

I 小刻的DP题

出题人:buns out

考察点:思维,状压,最短路,bfs

宣传一下我的模板库:

[Release ACM模板 · zxyDEDIRE/ACM-Code-Library \(github.com\)](#)

[zxyDEDIRE/ACM-Code-Library at QAQ \(github.com\)](#)

有LaTeX源码也有PDF版本

非常经典的最短路问题, 状压最短路, 此题也有非常非常多的解法, 时限放到了 $5s$, 除了非常非常暴力的做法, 一般都可以过。但是过的人有点少, 是大家没有很研究图论?

基本的思想就是状压维护最短路, 因为 k 只有 8 , 因此可以用二进制状态压缩维护 8 个二进制位, 如果说此二进制位上是 1 , 那么说明走过的路包含此位的佩洛, 这个为主要思想。

那么接下来询问 q 次, 可以先预处理出所有的情况, 或者说每次都跑一遍 DIJ 。

预处理的话 dij, bfs 都是可以的, 但是每次询问每次跑一边的话, 最好还是用 dij , 不然会T。

```
#include<bits/stdc++.h>
using namespace std;
#define endl "\n"
using ll=long long;
const ll INF=1e18+7;
const int maxn=1e6+100;
const int N=1e5+100;
struct node{
    ll id,op,dis;
    bool operator<(const node&a)const{
        return a.dis<dis;
    }
};
vector<array<ll,3>>v[maxn];
ll dis[N][550];
bool vis[N][550];
int n,m,k,s,t,q;
void dij()
{
    for(int i=1;i<=n;i++)
        for(int j=0;j<=(1<<(k+1));j++)
            dis[i][j]=INF;
    dis[s][0]=0;
    // priority_queue<node>q;
    queue<node>q;
    q.push({s,0,0});
    while(!q.empty())
    {
        // auto [x,f1,DIS]=q.top();q.pop();
```

```

        auto [x, fl, DIS] = q.front(); q.pop();
        // cout<<x<<" "<<fl<<" "<<DIS<<endl;
        if(DIS>dis[x][fl]) continue;
        for(auto [y, w, _op]: v[x])
        {
            int op = (_op | fl);
            if(dis[y][op]>dis[x][fl]+w)
            {
                dis[y][op] = dis[x][fl]+w;
                q.push({y, op, dis[y][op]});
            }
        }
    }
}

void solve()
{
    cin>>n>>m>>k>>s>>t;
    for(int i=1; i<=m; i++)
    {
        int op, x, y, w;
        cin>>x>>y>>w>>op;
        op--;
        v[x].push_back({y, w, 1<<op});
        v[y].push_back({x, w, 1<<op});
    }
    dij();
    // for(int i=0; i<(1<<k); i++)
    // for(int j=0; j<(1<<k); j++)
    //     if(i!=j&&(i&j)==i){
    //         for(int k=1; k<=n; k++)
    //             dis[k][j]=min(dis[k][i], dis[k][j]);
    //     }
    cin>>q;
    while(q-->0)
    {
        int num, fl=0, x;
        cin>>num;
        while(num-->0){
            cin>>x;
            x--;
            fl |= (1<<x);
        }
        ll ans=INF;
        for(int i=0; i<=(1<<k); i++)
            if((i&fl)==0)
                ans=min(ans, dis[t][i]);

        if(ans==INF) cout<<-1<<endl;
        else cout<<ans<<endl;
    }
}

signed main(){
    ios::sync_with_stdio(false);
    cin.tie(nullptr); cout.tie(nullptr);
    solve();
}

```

```
return 0;
```

```
}
```

```
/*
```

```
*/
```