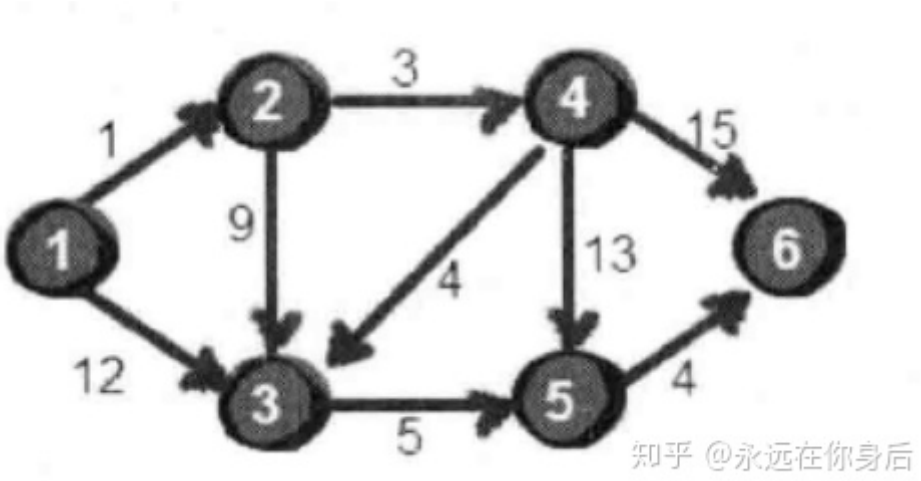
Dijkstra算法python详细实现

永远在你...
炼丹

关注他

先看图：



图片来自网络

图片来自网络

上面的图和相应的邻接矩阵，先说算法的具体过程，然后翻译成代码，算法的参数如下：

```
def startwith(start: int, mgraph: np.ndarray) -> list:
```

"""

给定一个邻接矩阵mgraph，从起始点start出发

首先将所有的点分为两类：

已经过的点：**passed**，存放已经确定最短距离的点

未经过的点：**nopass**，存放还不确定最短距离的点

显然，passed 初始化为 [start]，nopass初始化为其余的点

```
passed = [start]
nopass = [x for x in range(len(mgraph)) if x != start]
```

dis数组是要返回的结果，先初始化为 dis = mgraph[start]

```
dis = mgraph[start]     #[0   1   12   ∞   ∞   ∞]
```

现在已经过的点中只有起始点start，其他所有点都未经过，接下来要确定下一个最近的点：

遍历nopass，找出其对应在dis中最小的值，将其从nopass中放到passed中**（1）**

```
idx = nopass[0]
for i in nopass:
    if dis[i] < dis[idx]: idx = i

nopass.remove(idx)
passed.append(idx)
```

为什么能确定？因为：如果从s到t的直接路线中，P1最短，那么**不可能存在其他更短的路径（间接或直接）**，如果走任一别的直达路径P2，再经过另一条路PX到达t，可知：

P1 < P2 + PX

显然**PX不可能为负值**，所以光是P2就已经大于P1了，再加上就更远了

以下是代码运行的输出结果，算法很简洁，插入了很多print，基本上很直白了：

```
邻接矩阵如下:
[[  0   1  12 999 999 999]
 [999   0   9   3 999 999]
 [999 999   0 999   5 999]
 [999 999   4   0  13  15]
 [999 999 999 999   0   4]
 [999 999 999 999 999   0]]
现从0出发
nopass = [1, 2, 3, 4, 5]
passed = [0]
从0到各点最短路径为dis = [  0   1  12 999 999 999]

dis中未经过的点距离最短的为1, idx = 1
nopass = [2, 3, 4, 5]
passed = [0, 1]
base = dis[idx] = 1
line = mgraph[idx] = [999   0   9   3 999 999]
发现经过1点到2点的距离更短，更新dis[2] = 1 + 9
发现经过1点到3点的距离更短，更新dis[3] = 1 + 3
从0到各点最短路径为dis = [  0   1  10   4 999 999]

dis中未经过的点距离最短的为4, idx = 3
nopass = [2, 4, 5]
passed = [0, 1, 3]
base = dis[idx] = 4
line = mgraph[idx] =  [999 999   4   0  13  15]
发现经过3点到2点的距离更短，更新dis[2] = 4 + 4
发现经过3点到4点的距离更短，更新dis[4] = 4 + 13
发现经过3点到5点的距离更短，更新dis[5] = 4 + 15
从0到各点最短路径为dis = [ 0  1  8  4 17 19]

dis中未经过的点距离最短的为8, idx = 2
nopass = [4, 5]
passed = [0, 1, 3, 2]
base = dis[idx] = 8
line = mgraph[idx] =  [999 999   0 999   5 999]
发现经过2点到4点的距离更短，更新dis[4] = 8 + 5
从0到各点最短路径为dis = [ 0  1  8  4 13 19]
```

```
base = dis[idx] = 13
line = mgraph[idx] =   [999 999 999 999   0   4]
发现经过4点到5点的距离更短，更新dis[5] = 13 + 4
从0到各点最短路径为dis = [ 0  1  8  4 13 17]

dis中未经过的点距离最短的为17, idx = 5
nopass = []
passed = [0, 1, 3, 2, 4, 5]
从0到各点最短路径为dis = [ 0  1  8  4 13 17]
```

