

108資料結構

第四次作業

網路拓撲問題

- 網路拓撲有很多的節點與連線，它們構成一個巨大而複雜的圖。我們想探討這個圖所擁有的一些性質。
- 作業會大量用到matrix $m*n$ 。
- 名詞介紹
 - 網路拓撲中的節點稱為node
 - 網路拓撲中的連線稱為edge
 - 網路拓撲中的封包稱為packet
 - 嚴格上三角矩陣意義為一個矩陣的對角線下半部(含對角線)全為0
 - buffer指的是一個可以存放packet的queue

網路拓撲問題-1(20%)

- 描述
 - 以一個大小是 $m*n$ ($m=n$)的嚴格上三角matrix表示拓撲圖, $1 \leq m \leq 99$ 。
 - 矩陣中所有點的值介於0~99。0代表兩node之間沒有edge, 1~99代表此edge最大可接受packet數。
 - 假設所有node皆相連且都至少有一進一出(起點node只有出, 終點node只有進), 每個node的進入的packet數皆等於出去的packet數。
- 問題
 - 起點(1)到終點(m), 求起點所能丟出的最大packet數使得這個拓撲中的任何edge不會被塞爆。
- Input
 - 一個matrix(int)
- Output
 - 換行後輸出起點可以丟出的最大packet數

網路拓撲問題-1

example

- Input

0 10 0 5 0 15 0 0

0 0 9 4 15 0 0 0

0 0 0 0 15 0 0 10

0 0 0 0 8 4 0 0

0 0 0 0 0 0 15 10

0 0 0 0 0 0 30 0

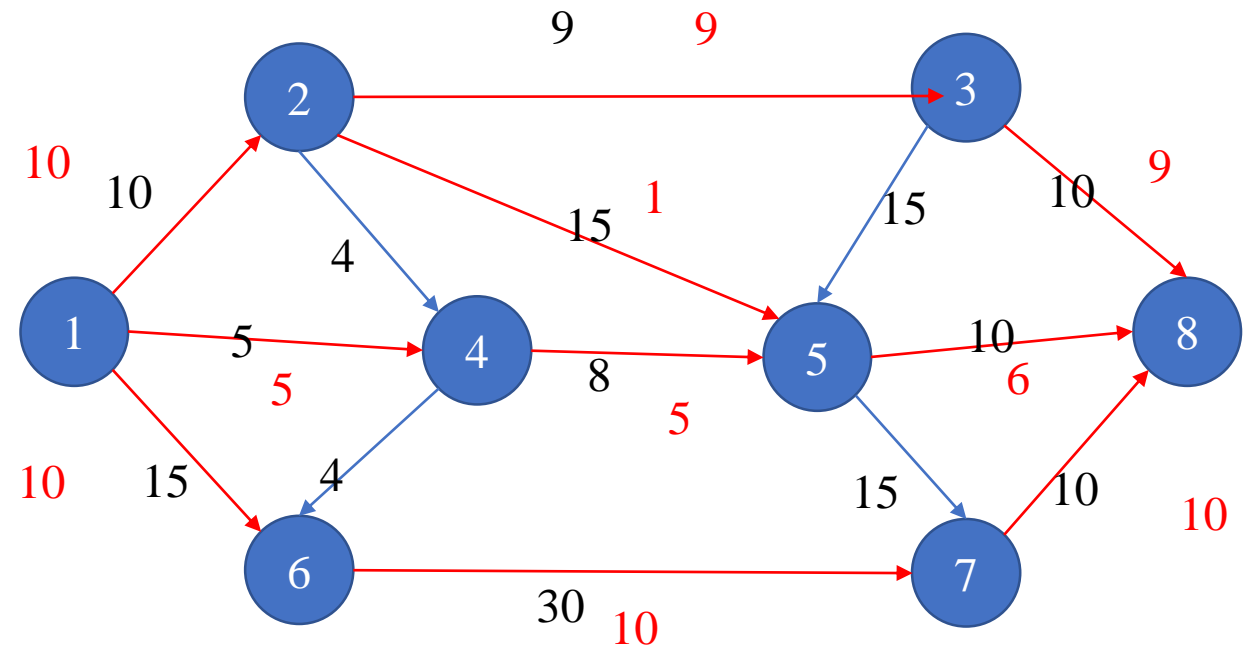
0 0 0 0 0 0 0 10

0 0 0 0 0 0 0 0

圖中紅色的線以及上面的數字代表本題目我是用了這些edge
並且在這些edge上傳輸了多少packet

- Output

25



網路拓撲問題-2(20%)

