

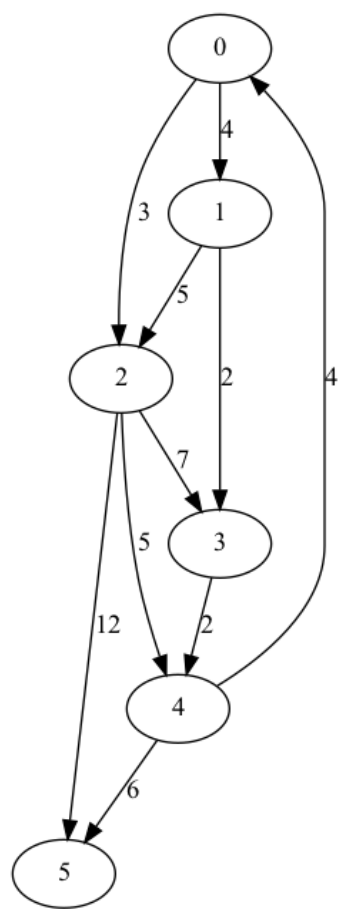
数据结构 2022秋 Lab7

TA：张皓捷 19302010021

说明

本Lab主要关于最短路算法。
问题1主要关于Floyd算法，问题2主要关于Dijkstra算法。

问题1



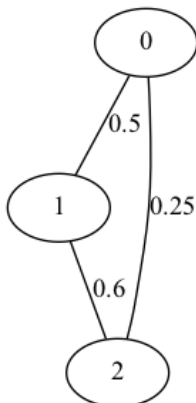
对于以上的有向图，请使用**Floyd算法**计算任意两个节点之间的最短路径。
参考PPT，写出计算过程，中间结果和最终结果都用矩阵（二维数组）表示。

问题2

现有 n 个路由器组成一个网络，路由器之间由通信链路连接。
但这些通信链路不很可靠。当在一条链路上发送一个数据包时，有一定概率出现丢包（数据包丢失）或错包（数据包内容在传输中发生了改变）等问题。在任何时候发生丢包或错包，则意味着数据包传输失败。
我们假设路由器都十分可靠，因此丢包和错包只会在通信链路上发生。

现给你一个这样的网络，并且从路由器 `start` 向路由器 `end` 传输一个数据包。请你计算在最好的传输路线下，该数据包有多大概率能传输成功。如果无法传输成功（例如 `start` 和 `end` 不联通），请返回 `0`。

例子



对于上面的网络， $0 \rightarrow 1$, $1 \rightarrow 2$, $0 \rightarrow 2$ 的传输成功率分别是 0.5, 0.6, 0.25。

最好的传输路线是 $0 \rightarrow 1 \rightarrow 2$ ，传输成功的概率是 $0.5 * 0.6 = 0.3$ 。

函数模版

```
1 double maxTransmissionProba(int n,  
2                             const std::vector<std::vector<int>>> &links,  
3                             const std::vector<double> &successProbas,  
4                             int start,  
5                             int end){  
6     //TODO  
7     return 0;  
8 }
```

`n`: 网络中路由器的个数。上述例子中为 3

`links`: 网络中的通信链路。上述例子中为 `[[0, 1], [1, 2], [0, 2]]`

`successProbas`: 每条链路的数据包传输成功率，上述例子中为 `[0.5, 0.6, 0.25]`

`start`: 数据包发送点

`end`: 数据包接受点

提示

- 问题1不需要编写代码，只需要将计算的过程和结果提交。手写拍照或电子版均可。
- 问题2的情景和PPT上Dijkstra算法的情景略有不同。如何修改算法，使之能适应问题2的情景？
- 对于问题2，需要修改的文件有 `src/MaxTransmissionProbaProblem.cpp`。
- 对于问题2，你可以在 `tests/MaxTransmissionProbaProblemTest.cpp` 里自己编写一些额外的测试用例，但不要将其提交。

截止日期

2022年11月20日 周日 23:59

提交

对于问题1，提交计算的结果和过程。

对于问题2，提交src/MaxTransmissionProbaProblem.cpp。

将上述文件打包后上传到elearning。文件命名为 学号-姓名-Lab7.zip，例如21302019999-张三-Lab7.zip。