# 实验四：最短路径算法

**一、实验目的**

（1）理解路径松驰技术的思想

（2）掌握迪杰斯特拉算法（Dijkstra）的基本思想

**二、实验内容**

实现单源最短路经的迪杰斯特拉算法（Dijkstra），要求：采用最小堆实现优先级队列。

1. **迪杰斯特拉算法（Dijkstra）基本思想**

**设dis[i]为源点到i的距离**

**每次选取未访问的点中距离最小的点，通过该点的所以路径更新dis**

**知道所以点都被访问**

1. **实验过程**

#include <bits/stdc++.h>

using namespace std;

const int N = 1e2+7;

int n, m;

int dis[N];

vector<pair<int, int>> e[N];

inline void Dijkstra() {

  priority\_queue<pair<int,int>,vector<pair<int,int> >,greater<pair<int,int>>> q;

  memset(dis, 0x7f, sizeof dis);

  dis[1] = 0;

  q.push(make\_pair(0, 1));

  while(q.size()) {

    int u = q.top().second;

    int d = q.top().first;

    q.pop();

    if (dis[u] < d) continue;

    for (auto &p : e[u]) {

      int v = p.first;

      int w = p.second;

      if(d+w >= dis[v]) continue;

      dis[v] = d+w;

      q.push(make\_pair(dis[v], v));

    }

  }

}

signed main() {

  cin >> n >> m;

  for (int i = 1, u, v, w; i <= m; ++i) {

    cin >> u >> v >> w;

    e[u].emplace\_back(v, w);

    e[v].emplace\_back(u, w);

  }

  Dijkstra();

  for (int i = 1; i <= n; ++i) {

    cout << dis[i] << " \n"[i==n];

  }

  return 0;

}

1. **实验结果**

输入

3 3

1 2 1

1 3 10

2 3 2

输出

0 1 3