- 描述

- 以一個大小是 $m*n$ ($m=n$) 的matrix表示拓撲圖, $1 \leq m \leq 99$ 。
- 矩陣中所有點的值介於0~99。0代表兩node之間沒有edge, 1~99代表packet通過這個edge所花的時間。
- 連通判斷依據是拓撲中任意一個node到其他任意一個node可去或可回(也就是說兩者都不滿足才不算連通)。
- 最大子拓撲 (最大子拓撲指一個至少有兩個node的拓撲無法再加入任何一個node使此拓撲保持連通, 若一個拓撲只含一個node則被視為連通)。

- 問題

- 輸出這個拓撲中有幾個子拓撲

- Input

- 一個matrix(int)

- Output

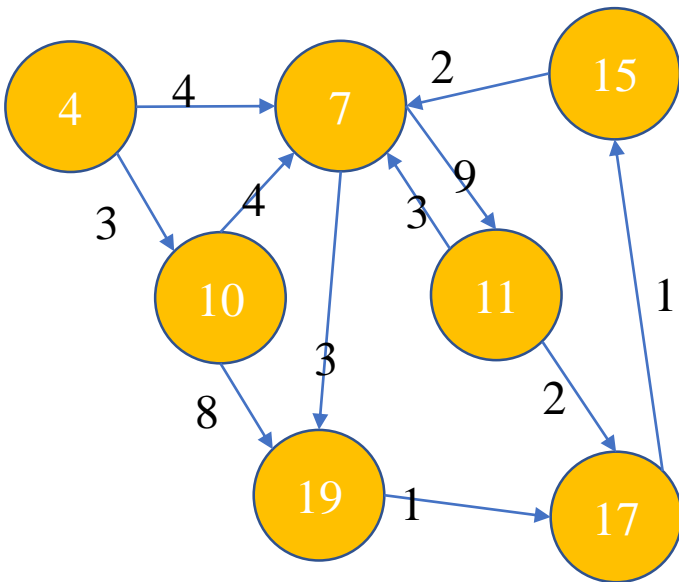
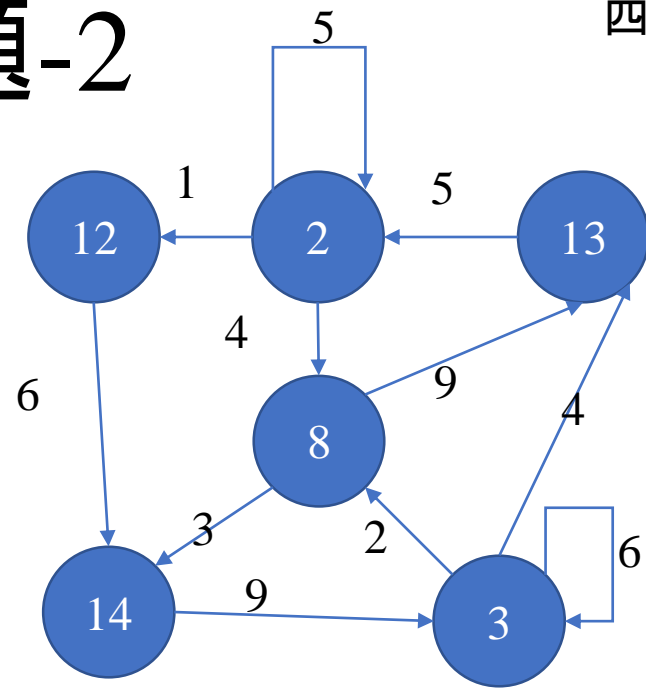
- 子拓撲個數(int)

