

ABC244

E.King Bombee

先不考虑X出现次数的奇偶问题

需要寻找的是从S到T的长度为K+1的不同路径数量

$dp[i][j]$ 表示从S出发,通过i步走到j时不同路径数量

现在考虑X的数量: $dp[i][j][e]$ e表示此时经过X的数量奇偶

可以得到递推公式: $dp[i+1][t][x \oplus (t == X)] += dp[i][f][x]$

从0~k出发,通过每轮使用所有边更新状态,可以得到答案为 $dp[K][T][0]$

```
using ll = long long;
const int max_n = 2e3+10;
const int mod_ = 998244353;

std::vector<std::pair<int,int>>edge;
int N,M,K,S,T,X;
int d1,d2;
//ll ans = 0;

int main(){
    scanf("%d %d %d %d %d %d",&N,&M,&K,&S,&T,&X);
    for(int i = 1;i <= M;i++){
        scanf("%d %d",&d1,&d2);
        edge.push_back({d1,d2});
    }
    //
    std::vector dp(K+1,std::vector(N+1,std::array<ll,2>{{0,0}}));
    //
    dp[0][S][0] = 1ll;
    for(int i = 0;i < K;i++){
        for(auto [f,t] : edge)
            for(int x : {0,1}){
                (dp[i+1][t][x ^ (t == X)] += dp[i][f][x]) %= mod_;
                (dp[i+1][f][x ^ (f == X)] += dp[i][t][x]) %= mod_;
            }
    }
    printf("%lld\n",dp[K][T][0]);
    return 0;
}
```

F.Shortest Good Path

S串01表示顶点出现次数的奇偶,可以将此转换为整数存储

$dis[i][j]$ 表示i整数对应S序列,路径最后为顶点j时的最短路径长

所以递推公式有 $dis[p_s][eg] = dis[s][p] + 1$

其中eg p为联通点

$p_s = (s \oplus (1 \ll (eg - 1)))$ 为最后加入eg点后,对应的S序列值

对每个顶点 i : $dis[1 \leq i-1][i] = 1$

于是从 $\{1 \leq i-1, i\}$ 出发进行bfs后

$dis[1 \sim (1 \leq n)-1][1 \sim n]$ 中的最小值即为最小路径长度,和即为答案

```
const int max_n = 20;
std::vector<int> edge[max_n];
std::deque<std::pair<int,int>> que;
int dis[1<17][18];
int n,m;
long long ans = 0;

int main(){
    scanf(" %d %d",&n,&m);
    int f,t;
    for(int i = 1;i <= m;i++){
        scanf(" %d %d",&f,&t);
        edge[f].push_back(t);
        edge[t].push_back(f);
    }

    int N = 1<n;
    for(int i = 1;i < N;i++)
        for(int j = 1;j <= n;j++)
            dis[i][j] = INT32_MAX;
    for(int i = 0;i < n;i++){
        dis[1<i][i+1] = 1;
        que.push_back({1<i,i+1});
    }

    while(!que.empty()){
        int s = que.front().first, p = que.front().second;
        que.pop_front();
        for(auto eg : edge[p]){
            int p_s = s ^ (1<(eg-1)); // 将eg加入后的s数值
            //已有,应为更小解
            if(dis[p_s][eg] < INT32_MAX)
                continue;
            dis[p_s][eg] = dis[s][p] + 1;
            que.push_back({p_s,eg});
        }
    }

    for(int i = 1;i < N;i++){
        int Val_s = INT32_MAX;
        for(int j = 1;j <= n;j++)
            Val_s = std::min(Val_s,dis[i][j]);
        ans += (long long)Val_s;
    }
    printf("%lld\n",ans);
    return 0;
}
```

Submission Time	Task	User	Language	Score	Code Size	Status	Exec Time	Memory	
2022-03-22 18:38:31	F - Shortest Good Path	tx995976 	C++ (Clang 10.0.0)	500	1328 Byte	AC	115 ms	16064 KB	Detail
2022-03-21 21:32:36	E - King Bombee	tx995976 	C++ (GCC 9.2.1)	500	874 Byte	AC	94 ms	65956 KB	Detail
2022-03-20 20:45:07	D - Swap Hats	tx995976 	C++ (Clang 10.0.0)	400	641 Byte	AC	10 ms	3164 KB	Detail
2022-03-20 20:29:38	C - Yamanote Line Game	tx995976 	C++ (Clang 10.0.0)	300	492 Byte	AC	27 ms	3852 KB	Detail
2022-03-20 20:09:44	B - Go Straight and Turn Right	tx995976 	C++ (Clang 10.0.0)	200	538 Byte	AC	13 ms	3272 KB	Detail
2022-03-20 20:01:34	A - Last Letter	tx995976 	C++ (Clang 10.0.0)	100	202 Byte	AC	12 ms	3188 KB	Detail