蛋白质网络中最短距离求解程序手册

项目简介

蛋白质网络中最短距离求解程序主要通过矩阵模拟了一个蛋白质网络。计算后，矩阵中存储的元素M[i][j]就是蛋白质i和蛋白质j的最短距离。该项目中内置了多个算法，同时部分算法中含有多种数据输入格式。使用者可以自行选择算法和数据输入格式进行计算，之后程序就将输出所有蛋白质之间的最短路径长度。

使用方法

首先，执行程序后，用户首先会看到关于该程序的一些描述，之后就会看到第一个选择提示（用于选择用户希望的算法）。如下图所示：

```
**********PROTEIN NETWORK'S SHORTEST PATH CALCULATING PROGRAM**********

                                          ---------------------BY GROUP 7

This program is created to solve the shortest distance in the proteins network

Firstly,please enter your choice of the algorithm you want
d represents the Dijkstra algorithm.
b represents the Bellman-Ford algorithm.
f represents the Flolyd  algorithm.
q represents the exit operation
```

其中，d/b/f/q分别代表着不同的操作和算法。

而对于不同算法，其内置选择界面是不同的。对于Dijkstra algorithm和Flolyd algorithm，选择后首先会要求用户输入网络中的元素个数：

```
At the beginning,you need to enter the number of proteins in this network:
```

之后会要求用户输入数据的传入方式（1是交互式，2是文件解析式）：

```
Now choose the way of data entering:
1 means you want to enter the data by hand;2 means you want to enter the path of the file.
```

对于交互式，用户需要不断输入指定两蛋白质之间的距离**（99指不相邻，1~98指相邻）**

```
Then you need to enter the distace between every two proteins:

Be aware that if two proteins are not neighbors, then set the distace between them 99)

Please enter the distance between protein 0 and protein 1:6
Please enter the distance between protein 0 and protein 2:
```

而对于文件解析式输入，用户需要给出文件名（同时确保文件和程序在统一目录之下）：

```
Please enter the name of the file:
```

之后程序就会计算出结果：

```
The shortest distance between protein 5 and protein 0 is 2
The shortest distance between protein 5 and protein 1 is 2
The shortest distance between protein 5 and protein 2 is 3
The shortest distance between protein 5 and protein 3 is 5
The shortest distance between protein 5 and protein 4 is 3
The shortest distance between protein 5 and protein 5 is 0
```

而对于 Bellman-Ford algorithm，选择进入界面后，首先会要求输入元素个数、边个数、源节点：

```
Enter number of vertices in graph
5
Enter number of edges in graph
6
Enter your source vertex number
0
```

之后就会要求输入每一条边的相关信息（起点、终点、长度）：

```
Enter edge 0 properties Source, destination, weight respectively
0
2
1

Enter edge 1 properties Source, destination, weight respectively
0
1
3
```

输入完毕之后就会得到计算结果：

```
Vertex          Distance from Source 2
 0               1
 1               4
 2               0
 3               2
 4               1

Vertex          Distance from Source 3
 0               3
 1               4
 2               2
 3               0
 4               3

Vertex          Distance from Source 4
 0               2
 1               5
 2               1
 3               3
 4               0
```