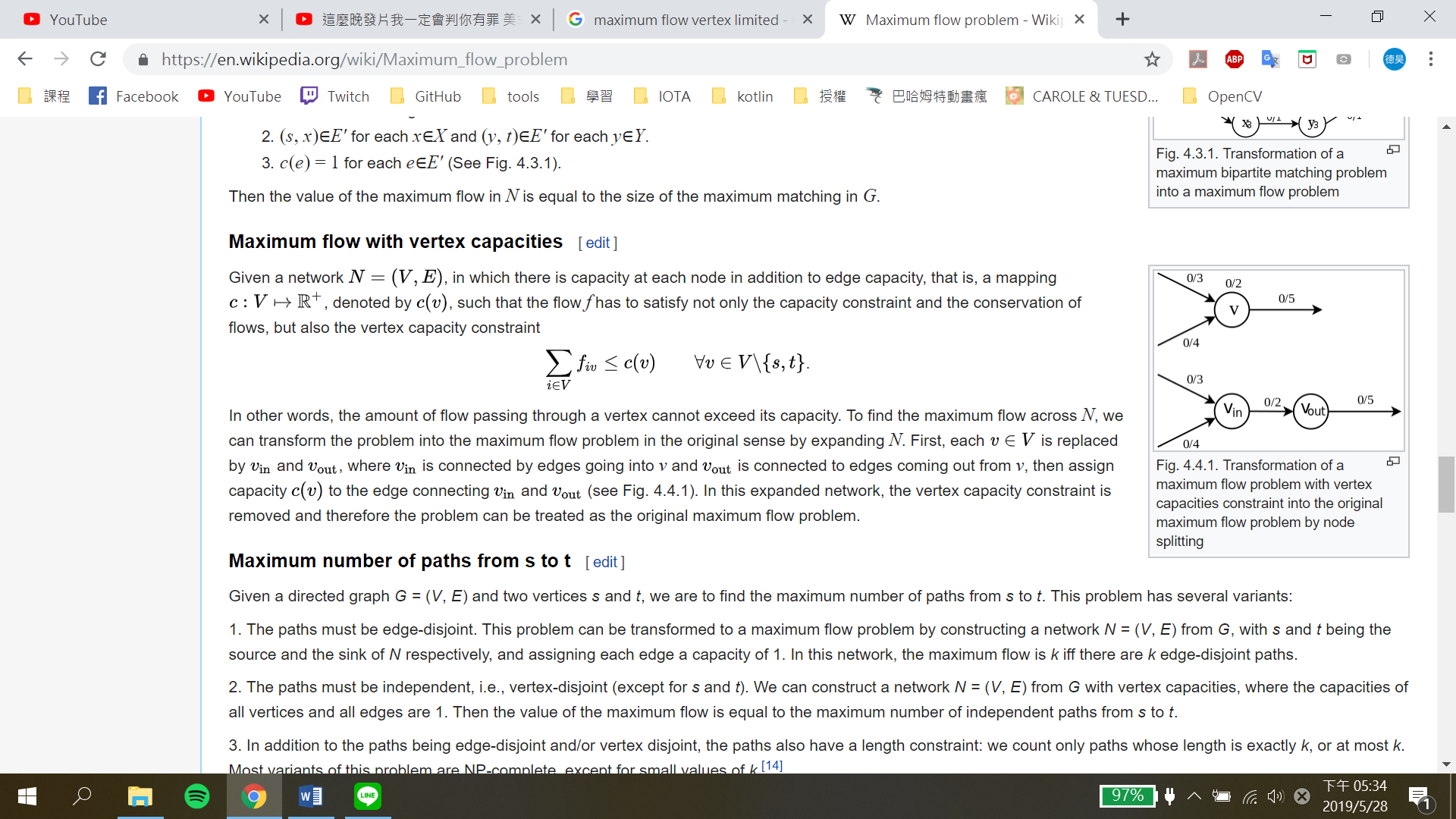
1. Suppose that, in addition to edge capacities, a flow network has vertex capacities. That is each vertex v has a limit l(v) on how much flow can pass through v. Show how to transform a flow network G = (V, E) with vertex capacities into an equivalent flow network G’= (V’, E’) without vertex capacities, such that a maximum flow in G ’ has the same value as a maximum flow in G. How many vertices and edges does G’ have?

