

## Homework 3 Maximum Flow

---

### a. Pseudo Code

- **Maximum Flow (Edmonds-Karp)**

基本上與課本講義一樣

**Maximum-Flow**( $G, s, t$ )

```

1   initialize flow  $f$  to 0
2   while there exists an augmenting path  $p$  in the residual network  $G_f$ 
3       augment flow  $f$  along  $p$ 
4   return  $f$ 
```

Line 3 的部分用BFS尋找路徑，BFS如下

- **Breadth-First Search (BFS)**

BFS( $G, s$ )

```

1   for each vertex  $u \in G.V - \{s\}$ 
2        $u.color = WHITE$ 
3        $u.\pi = NIL$ 
4    $s.color = GRAY$ 
5    $s.\pi = NIL$ 
6    $Q = \emptyset$ 
7   ENQUEUE( $Q, s$ )
8   while  $Q \neq \emptyset$ 
9        $u = DEQUEUE(Q)$ 
```

```

10      for each  $v \in G.Adj[u]$ 
11          if  $v.color == WHITE$ 
12               $v.color = GRAY$ 
12               $v.\pi = u$ 
13               $ENQUEUE(Q, v)$ 
14       $u.color = BLACK$ 

```

這次作業沒有用到 distance 所以省略課本 pseudo code 中有 distance 的部分

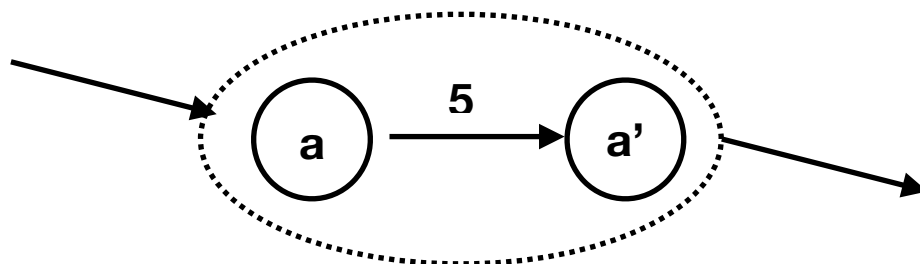
另外課本也有提到只要用兩個顏色就可以達到同樣結果所以我的 code 是使用 true false 代表顏色

## b. Design Concept

這次作業是在實作Chapter 26的東西，所以有關Maximum Flow的內容課本都有詳述，這邊暫且先忽略。以下是我實現Bonus的方法

### 1. 實現 Vertex Capacity 的方式

一個只能容納5單位的vertex A等同於下圖



相當於把一個vertex拆成兩個，然後用一條 capacity = 5 的路線連結

把原本 所有A的in-edge接到 a，out-edge接到 a'。

比較麻煩的是如何把path二維陣列中的每個點轉成上面那個樣子。

(see discussion)

## c. Discussion

## 1. 如何將原本的點拆成兩個點

擴增path二維陣列

Ex: A 到 B 有一條  $\text{cap} = 4$  的路,  $\text{cap A} = 3$ ,  $\text{cap B} = 5$ , path的矩陣為

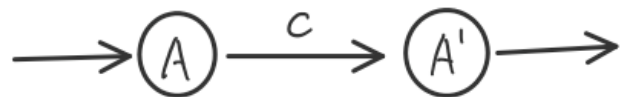
	A	B
A	0	4
B	0	0

擴增時分成兩種情況

1.  $i == j$  時這時就是把一個vertex拆成兩個, 原本的  $a_{11}$  會變成

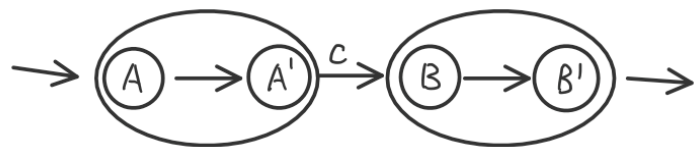
	A	A'
A	0	C
A'	0	0

C 為 vertex capacity

2.  $i \neq j$  時

	B	B'
A	0	0
A'	C	0

C 為 edge AB 的capacity



擴增完變成

	a	a'	b	b'
a	0	3	0	0
a'	0	0	4	0
b	0	0	0	5
b'	0	0	0	0

如此一來就會生成等效的新圖

剩下依照原本求Max Flow的方法即可