

## 单源最短路径(dijkstra)

```
#include <bits/stdc++.h>
using namespace std;
typedef long long ll;
const int MaxM=500010,MaxN=100010;
const ll oo=0x7fffffffffffffff;
int n,m,tot=0;
int head[MaxN];
ll dis[MaxN];
bool vis[MaxN];
struct edge{
    int to,next;
    ll dis;
}e[MaxM];
struct node{
    ll dis;int pos;
    bool operator<(const node &x)const{return x.dis<dis;}
};

inline void add(int u,int v,int d){
    e[++tot].dis=d;
    e[tot].to=v;
    e[tot].next=head[u];
    head[u]=tot;
}

priority_queue<node>q;
inline void dijkstra(int s){
    for(int i=1;i<=n;i++) dis[i]=oo;
    memset(vis,0,sizeof(vis));
    dis[s]=0;
    q.push((node){0,s});
    while (!q.empty()){
        node tmp=q.top();
        q.pop();
        int x=tmp.pos;
        ll d=tmp.dis;
        if(vis[x]) continue;
        vis[x]=1;
        for(int i=head[x];i;i=e[i].next){
            int y=e[i].to;
            if(dis[y]>dis[x]+e[i].dis){
                dis[y]=dis[x]+e[i].dis;
                if(!vis[y]) q.push((node){dis[y],y});
            }
        }
    }
}

int main(){
    return 0;
}
```