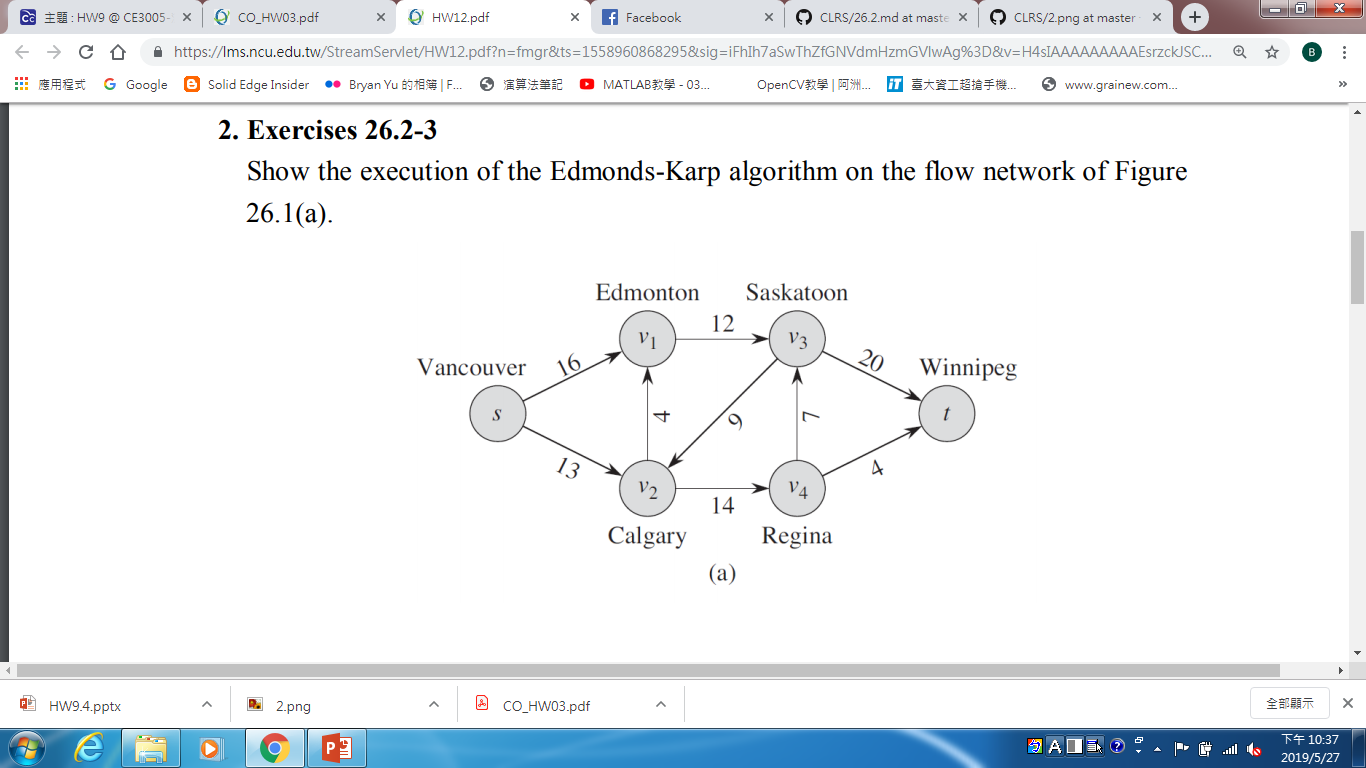
G’= (V’, E’)

