

SPFA+Dijkstra 最短路模板

参考代码

```
#include<bits/stdc++.h>
using namespace std;
const int N=1e5+5,M=1e6+5,INF=2147483647;
int head[N],to[M],nxt[M],dis[N],cost[M];
bool vis[N];
int tot=1;
int n;
void add(int u,int v,int c){
    nxt[++tot]=head[u];
    head[u]=tot;
    to[tot]=v;
    cost[tot]=c;
}
void dijkstra(int s){
    for(int i=1;i<=n;i++) dis[i]=INF;
    dis[s]=0;
    priority_queue<pair<int,int>,vector<pair<int,int>>,greater<pair<int,int>>>pq;
    pq.push(make_pair(0,s));
    while(!pq.empty()){
        int u=pq.top().second;
        pq.pop();
        if(vis[u]) continue;
        vis[u]=1;
        for(int i=head[u];i;i=nxt[i]){
            int v=to[i];
            if(!vis[v]&&dis[u]+cost[i]<dis[v]){
                dis[v]=dis[u]+cost[i];
                pq.push(make_pair(dis[v],v));
            }
        }
    }
}
void spfa(int s){
    for(int i=1;i<=n;i++) dis[i]=INF;
    queue<int>que;
    vis[s]=1;
    dis[s]=0;
    que.push(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(!vis[v]&&dis[u]+cost[i]<dis[v]){
                dis[v]=dis[u]+cost[i];
                que.push(v);
            }
        }
    }
}
```

```
        int v=to[i];
        if(dis[v]>dis[u]+cost[i]){
            dis[v]=dis[u]+cost[i];
            if(!vis[v]){
                vis[v]=1;
                que.push(v);
            }
        }
    }
}

int main(){
    int m,s;
    scanf("%d %d %d",&n,&m,&s);
    while(m--){
        int u,v,w;
        scanf("%d %d %d",&u,&v,&w);
        add(u,v,w);
    }
    dijkstra(s);
    // spfa(s);
    for(int i=1;i<=n;i++) printf("%d ",dis[i]);
    return 0;
}
```

From:
<https://wiki.cvbbacm.com/> - CVBB ACM Team

Permanent link:
https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:legal_string:lgwza:spfa_dijkstra_%E6%9C%80%E7%9F%AD%E8%B7%AF%E6%A8%A1%E6%9D%BF

Last update: 2020/08/21 08:53

