

{% note info %} **摘要** Title: 853. 有边数限制的最短路 Tag: Bellman-ford、最短路 Memory Limit: 64 MB Time Limit: 1000 ms {% endnote %}

Powered by: NEFU AB-IN

[Link](#)

@TOC

853. 有边数限制的最短路

- 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，边权可能为负数。请你求出从 1 号点到 n 号点的最多经过 k 条边的最短距离，如果无法从 1 号点走到 n 号点，输出 impossible。

- 思路

有边数限制的最短路只能用Bellman-ford来做

Bellman - Ford 算法

for n 次

for 所有边 a, b, w

松弛操作

$dis[b] = \min(dis[b], dis[a] + w);$

$$dis[b] \leq dis[a] + w$$

三角不等式

{% note info %} **外层迭代n次的含义？** 迭代了 k 次，代表从起点经过不少于 k 条边的走到每个点的最短距离 {% endnote %} {% note info %} **为什么不少于？** 因为有可能发生串联，即刚更新完 b ，又拿 b 去更新别的点，这样一次迭代可以更新两条边 {% endnote %} {% note info %} **如何实现有边数限制？** 实现一个备份数组，每次是从备份数组中的数去更新，保证不是实时更新 {% endnote %}

- 代码

平时写的时候不用加**备份数组**

```
'''
Author: NEFU AB-IN
Date: 2022-03-04 14:01:18
FilePath: \ACM\Acwing\853.py
LastEditTime: 2022-03-04 14:49:56
'''

from copy import deepcopy

N = 550
INF = int(2e9)

class Edge():
    def __init__(self, u, v, w):
        self.u = u
        self.v = v
        self.w = w

dist = [INF] * N

def bellman_ford():
    dist[1] = 0
    for i in range(k): #迭代k次
        backup = deepcopy(dist)
        for j in range(m): #遍历m条边
            u, v, w = lst[j].u, lst[j].v, lst[j].w
            dist[v] = min(dist[v], backup[u] + w) #用backup数组更新
    if dist[n] > INF // 2:
        return INF
    return dist[n]

lst = []
n, m, k = map(int, input().split())
for i in range(m):
    u, v, w = map(int, input().split())
    lst.append(Edge(u, v, w))

res = bellman_ford()
if res == INF:
    print("impossible")
else:
    print(res)
```