Dijkstra算法

题目描述

给定邻接矩阵表示的带权有向图，其中每条边的权都是非负整数，并以-1代表不连通。给定图中的一个顶点V，称为源点。现在要计算从源到所有其它各顶点的最短路长度。

输入格式

输入为一行整数，以单一空格隔开，最后一个整数的后面无空格。

第1个数n为节点总数

第2个数至第n^2+1个数为邻接矩阵，以行优先方式存储（即：第2至第$n+1$个数为矩阵第一行，以此类推）

第n^2+2个数为源点编号，节点编号从0开始

输出格式

一行共$n-1$个数，以单一空格隔开，依次代表源点到除源点外的各个节点的最短路径。最后一个数后面无空格，无换行、回车符。

样例 #1

样例输入 #1

```
2 0 1 1 0 0
```

样例输出 #1

```
1
```

样例 #2

样例输入 #2

```
3 0 1 1 1 0 1 1 1 0 1
```

样例输出 #2

```
1 1
```

提示

$100 \leq n \leq 1000$

路径距离、最短距离均在int范围内

solution

```
def solve(n, matrix, v):
    ans = matrix[v].copy()
    to_scan = set(range(n)) - {v}
    for _ in range(n - 1):
        to_scan.remove(index := min([i for i in to_scan if ~ans[i]],
key=lambda x: ans[x]))
        total = ans[index]
        row = matrix[index]
        for i in to_scan:
            if ~row[i] and ((dist := row[i] + total) < ans[i] or
ans[i] == -1):
                ans[i] = dist
```

```
    return ans[:v] + ans[v + 1:]


if __name__ == '__main__':
    print(*map(str, solve(lth := next(numbers := map(int,
input().split())),
                          [[next(numbers) for _ in range(lth)] for _
in range(lth)],
                          next(numbers))), end="")
```

> 因为当时测试点有误，所以白写了一个检验脚本 `checker.py`

```
from solution import solve
from pickle import load

inp = load(open("input.pkl", "rb"))
ans = [1829, 2179, 648, 1592, 1345, 1266, 2274, 1444, 1262, 2210,
1503, 791, 2536, 2477, 2316, 885, 2112, 1781, 1975,
       1848, 1717, 1954, 1627, 1700, 1450, 1479, 1938, 1834, 2849,
1389, 1850, 2258, 2683, 1354, 2130, 2026, 1845, 2005,
       2394, 2066, 1107, 1902, 1532, 610, 1910, 2236, 555, 2096,
1751, 1084, 2229, 1663, 166, 2485, 2067, 1338, 1718,
       1166, 829, 1722, 1319, 1684, 1629, 1298, 1392, 587, 1677,
2783, 1321, 21, 1983, 1369, 2041, 1636, 1584, 1878,
```

```
       777, 1619, 1702, 1898, 1966, 1886, 1031, 1580, 2088, 735,
1410, 1651, 883, 1689, 1547, 1481, 2117, 1769, 1590,
       1322, 2365, 1566, 1899, 1404, 1408, 1411, 1129, 1990, 1395,
2715, 1445, 536, 829, 2660, 1034, 2018, 892, 1617,
       2373, 1480, 1648, 1528, 2221, 1353, 1897, 1241, 509, 2187,
833, 1508, 1380, 2239, 2203, 1173, 1974, 948, 2074,
       2313, 1962, 1941, 2224, 1403, 1464, 905, 1884, 2444, 1553,
2165, 3317, 1860, 2141, 1318, 3043, 527, 477, 2303,
       1478, 363, 892, 1407, 1825, 1287, 1489, 1109, 1063, 1782,
2607, 2081, 1293, 2008, 1469, 2295, 1362, 3493, 2009,
       2240, 1637, 676, 768, 1530, 2220, 2249, 2618, 2115, 2419,
936, 1813, 917, 1667, 1850, 1122, 1846, 2260, 1720,
       2558, 1671, 546, 1176, 1831, 1505, 2487, 2787, 1533, 1288,
1409, 1779, 2494, 2468, 2495, 702, 2126, 1404, 1968,
       1516, 903, 1219, 2242, 830, 1025, 1801, 1381, 1830, 1949,
2277, 2368, 557, 1392, 1696, 1375, 1603, 1316, 2923,
```

```
       1318, 2338, 1547, 1644, 2095, 2632, 2584, 1874, 1618, 882,
1485, 1979, 2473, 2486, 1982, 2275, 1179, 1922, 1592,
       1503, 1719, 1648, 2516, 2026, 1713, 1050, 1171, 3172, 1412,
1465, 1538, 1517, 768, 2523, 1610, 1562, 1309, 1899,
       2063, 1448, 1136, 1568, 2058, 1062, 2053, 2691, 819, 1538,
1780, 1011, 2030, 1377, 3243, 803, 2328, 2075, 1791,
       1898, 661, 1558, 1801, 1970, 1490, 1194, 662, 888, 1992,
1496, 808, 1551, 1910, 2900, 1843, 2809, 2363, 902]

result = solve(n := inp[0], [inp[1 + n * i:1 + n * i + n] for i in
range(n)], inp[-1])
print(list(filter(lambda x: x[1], enumerate([ans[i] != result[i]
for i in range(n - 1)]))), sep="\n")
```