

{% note info %} **摘要** Title: 850. Dijkstra求最短路 II Tag: 最短路、Dijkstra Memory Limit: 64 MB Time Limit: 1000 ms {% endnote %}

Powered by:NEFU AB-IN

[Link](#)

@TOC

850. Dijkstra求最短路 II

- 题意

给定一个 n 个点 m 条边的有向图，图中可能存在重边和自环，所有边权均为正值。请你求出 1 号点到 n 号点的最短距离，如果无法从 1 号点走到 n 号点，则输出 -1 。

- 思路

堆优化Dijkstra的板子 $1 \leq n, m \leq 1.5 \times 10^5$ 属于稀疏图

{% note info %} **如何处理自环和重边** 自环不必处理，因为Dijkstra默认的运用环境，就是无负权的边，所以不会有环在最短路中 重边不必处理，因为会自动挑选出最小的 {% endnote %}

- 代码

```
...
Author: NEFU AB-IN
Date: 2022-03-02 22:20:20
FilePath: \ACM\Acwing\849.py
LastEditTime: 2022-03-02 22:34:56
...

import heapq

N = int(1e3 + 10)
INF = int(2e9)
st, dist = [0] * N, [INF] * N
g = [[] for _ in range(N)]

def dij(s):
    dist[s] = 0
    q = []
    heapq.heappush(q, [0, s])
    while q:
        t = heapq.heappop(q)
        distance, u = t
        if st[u]:
            continue
        st[u] = 1
```

```
        for v, w in g[u]:
            if dist[v] > dist[u] + w:
                dist[v] = dist[u] + w
                heapq.heappush(q, [dist[v], v])
    if dist[n] == INF:
        return -1
    return dist[n]

n, m = map(int, input().split())
for i in range(m):
    x, y, z = map(int, input().split())
    g[x].append([y, z])

print(dij(1))
```