705;
const int M = 490005;
int n, m, u, v, ans = 0x7fffffff;
bool g[N][N], s[N], t[N];
int col[N], cnt[2];

void dfs(int x, int c){
    col[x] = c;
    cnt[c == 1]++;
    for(int i = 1; i <= n; i++){
        if(i == x || g[x][i])   continue;
        if(!col[i]) dfs(i, -c);
        else if(col[i] == c){
            puts("-1");
            exit(0);
        }
    }
}
```

```
int main(){
    scanf("%d%d", &n, &m);
    for(int i = 1; i <= m; i++){
        scanf("%d%d", &u, &v);
        g[u][v] = g[v][u] = 1;
    }
    s[0] = 1;
    for(int i = 1; i <= n; i++){
        if(col[i])  continue;
        cnt[0] = cnt[1] = 0;
        dfs(i, 1);
        memset(t, 0, sizeof t);
        for(int j = 0; j <= n; j++){
            t[j+cnt[0]] |= s[j];
            t[j+cnt[1]] |= s[j];
        }
        for(int j = 0; j <= n; j++)
            s[j] = t[j];
    }
    for(int i = 0; i <= n; i++)
        if(s[i])
            ans = min(ans, i*(i-1)/2 + (n-i)*(n-i-1)/2);
    printf("%d\n", ans);
    return 0;
}
```

2708 青青草原的表彰大会

解

不妨设第i只羊获得的奖金为a_i，那么a序列中不同的数的个数不超过$O(logn)$，且所有数字按升序排列，因此可以枚举a序列中不同的数的个数d,不妨假设这些不同的数分别为$p_1,p_2,p_3...p_d$，考虑统计p序列的个数： 用$f[i][j]$表示长度为i且$p_i=j$的p序列个数，用$f[i+1][t*j]+=f[i][j]$来更新即可。那么$cnt[i]=sigma(f[i][j])$即为长为i的p序列个数。

接下来考虑求出$p_1,p_2,p_3...p_d$放入a序列中的方案数C_{k-1}^{d-1}，用隔板法可以求出这个方案数为,因而总的方案数为$\sum_{i=1}^{logk} cnt[d]*C_{k-1}^{d-1}$

标准代码

C++11：

```
//#include<bits/stdc++.h>
#include<cstdlib>
#include<cstring>
#include<cstdio>
#include<cmath>
#define mod 1000000007
#define N 1000020
#define M 22
using namespace std;
inline int read()
{
```

```
    int x=0,f=1;char ch=getchar();
    while(ch<'0'||ch>'9'){if(ch=='-')f=-1;ch=getchar();}
    while(ch>='0'&&ch<='9'){x=x*10+ch-'0';ch=getchar();}
    return x*f;
}
int K,n;
int qm(int x,int y)
{
    int ret=1;
    while(y)
    {
        if(y&1)ret=1ll*ret*x%mod;
        x=1ll*x*x%mod;y>>=1;
    }
    return ret;
}
int f[M][N],cnt[M];
int pre[N],ans;
int C(int x,int y){return 1ll*pre[x]*qm(pre[y],mod-2)%mod*qm(pre[x-y],mod-
2)%mod;}
int main()
{
    pre[0]=1;for(int i=1;i<N;i++)pre[i]=1ll*pre[i-1]*i%mod;
    n=read(),K=read();
    for(int j=1;j<=n;j++)f[1][j]=1;
    for(int i=1;i<M-1;i++)
        for(int j=1;j<=n;j++)
            for(int k=j+j;k<=n;k+=j)
                f[i+1][k]=(f[i+1][k]+f[i][j])%mod;
    for(int i=1;i<M;i++)
        for(int j=1;j<=n;j++)cnt[i]=(cnt[i]+f[i][j])%mod;
    for(int i=1;i<M&&i<=K;i++)ans=(ans+1ll*cnt[i]*C(K-1,i-1)%mod)%mod;
    printf("%d\n",ans);
}
```