类 Dinic 算法

我们可以在 Dinic 算法的基础上进行改进，把 BFS 求分层图改为用 SPFA（由于有负权边，所以不能直接用 Dijkstra）来求一条单位费用之和最小的路径，也就是把 $w(u,v)$ 当做边权然后在残量网络上求最短路，当然在 DFS 中也要略作修改。这样就可以求得网络流图的 **最小费用最大流** 了。

如何建 **反向边**？对于一条边 (u,v,w,c)（其中 w 和 c 分别为容量和费用），我们建立正向边 (u,v,w,c) 和反向边 $(v,u,0,-c)$（其中 $-c$ 是使得从反向边经过时退回原来的费用）。

优化：如果你是“关于 SPFA 它死了”言论的追随者，那么你可以使用 Primal-Dual 原始对偶算法将 SPFA 改成 Dijkstra。

时间复杂度：可以证明上界为 $O(nmf)$，其中 f 表示流量。

参考代码：

```
#include<bits/stdc++.h>
using namespace std;
const int N=5e3+5,M=1e5+5,INF=0x3f3f3f3f;
int head[N],cur[N],to[M],nxt[M],dis[N],val[M],tot=1,cost[M];
bool vis[N];
bool adj[N][N];
void add(int u,int v,int c,int w){
    nxt[++tot]=head[u];
    head[u]=tot;
    to[tot]=v;
    val[tot]=c;
    cost[tot]=w;
}
void addedge(int u,int v,int c,int w){
    add(u,v,c,w);
    add(v,u,0,-w);
}
bool spfa(int s,int t){
    memset(dis,0x3f,sizeof(dis));
    queue<int>que;
    que.push(s);
    dis[s]=0;
    vis[s]=1;
    cur[s]=head[s];
    while(!que.empty()){
        int u=que.front();
        que.pop();
        vis[u]=0;
        for(int i=head[u];i;i=nxt[i]){
            int v=to[i];
            if(val[i]&&dis[v]>dis[u]+cost[i]){
                dis[v]=dis[u]+cost[i];
                cur[v]=head[v];
```

```
                if(!vis[v])
                    que.push(v),vis[v]=1;
            }
        }
    }
    return dis[t]!=INF;
}
int Cost;
int dfs(int u,int ret,int s,int t){
    if(u==t||ret==0) return ret;
    int Flow=0;
    vis[u]=1;
    for(int i=cur[u];i&&ret;i=nxt[i]){
        int v=to[i];
        cur[u]=i;
        if(!vis[v]&&val[i]>0&&dis[v]==dis[u]+cost[i]){
            int f=dfs(v,min(ret,val[i]),s,t);
            if(f==0) vis[v]=0;
            Cost+=f*cost[i];
            val[i]-=f;
            val[i^1]+=f;
            Flow+=f;
            ret-=f;
        }
    }
    vis[u]=0;
    return Flow;
}
int mcmf(int s,int t){
    int Flow=0;
    while(spfa(s,t)){
        Flow+=dfs(s,INF,s,t);
    }
    return Flow;
}
int main(){
//    freopen("P3381_8.in","r",stdin);
    int n,m,s,t;
    scanf("%d %d %d %d",&n,&m,&s,&t);
    for(int i=1;i<=m;i++){
        int u,v,c,w;
        scanf("%d %d %d %d",&u,&v,&c,&w);
        if(!adj[u][v])
            addedge(u,v,c,w);
        adj[u][v]=1;
    }
    printf("%d",mcmf(s,t));
    printf(" %d",Cost);
    return 0;
```

```
}
```

习题

【Luogu 3381】【模板】最小费用最大流

【Luogu 4452】航班安排

【SDOI 2009】晨跑

【SCOI 2007】修车

【HAOI 2010】订货

【NOI 2012】美食节

参考链接

OI Wiki