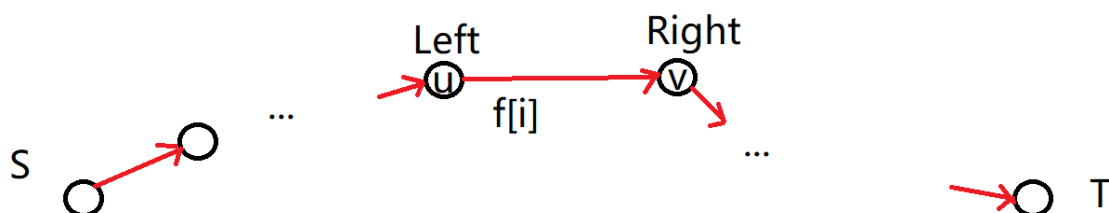


网络流_6 关键边

n 个点 m 条边的网络，给定源点 S 和汇点 T ，求如果有这样边：只给其扩大容量之后整个流网络的最大流能够变大，对于这样的边我们称之为**关键边**。求这样的边的个数。

思路：



- 1.求当前网络上的最大流，任意一个都行
- 2.在当前最大流的残留网络上进行搜索，搜索在其残留网络上沿着容量大于0的边走，所有从源点能够到达的点和所有汇点能够到达的点，并打上标记。
- 3.对于所有关键边 v 都满足这样的性质：

- $f(u,v) < c(u,v)$
- S 可达 `point v.left` , T 可达 `point v.right`

因此我们在求完最大流之后，只需要枚举每一条正向边，当此边的做右端点都满足上述性质时，说明此边满足要求，为关键边。

```
const int INF = 1e9;
const int N = 550,M = (N + 5010)*2;
int n,m,k,S,T,ans=0;
int e[M],ne[M],f[M],h[N],idx = 0;
int d[N],cur[N],vis_s[N],vis_t[N],q[M];

inline void add(int a,int b,int c){
    e[idx] = b , f[idx] = c , ne[idx] = h[a] , h[a] = idx ++;
    e[idx] = a , f[idx] = 0 , ne[idx] = h[b] , h[b] = idx ++;
}

void dfs(int st[],int u,int k){
    st[u] = true;
    for(int i=h[u];~i;i=ne[i]){
        int j = i ^ k , ver = e[i];
        if(f[j] && !st[ver])
            dfs(st,ver,k);
    }
}

bool bfs(){
    memset(d,-1,sizeof d);
```

```

int hh = 0 , tt = 0;
q[0] = s , d[s] = 0 , cur[s] = h[s];

while(hh <= tt){
    int u = q[hh ++];
    for(int i=h[u];~i;i=ne[i]){
        int ver = e[i];
        if(d[ver] == -1 && f[i]){
            d[ver] = d[u] + 1;
            cur[ver] = h[ver];
            if(ver == T) return true;
            q[++ tt] = ver;
        }
    }
}
return false;
}

int find(int u,int limit){
    if(u == T) return limit;
    int flow = 0;

    for(int i=cur[u];~i && flow < limit ; i = ne[i]){
        cur[u] = i;

        int ver = e[i];
        if(d[ver] == d[u] + 1 && f[i]){
            int t = find(ver,min(f[i],limit-flow));
            if(!t) d[ver] = -1;
            f[i] -= t , f[i^1] += t , flow += t;
        }
    }
    return flow;
}

int dinic(){
    int ans = 0 , flow = 0;
    while(bfs()) while(flow = find(S,INF)) ans += flow;
    return ans;
}

//=====
int main(){
    memset(h,-1,sizeof h);

    n = read() , m = read();
    S = 0 , T = n - 1;
    rep(i,1,m){
        int a = read() , b = read() , c = read();
        add(a,b,c);
    }

    dinic();

    dfs(vis_s,S,0);
    dfs(vis_t,T,1);

    for(int i=0;i<idx;i+=2){

```

```
        if(vis_s[e[i^1]] && vis_t[e[i]] && !f[i])
            ans ++ ;
    }

    print(ans);

    return 0;
}
```