1. 將v分為v1, v2 edge(v1,v2)=l(v)
2. 兩點之間，edge(u2, v1)=capability(v, u)



|V’|=2|V|，一個點變兩個

|E’|=|E|+|V|，原本的邊加上每個點分裂之間的邊

2. Show the execution of the Edmonds-Karp algorithm on the flow network of Figure 26.1(a).

S1 : Star with zero flow everywhere

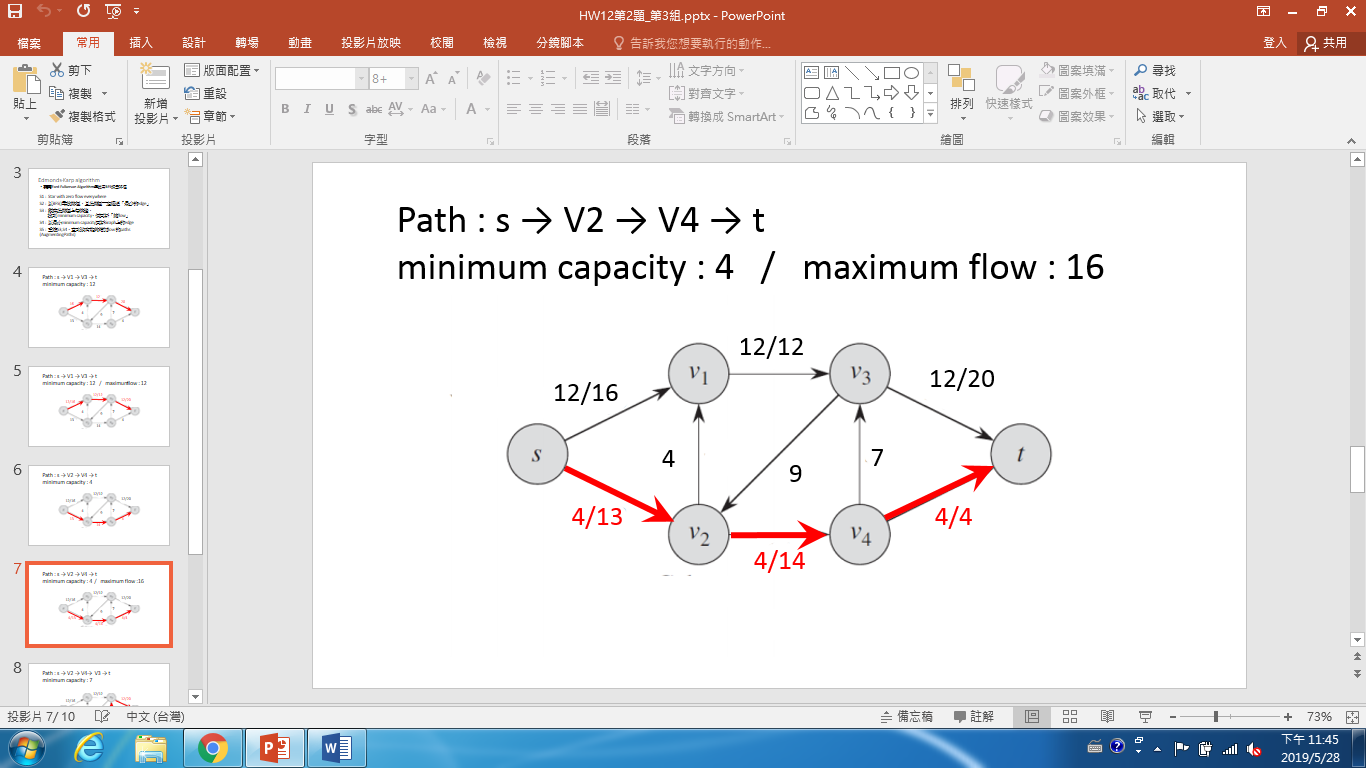
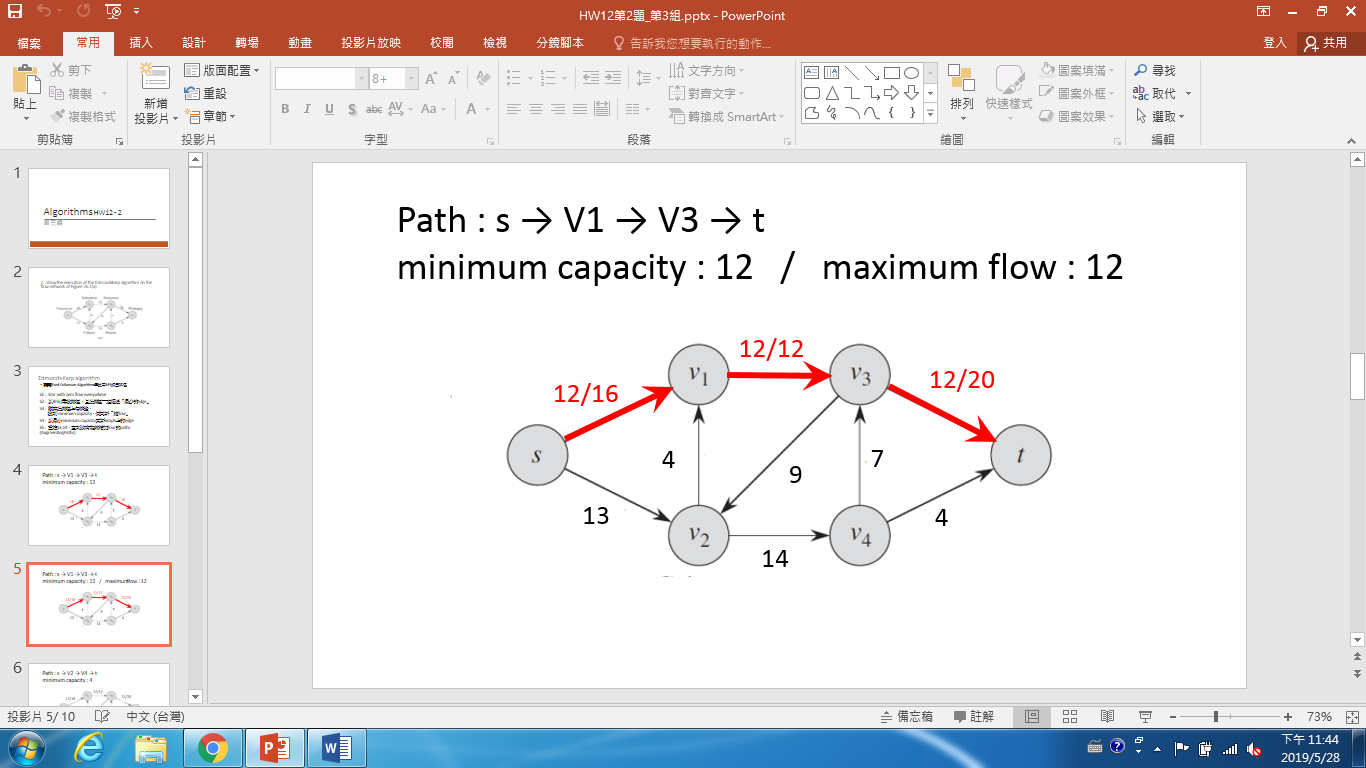
S2 : 以BFS()尋找路徑，且此路徑一定經過「最少的edge」

