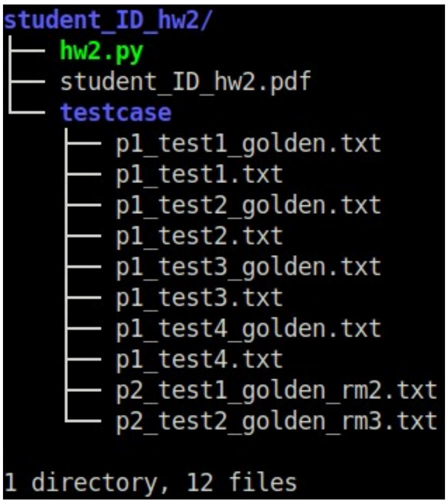
Homework2

Due date: 2022/12/8 21:00

Homework Policy: (Read before you start to work)

1. 作業請勿抄襲。疑似抄襲，助教會請你說明。如果認定抄襲，作業以零分計算。
2. 如果作業上遇到困難可以討論（並非合作！），程式碼的部分請自己完成，並且在程式第一行註明討論同學的姓名及學號。
3. 程式作業請於期限內至 NTU Cool 作業區上傳，格式為 zip 檔，解壓縮後應恰為一個以學號 (英文字母小寫)_hw2 為名的資料夾，資料夾內有一個子資料夾 (testcase)、hw2.py 與一個以自己學號為名的 pdf 檔 (student_ID_hw2.pdf)，如下所示：



```
student_ID_hw2/
├── hw2.py
├── student_ID_hw2.pdf
└── testcase
    ├── p1_test1_golden.txt
    ├── p1_test1.txt
    ├── p1_test2_golden.txt
    ├── p1_test2.txt
    ├── p1_test3_golden.txt
    ├── p1_test3.txt
    ├── p1_test4_golden.txt
    ├── p1_test4.txt
    ├── p2_test1_golden_rm2.txt
    └── p2_test2_golden_rm3.txt

1 directory, 12 files
```

Figure 1: Folder Structure.

4. 逾期繳交一天，分數 ×$\frac{2}{3}$；超過一天未滿兩天，分數 ×$\frac{1}{3}$；超過兩天則不予計分，請務必盡早開始，並努力完成。
5. 若助教無法編譯你的程式，以 0 分計算。繳交檔案格式錯誤以及包含多餘檔案會扣總分 50 分，請按照格式規則。
6. 如有任何問題歡迎來信，請同時寄信給兩位助教。
   範例: [2022ICN] HW2
   信箱: 林致佑 d09921030@ntu.edu.tw、陳嘉宏 r11921051@ntu.edu.tw

[**Problem description**]

Suppose there is a network with arbitrary number of routers. These routers are connected arbitrarily. The network topology can be represented as a matrix, indicating the cost of links between any directly connected routers. Initially, each router only has its link information.

In this homework, you are going to develop a simulator adopting Link-State Routing Protocol. First, the simulator reads a input file containing the information of network topology. A network topology graph can be determined. Then, the simulator applies Dijkstra's algorithm (or other algorithms) to find the shortest path between a source and a destination. Finally, the simulator generates a routing table for each router. In summary, the simulator should include two functions:

- Compute the least cost path (which we can refer it as an optimal path) between any two routers.
- Generate a routing table for each router in a given network.

Part I: Link-State Routing Protocol – 70%

Consider a network topology in Figure 2. R1 to R6 are routers and there is a link cost of a link between two routers. (E.g., the link cost between R1 and R2 is 3.)

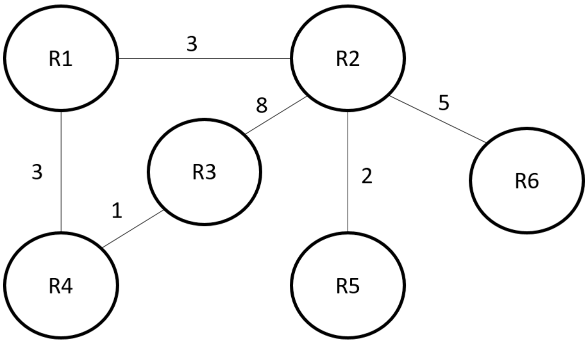


Figure 2: A network topology with cost on links

We define 'indirect connection' for a node can not reach to another node via one hop in an undirected graph. For example, R1 and R3 is indirectly connected, and R2 and R4 is indirectly connected.

