朴素Dijkstra

```
#include <bits/stdc++.h>
using namespace std;

const int N = 1000 + 5;
int n, m, s;
int g[N][N], dist[N];
bool vis[N];

void dijkstra(){
    memset(dist, 0x3f, sizeof(dist));
    dist[s] = 0;
    for(int i = 0; i < n; i++){
        int t = -1;
        for(int j = 1; j <= n; j++){
            if(!vis[j] && (t == -1 || dist[j] < dist[t])){
                t = j;
            }
        }
        vis[t] = true;
        for(int j = 1; j <= n; j++){
            dist[j] = min(dist[j], dist[t] + g[t][j]);
        }
    }
}

int main(){
    cin >> n >> m >> s;
    memset(g, 0x3f, sizeof(g));
    for(int i = 0; i < m; i++){
        int u, v, w;
        scanf("%d%d%d", &u, &v, &w);
        g[u][v] = min(g[u][v], w);
    }

    dijkstra();

    for(int i = 1; i <= n; i++){
        printf("%d ", dist[i]);
    }
    return 0;
}
```