example

- input

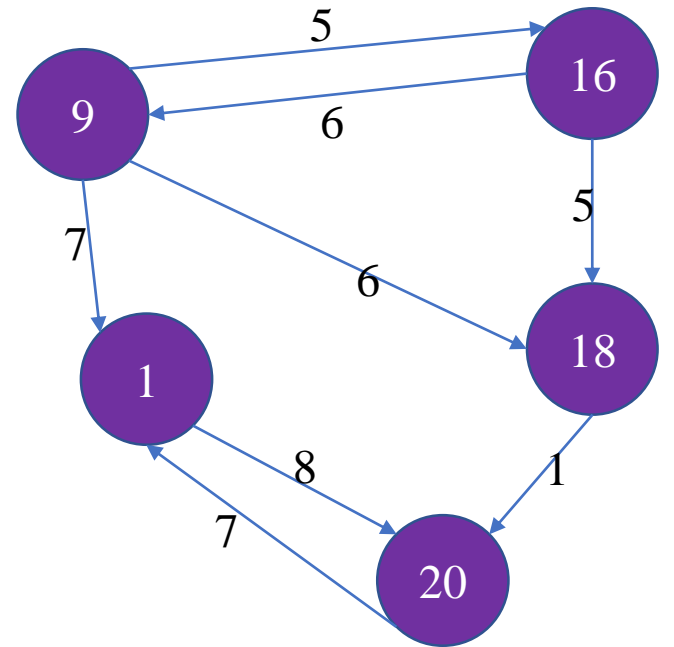
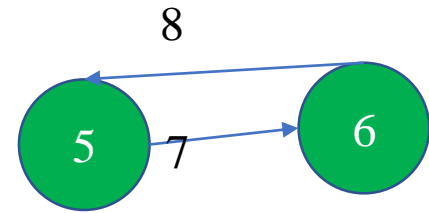
00000000000000000000000008
050000004000100000000000
006000002000004000000000
000000040030000000000000
000000700000000000000000
000080000000000000000000
00000000000900000000030
0000000000000930000000
70000000000000000060600
0000004000000000000080
0000003000000000002000
000000000000006000000
0500000000000000000000
0090000000000000000000
000000200000000200000
00000000500000000500
000000000000000100000
000000000000000000001
0000000000000000001000
7000000000000000000000

四個顏色代表四個拓撲



- Output

4



網路拓撲問題-3(20%)

- 描述

- 以一個大小是 $m \times n$ ($m=n$)的matrix表示拓撲圖, $1 \leq m \leq 99$ 。
- 矩陣中所有點的值介於-99~99。0代表兩node之間沒有edge, -99~-1, 1~99代表packet通過這個edge所需要花的時間。
- 判斷這個拓撲中有沒有負循環(負循環的意思就是反復走負邊可以使packet時間越來越短), 若有輸出0。若沒有, 找到packet到達目的地的最短時間。
- 輸入任意一對數字i(代表起點), j(代表終點)。
- 輸出packet從node i到node j的最短時間。

- 問題

- 判斷這個拓撲中有沒有負循環, 或輸出packet從node i到node j的最短時間。

- Input

- 一個matrix(int)
- 一對數字i, j中間以空格間隔(int)

- Output

- 0或最短時間(int)