In Figure 2, we assume the cost between a router and itself is 0, and the indirect connection cost between two routers is -1. Value -1 has the same meaning of the infinite value (∞) in link state routing algorithm mentioned in the class. The network topology in Figure 2 can be represented as a matrix shown in Table 1. Table 1 will be given as an input file. Please notice that your simulator is required to handle a network with arbitrary number of routers and costs.

Table 1: The network topology matrix in Figure 2

	R1	R2	R3	R4	R5	R6
R1	0	3	-1	3	-1	-1
R2	3	0	8	-1	2	5
R3	-1	8	0	1	-1	-1
R4	3	-1	1	0	-1	-1
R5	-1	2	-1	-1	0	-1
R6	-1	5	-1	-1	-1	0

- **Sample input**
  Take the network topology in Figure 2 for example, the format of input file is a matrix representation as below. Row 1 represents the number of routers. Row 2 represents the direct link cost from R1 other routers, and row 3 represents the direct link cost from R2 to other routers, etc. Your simulator should be able to handle a input file with this format.

  ```
  6
  0 3 -1 3 -1 -1
  3 0 8 -1 2 5
  -1 8 0 1 -1 -1
  3 -1 1 0 -1 -1
  -1 2 -1 -1 0 -1
  -1 5 -1 -1 -1 0
  ```

- **Sample output**
  The output file should contain the routing table of all routers, including the cost (column 1) of the least cost path from a source router to destination router, and the next hop (column 2) information.
  Your simulator should output a file [input file name]_GenTable.txt under the same directory of the input file. (E.g., if the input file is testcase/p1_test1.txt, the output file should be testcase/p1_test1_GenTable.txt.)

  ```
  Routing table of router 1:
  0 1
  3 2
  4 4
  3 4
  5 2
  8 2
  Routing table of router 2:
  3 1
  0 2
  7 1
  6 1
  2 5
  5 6
  ......
  ```

- **Report**
  Detailed description of your Part I program. (You may attach some important part of your code.)

- **Grading Policy**
  56 points for the programming part. There will be four public testcase and three private testcase. Each testcase you will get 8 points.
  14 points for the report part.

Part II: Router removed – 30%
Once the routing table has been established in Part I, the routing table remains unchanged. However, if one of the routers is removed, the whole network topology will be change. Hence, in the part, your simulator should be able to handle such change and reconstruct a new routing table.
The routing table of the removed router is omitted in the output file. If removing a router makes one router be not able to reach another routers, then set both the least cost path and the next hop in the routing table to -1. The router number remains unchanged before and after removing.
Your simulator should output a file [input file name]_RmRouter[Removed Router Number].txt under the same directory of the input file. (E.g., if the input file is testcase/p1_test1.txt, after removeing R2, the output file should be testcase/p1_test1_RmRouter2.txt.)

Example: After constructing routing tables from Figure 2, R2 is removed:
Routing table of router 1:
0 1
-1 -1
4 4
3 4
-1 -1
-1 -1
Routing table of router 3:
4 4
-1 -1
0 3
1 4
-1 -1
-1 -1
……

- **Report**
  Detailed description of your Part II program. (You may also attach some important part of your code.)

- **Grading Policy**
  24 points for the programming part. There will be two public testcase and two private testcase. Each testcase you will get 6 points.
  6 points for the report part.

[**Program Rules**]

- Please use python to implement the simulator. The python version should be 3.7 or higher.
- Routers are numbered from 1.
- You do not need to do error handling. For example, check whether the number of the removed router invalid.
- The output file should be the same as the corresponding golden file in public testcase. Otherwise, you will get no point.
- For Part I: Please run the program by python hw2.py p1_test*.txt .
  For example, python hw2.py p1_test1.txt
  We provide four public testcases and golden files. (testcase/p1_*)
- For Part II: Please run the program by python hw2.py p1_test*.txt [Removed Router Number] .
  For example, python hw2.py p1_test1.txt 2 indicates that after constructing routing table from Part I, we want to remove R2 and reconstruct routing tables.
  We provide two golden files (testcase/p2_*).
  p2_test1_golden_rm2.txt: After constructing routing table from p1_test1.txt, remove router 2.
  p2_test2_golden_rm3.txt: After constructing routing table from p1_test2.txt, remove router 3.