实现代码：

```
def startwith(start: int, mgraph: list) -> list:
    passed = [start]
    nopass = [x for x in range(len(mgraph)) if x != start]
    dis = mgraph[start]

    while len(nopass):
        idx = nopass[0]
        for i in nopass:
            if dis[i] < dis[idx]: idx = i

        nopass.remove(idx)
        passed.append(idx)

        for i in nopass:
            if dis[idx] + mgraph[idx][i] < dis[i]: dis[i] = dis[idx] + mgraph[idx][i]
    return dis


if __name__ == "__main__":
    inf = 10086
    mgraph = [[0, 1, 12, inf, inf, inf],
              [inf, 0, 9, 3, inf, inf],
              [inf, inf, 0, inf, 5, inf],
              [inf, inf, 4, 0, 13, 15],
              [inf, inf, inf ,inf, 0, 4],
              [inf, inf, inf, inf ,inf, 0]]

    dis = startwith(0, mgraph)
```

推荐阅读

维特比(Viterbi's algorithm)算法超不正经讲解&介绍

NeroR

Louvain | 学习三步曲

图论常识louvain介绍代码调试(python)学习笔记，供参考。 代码调试：我是找的louvain程序包（注：工程类的最好还是不要用jupyter）。 先把包安装好： pip install python-louvain再把程序...

nitri... 发表于一些实证学...

我写了一个 算法变成了

我写了一个模 变成了默写题 会了算法套路 LeetCode 上 延迟时间（中 （中等）最小

labuladong

21 条评论

切换为时间排序

写下你的评论...

mr.WU 2020-11-27

如果我不止要最短路径值，还要路径经过的节点列表，有没有简洁的写法获取到?

2

刘思黎 回复 mr.WU 01-12

每次更新最短路径的时候顺便保存父节点就行了。

赞

Theonnne 回复 mr.WU 06-28

ododo 05-22

```
def startwith(start: int, mgraph: list) -> list:
# 存储已知最小长度的节点编号 即是顺序
passed = [start]
nopass = [x for x in range(len(mgraph)) if x != start]

dis = mgraph[start]
# 创建字典 为直接与start节点相邻的节点初始化路径
dict_ = {}
for i in range(len(dis)):
if dis[i] != np.inf:
dict_[str(i)] = [start]

while len(nopass):
idx = nopass[0]
for i in nopass:
if dis[i] < dis[idx]: idx = i

nopass.remove(idx)
passed.append(idx)

for i in nopass:
if dis[idx] + mgraph[idx][i] < dis[i]:
dis[i] = dis[idx] + mgraph[idx][i]
dict_[str(i)] = dict_[str(idx)] + [idx]


return dis,dict_
```

贡献一个保存路径的版本。

1

唐刀九连 回复 ododo 05-24

没缩进，第九行的np是啥😳

赞

fang 回复 唐刀九连 07-17

SimonHU 09-28

不知道对不对，passed只是遍历过的点，如果将1->3的距离改为2，可以看到，passed为[0,1,2,3,4,5]

赞

lee 05-06

都能看懂，就是自己码不出来=-=，我是不是废了

赞

rey 04-28

感谢您留下来的精神遗产

赞

三浪 04-04

如果有相同长度的路径怎么办?

赞

甘儿 2020-12-18

怎么无论我怎么加Print,都没有输出啊

赞

大力水手 2020-09-07

代码结合图文讲，这样才是讲算法的正确方式！赞！

赞

yshan 2020-07-12

复杂度太高了

赞

会飞的鱼儿 2020-03-07

吴彦祖可以吧print位置加一下不

赞

Aurora 2019-10-11

太赞了！写的特别清楚与简洁！

1

陈怡建 2019-06-07

代码没了😭

赞

永远在你身后 (作者) 回复 陈怡建 2019-06-07

谢谢提示，已经补上了

1