網路拓撲問題-3

example

• Input

0 -2 -1 5 0

6 0 0 4 0

0 2 3 3 0

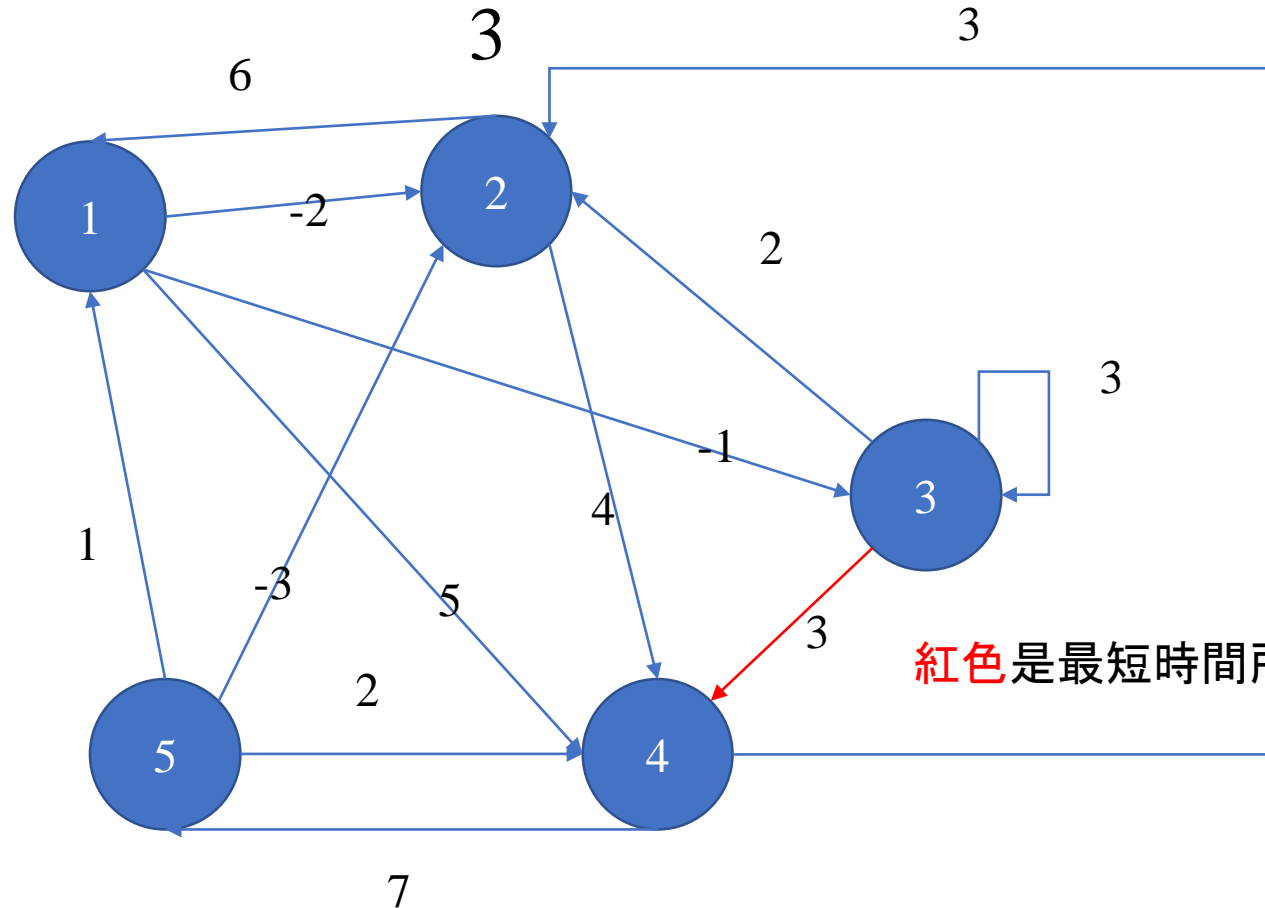
0 3 0 0 7

1 -3 0 2 0

3 4

• Output

3



紅色是最短時間所走的路徑

這一題裡面沒有負循環，但假如node2到node4的值是-5
則這一題就有負循環，要輸出0
因為node5到node2到node4到node5花-1個時間又形成一個
cycle，所以反復走這些路徑會使得所用的時間越來越少，
這題無解

網路拓撲問題-4(20%)

- 描述

- 以一個大小是 $m*n$ ($m=n$)的matrix表示拓撲圖, $1 \leq m \leq 99$ 。
- 矩陣中所有點的值介於0~99。0代表兩node之間沒有edge, 1~99代表packet通過這個edge需要花的時間。
- 假設有若干個node i ($1 \leq i \leq m$)同時送一個不定大小的packet給node j ($1 \leq j \leq m$)。
- 假設所有node都有buffer, buffer可以裝下的packet數量是自己的node號碼。**每個非起點終點的node處理自己buffer中的packet, 一次只能處理一個packet的一個單位長度。**

- 問題

- 輸出node j 的buffer中的packet

- Input

- 一個matrix(int)
- 一串數字有若干個中間以空格隔開是 i (int)(是送的node)
- 一串數字有若干個中間以空格隔開與上一行一一對應代表此node送出的packet大小(int)
- 一個數字是 j (int)(是收的node)

