

在本題中，我們將請你實做上課教的 augmenting path algorithm，不過你不用完成整個實做，只要照規則找出一條 augmenting path 就好了。本題要求你使用上課教的 BFS（breath-first search，廣度優先）方法，在每一階層中考慮由  $s$  通往目標節點之子路徑的最大運量，由大而小放進 search queue（搜尋佇列）中，若平手則將最終節點編號較小的先放入 queue。請在找到第一條可以由  $s$  運送到  $t$  的路徑時結束，並且輸出該路徑。

讓我們以圖 1 為例。根據上課教的 BFS 搜尋法，我們將執行以下步驟：

1. 首先我們要在 queue 中放入  $(s, a; 9)$ 、 $(s, c; 7)$  與  $(s, e; 5)$  三個路徑，由於  $9 > 7 > 5$ ，我們將按照上面的順序放入 queue。完成此步驟後，queue 中含有的路徑依序為

$$\{(s, a; 9), (s, c; 7), (s, e; 5)\}。$$

2. (a) 接著我們從 queue 中拿出  $(s, a; 9)$ ，放入從  $a$  再走一步的  $(s, a, b; 6)$ 。請注意由  $a$  往下只有一個節點  $b$ ，且由於  $u_{ab} < 9$ ，放入 queue 中的路徑的最大運量只有 6。完成此步驟後，queue 中含有的路徑依序為

$$\{(s, c; 7), (s, e; 5) \mid (s, a, b; 6)\}。$$

那一槓之前的是步驟一放入的路徑（長度為 1），之後的則是步驟二放入的路徑（長度為 2）。

- (b) 接著我們從 queue 中拿出  $(s, c; 7)$ ，放入從  $c$  再走一步的  $(s, c, b; 3)$ 。完成此步驟後，queue 中含有的路徑依序為

$$\{(s, e; 5) \mid (s, a, b; 6), (s, c, b; 3)\}。$$

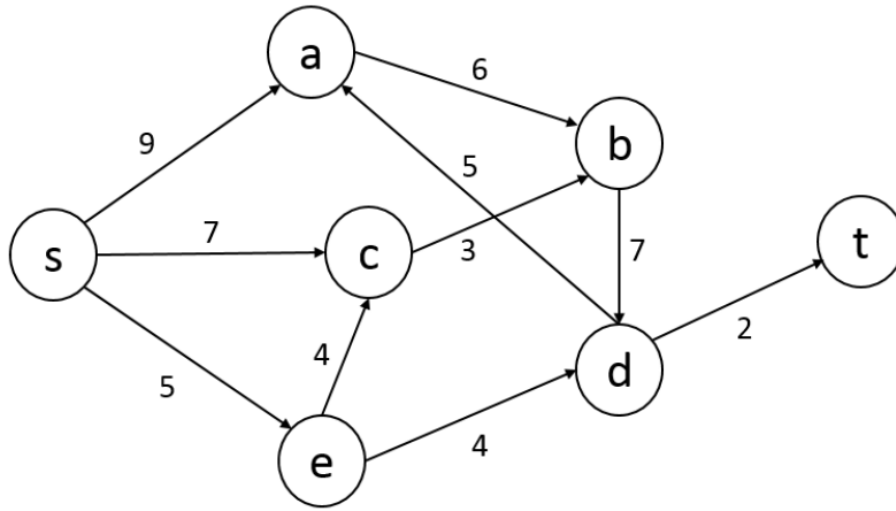


圖 1: Augmenting path 搜尋範例

- (c) 接著我們從 queue 中拿出  $(s, e; 5)$ ，放入從  $e$  再走一步的  $(s, e, c; 4)$  與  $(s, e, d; 4)$ 。由於兩個路徑的最大運量相同，我們將最終節點編號較小的先放入 queue。完成此步驟後，queue 中含有的路徑依序為

$$\{(s, a, b; 6), (s, c, b; 3), (s, e, c; 4), (s, e, d; 4)\}。$$

3. (a) 接著我們從 queue 中拿出  $(s, a, b; 6)$ ，放入從  $b$  再走一步的  $(s, a, b, d; 6)$ 。完成此步驟後，queue 中含有的路徑依序為

$$\{(s, c, b; 3), (s, e, c; 4), (s, e, d; 4) \mid (s, a, b, d; 6)\}。$$

- (b) 接著我們從 queue 中拿出  $(s, c, b; 3)$ ，放入從  $b$  再走一步的  $(s, c, b, d; 3)$ 。完成此步驟後，queue 中含有的路徑依序為

$$\{(s, e, c; 4), (s, e, d; 4) \mid (s, a, b, d; 6), (s, c, b, d; 3)\}。$$

- (c) 接著我們從 queue 中拿出  $(s, e, c; 4)$ ，放入從  $c$  再走一步的  $(s, e, c, b; 3)$ 。完成此步驟後，queue 中含有的路徑依序為

$$\{(s, e, d; 4) \mid (s, a, b, d; 6), (s, c, b, d; 3), (s, e, c, b; 3)\}。$$

- (d) 接著我們從 queue 中拿出  $(s, e, d; 4)$ ，考慮從  $d$  再走一步的  $(s, e, d, a; 4)$  與  $(s, e, d, t; 2)$ 。此時由於已經發現從  $s$  到  $t$  的一個路徑了，因此結束搜尋，並且印出該路徑。

## 輸入輸出格式

系統會提供一共 15 組測試資料，每組測試資料裝在一個檔案裡。在每個檔案中，會有  $m + 2$  列資料，其中第一列是兩個正整數  $n$  和  $m$ ，第二列是  $n$  個只含有最多十個大寫英文字母的字串  $s_1$ 、 $s_2$  直到  $s_n$ ，其中  $s_i$  代表節點  $i$  的編號（本題中我們用字串當編號），且  $s_1$  為運送起點、 $s_n$  為運送終點。第三列起的每一列含有兩個字串  $x_j$ 、 $y_j$  與一個非負整數  $u_{x_j, y_j}$ ，代表有向路段  $(x_j, y_j)$  存在於給定的 network 中，且其運量上限為  $u_{x_j, y_j}$ 。每一列中相鄰的兩個值都以一個逗點隔開。已知  $1 \leq n \leq 20$ 、

$1 \leq m \leq n(n-1)$ 、 $0 \leq u_{ij} \leq 100$ ，給定的路段資訊不包含重複的路段，但可能包含兩個節點間兩個方向（一來一往）的資訊。

讀入以上資訊後，請按照題目指定的方法找出一條 augmenting path，並依序印出該路徑由起點  $s_1$  開始要通過的節點，一直到終點  $s_n$  為止，最後印出該路徑的最大運量。任兩個印出的節點編號之間以一個逗點隔開，終點  $s_n$  與最大運量之間用一個分號隔開。如果給定的 network 中不存在 augmenting path，就輸出 -1。舉例來說，若輸入為

```
7,10
s,a,b,c,d,e,t
s,a,9
s,c,7
s,e,5
d,t,2
d,a,3
b,d,7
e,c,4
e,d,4
c,b,3
a,b,6
```

則輸出應該為

```
s,e,d,t;2
```

若輸入為

```
7,10
s,a,b,c,d,e,t
s,a,9
s,c,7
s,e,5
d,t,0
d,a,3
b,d,7
e,c,4
e,d,4
c,b,3
a,b,6
```

則輸出應該為

```
-1
```