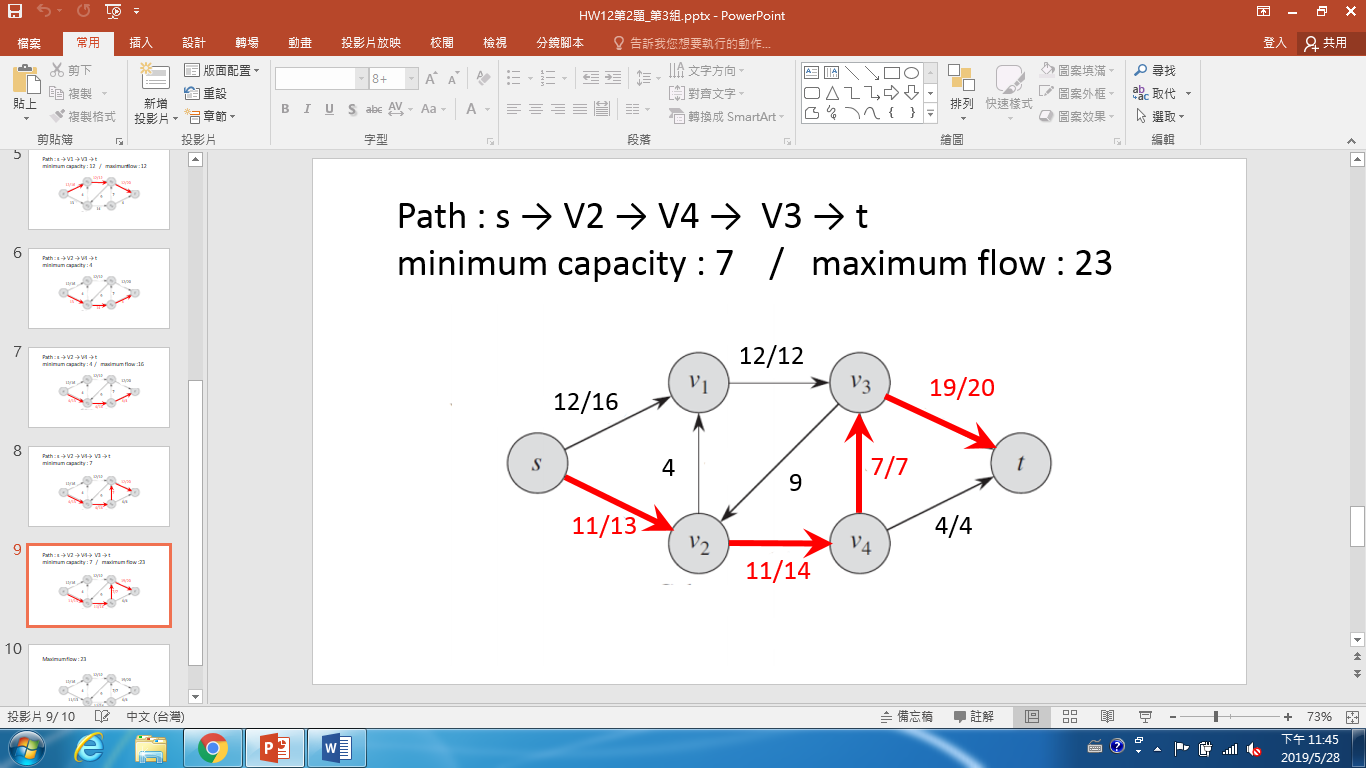
S3 : 觀察此路徑上每條邊，

找到 minimum capacity，便更新「總flow」

S4 : 以最小minimum capacity 更新Graph上的edge

S5 : 重複S3,S4，直到沒有能夠增加flow 的paths (Augmenting Paths)





3. The edge connectivity of an undirected graph is the minimum number k of edges that must be removed to disconnect the graph. For example, the edge connectivity of a tree is 1, and the edge connectivity of a cyclic chain of vertices is 2. Show how to determine the edge connectivity of an undirected graph G = (V, E) by running a maximum-flow algorithm on at most |V| flow networks, each having O(V) vertices and O(E) edges. (Find the value of edge connectivity.)

此題在問的是如何找edge connectivity，例如對tree來說edge connectivity為1，對一個環來說為2。

1. 將無相圖G的每個edge capacity設為1
2. 找任意一點當作s任意另一點當作t，利用maximum flow algorithm找這兩點的min cut
3. 對所有任意兩點組合做步驟1與2並找到最小的min cut 就是edge connectivity



例如上圖min cut皆為2，所以edge connectivity為1。

時間複雜度分析：對所有任意兩點找maximum flow，因此做C(n,2)=n\*(n-1)/2次，maximum flow 對固定兩點為O(|V|\*|E|^2)，因此為O(|V|^3\*|E|^2)。

4. Suppose that you wish to find, among all minimum cuts in a flow network G with integral capacities, one that contains the smallest number of edges. Show how to modify the capacities of G to create a new flow network G′ in which any minimum cut in G′ is a minimum cut with the smallest number of edges in G.