- Output

- j 的buffer中的packet(int)中間以空格間隔

網路拓撲問題-4

example

- Input

0 4 11

6 0 2

3 0 0

1 2 3

5 3 1

1

- Output

3

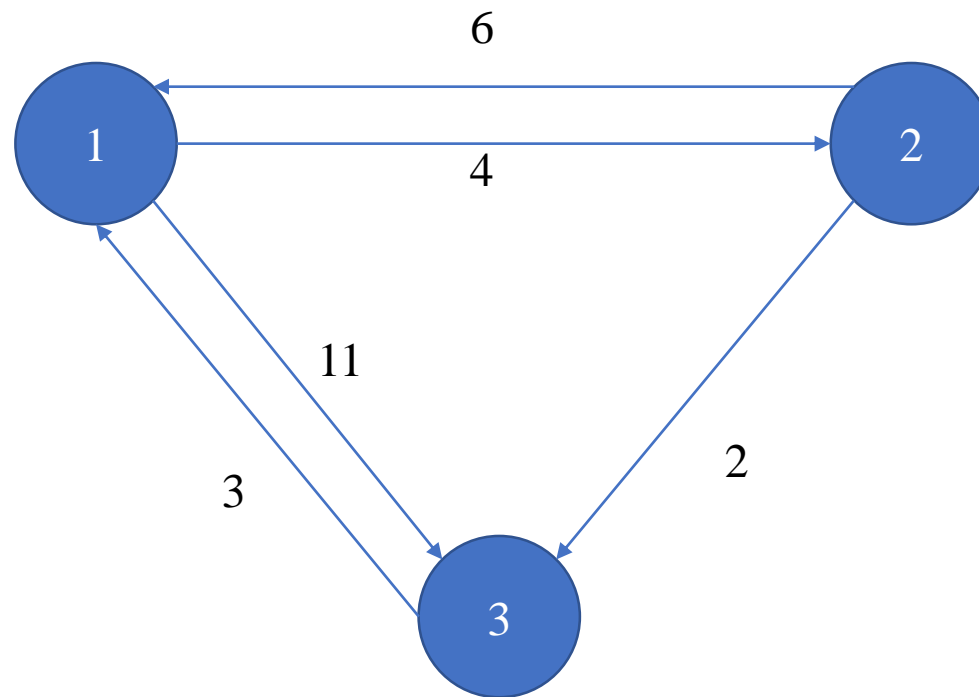
node1的buffer大小是1(只能裝1個packet), node2大小是2, node3大小是3
node選擇使packet可以最快到達終點的edge, 丟它的packet

node1, 2, 3分別送一個packet給node1

而node1送出一個大小為5的packet, 也就是說這個packet無論路過哪個node
都要被處理5個時間單位. 若node2的buffer裡面已經有一個長度是3的packet,
一個長度是5的packet, 則node2要處理3個時間長度才能送出長度3的packet,
再花5個時間才能送出長度5的packet.

若這兩個packets都還沒被處理完, 任何再來的packet都會被丟,
因為node2的大小就只有2而已.

(上面文字的顏色與example中的顏色對應)



網路拓撲問題-5(20%)

- 描述

- 以一個大小是 $m*n$ ($m=n$) 的matrix表示, $1 \leq m \leq 99$ 。
- 矩陣中所有點的值介於0~99。0代表兩node之間沒有edge, 1~99代表packet通過這個edge所需要花的時間。
- 連通判斷依據是拓撲中任意一個node到其他任意一個node可去且可回(也就是說若有一個不滿足就不算連通)。
- 找到這個拓撲中有幾個最大子拓撲(最大子拓撲指一個至少有兩個node的拓撲無法再加入任何一個node使此拓撲保持連通, 若一個拓撲只含一個node則被視為連通)。
- 假設找到的子拓撲數是 i , 找到 j 個node並從拓撲中刪除這些node使得整個拓撲有 $(i+j)$ 個子拓撲, 輸出你找到的那些node j 。若拓撲無法滿足則輸出0。

