

## 单源最短路径(spfa&判负环)

```
#include <bits/stdc++.h>
using namespace std;
const int maxn=3005;
const int maxm=3010;
int dis[maxn], n, m, s, u, v, w, head[maxn], tot, cnt[maxn];
bool vis[maxn];
queue<int> q;
struct node{
    int to, w, nxt;
} e[maxm<<1];
void add(int u, int v, int w){
    e[++tot].to=v;
    e[tot].w=w;
    e[tot].nxt=head[u];
    head[u]=tot;
}
void SPFA(){
    fill(dis+1, dis+n+1, INT_MAX);
    memset(cnt, 0, sizeof(cnt));
    memset(vis, 0, sizeof(vis));
    while (!q.empty()) q.pop(); // 特别注意一下
    dis[s]=0; vis[s]=1;
    q.push(s);
    while (!q.empty()){
        int u=q.front(); q.pop();
        vis[u]=0;
        for(int i=head[u]; i; i=e[i].nxt){
            int v=e[i].to;
            if(dis[v]>dis[u]+e[i].w){
                dis[v]=dis[u]+e[i].w;
                if(!vis[v]){
                    if(++cnt[v]>n){
                        printf("YES\n"); return;
                    }
                    vis[v]=1; q.push(v);
                }
            }
        }
    }
    printf("NO\n");
}
void input(){
    cin >> n >> m;
    s=1; tot=0;
    memset(head, 0, sizeof(head));
    for(int i=1; i<=m; i++) {
        cin >> u >> v >> w;
        add(u, v, w);
        if(w>=0) add(v, u, w);
    }
}
int main(){
}
```

```
int tt;
cin >> tt;
while (tt--){
    input();
    SPFA();
}
return 0;
}
```