- 問題

- 輸出找到的node j 若有多個則中間以空格間隔, 或0

- Input

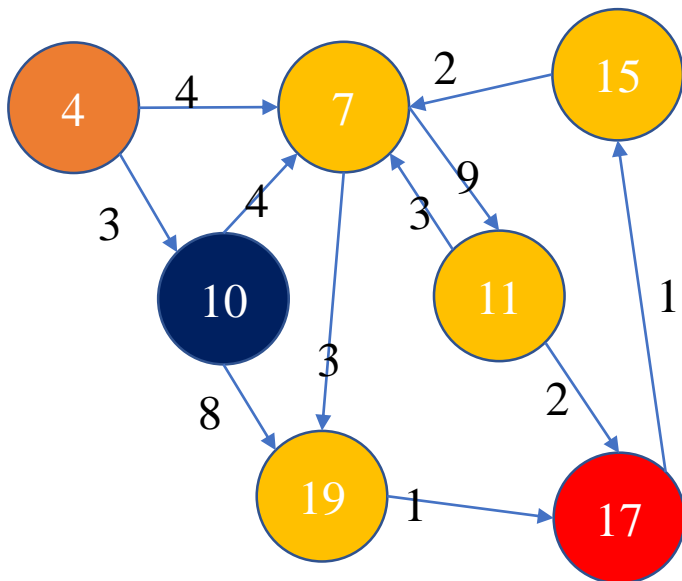
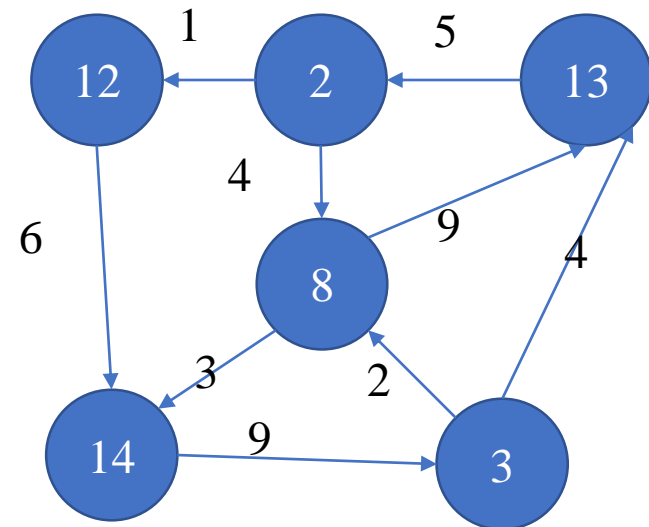
- 一個matrix(int)

- Output

- 輸出找到的node j (若有多個則中間以空格間隔), 或0(int)

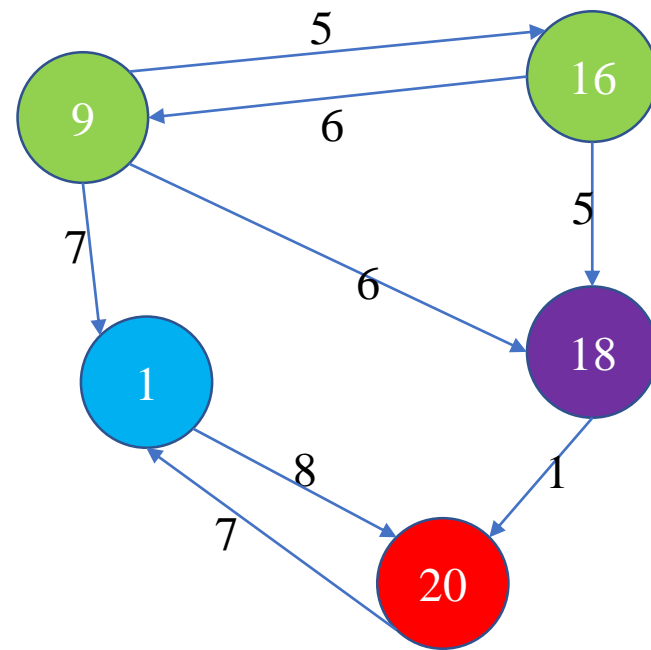
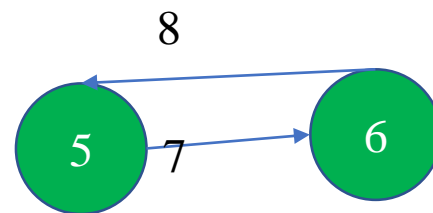
example

00000000000000000000000008
000000004000100000000000
000000002000040000000000
000000040030000000000000
000007000000000000000000
000080000000000000000000
00000000000900000000030
0000000000000930000000
70000000000000000060600
00000040000000000000080
0000003000000000002000
0000000000000060000000
050000000000000000000000
009000000000000000000000
0000002000000002000000
000000005000000000500
0000000000000000100000
0000000000000000000001
00000000000000000001000
700000000000000000000000



原本node17是橘色, node20是藍色

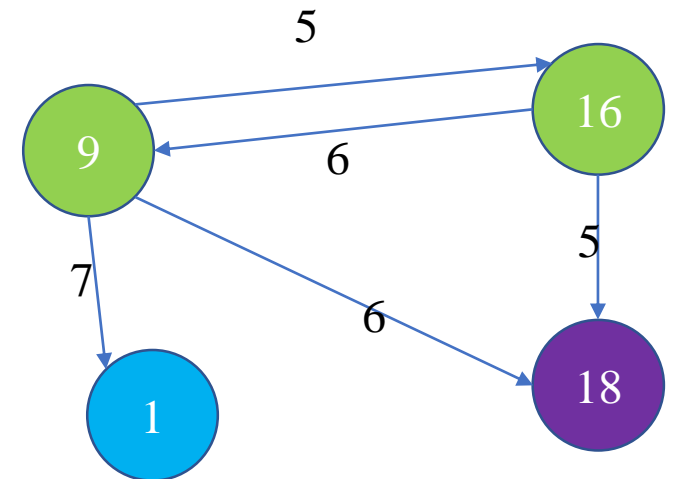
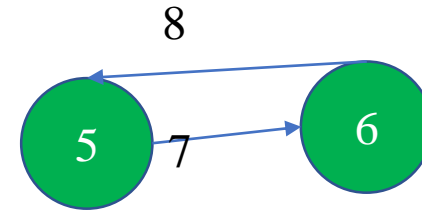
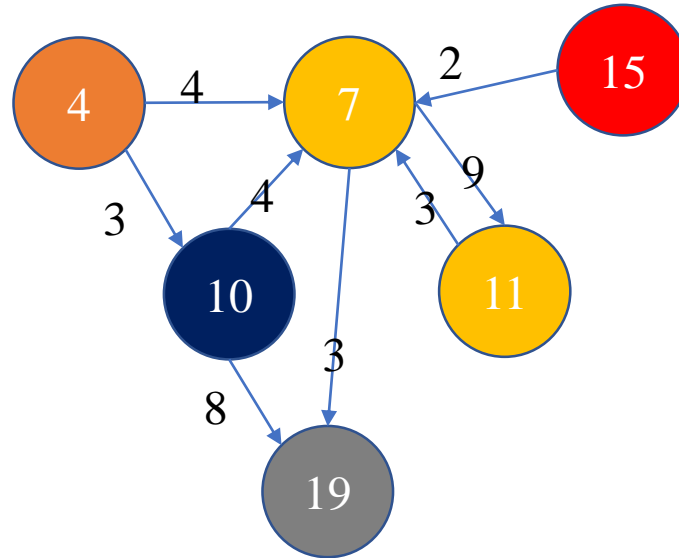
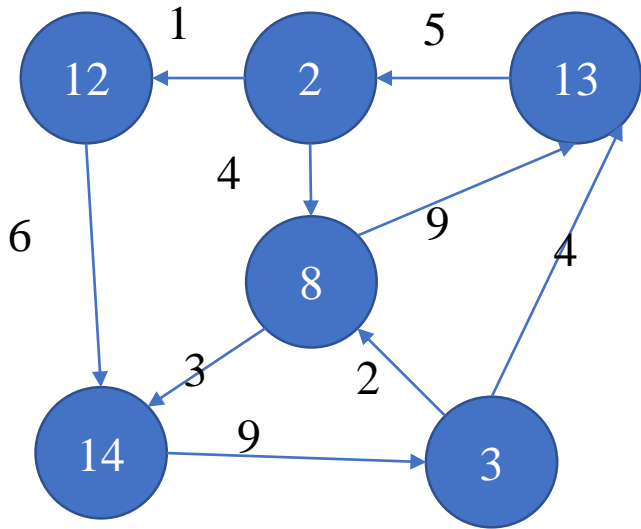
- Output
17 20



網路拓撲問題-5

example

上一頁的兩個node17, 20。形成了10個最大子拓撲



繳交方式

- Accepted C/C++
- Deadline : 2019/12/30(一) 中午12:00。不接受遲交若有正當理由可提出。
- Input必須是read file!(input1.txt , input2.txt , ...) **不可以用絕對路徑**。
- Output也必須是read file(output1.txt , output2.txt , ...)output檔不用繳交
- 檔名 : main.c若有很多檔案請標清楚題號比如, main1.c, main2.c, ...
- 繳交方式:將你寫的所有檔案(包含main檔和input檔)放在一起壓縮, 以i-learning方式繳交, 檔名為“學號+姓名+homework4”。
- 禁止抄襲, 可以的話打上註解方便閱讀
- 至少要跑出我給的測資出來, 批改的時候還會測其它測資!
- 我們會以Dev-C++來做批改