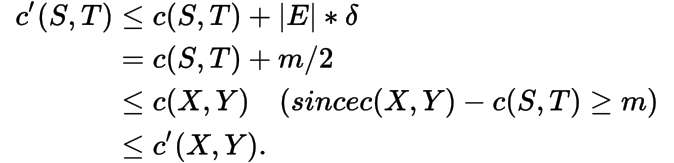
ANS:

Let (*S*,*T*) and (*X*,*Y*) be two cuts in *G* (and *G*′). Let *c*′ be the capacity function of *G*′. One way to define *c*′ is to add a small amount *δ* to the capacity of each edge in *G*. Thus, we set

However, no matter how many more edges (*S*,*T*) has than (*X*,*Y*), we still need to have *c*′(*S*,*T*)<*c*′(*X*,*Y*).

Then we should choose the value of *δ. Let m be the minimum difference between capacities of two unequal-capacity cuts in G. Choose δ=m/(2∣E∣). For any cut (S,T), since the cut can have at most ∣E∣ edges, we can bound c′(S,T) by*

Therefore, we need to prove that . We have



Because all capacities are integral, we can choose *m*=1, obtaining *δ*=1/2∣*E*∣. To avoid dealing with fractional values, we can scale all capacities by 2∣*E*∣ to obtain

*.*

5.The vertex connectivity of an undirected graph is the minimum number k of vertices that must be removed to disconnect the graph. When we remove a vertex, we must also remove the edges incident to it. For example, the vertex connectivity of a tree is 1, and the vertex connectivity of a cyclic chain of vertices is 2. Show how to determine the vertex connectivity of an undirected graph G = (V, E) by running a maximum-flow algorithm on at most |V| flow networks, each having O(V) vertices and O(E) edges. (If the graph is complete graph, the vertex connectivity is |V| - 1.)

Vertex\_Connectivity(G)

K = INT\_MAX // vertex connectivity

1. 找到一個u ∈ V，u要連接的邊數是最少，將u設為s

如果u連接的邊數為|V|-1，則為complete graph，vc = |V|-1

1. 對所有v ∈ V-{u},做maximum-flow algorithm,

把v設為t, capacity全部設為1，flow從0開始，找到cut後，對其他(s, t之外)s無法走到的v計算有幾個flow經過，

算出mf =# of Max flows

K = min(mf , K)

6. Exercise 26.3-4 A perfect matching is a matching in which every vertex is matched. Let G = (V, E) be an undirected bipartite graph with vertex partition V = L  R, where |L| = |R|. For any X  V, define the neighborhood of X as N(X) = { y V : (x, y) E for some x X } that is, the set of vertices adjacent to some member of X. Prove Hall's theorem: there exists a perfect matching in G if and only if |A| ≤ |N(A)| for every subset A L.

Hall Theorem：

圖G存在perfect matching ⬄ |A| <= |N(A)|，A屬於一邊的子集合

Prove：

“=>”

* 反證，假設圖G存在perfect matching且|A| > |N(A)|
* 設|A| =k ，k=任意正整數
* |N(A)| <k
* 另一邊有不到k個點被連到，所以有點沒有被連到 (因為2邊點個數一樣)
* 不是perfect matching，矛盾

“<=”

反證，假設圖G不存在perfect matching且 |A| <= |N(A)|

* A中每一個點至少連向1個點
* A中每一個點都有連線
* 有perfect matching，矛盾

因為 “=>” 和 “<=” ，得證。

7. EXT 10-1 There are two extended ways used to find the augmenting path that we have mentioned in class (refers to slides p.14, Unit 10), please design an efficient algorithm with the argument of second method to find the augmenting path. Argue that your algorithm is correct and also analyze the time-complexity.

想法：

1.修改自Dijkstra's最短路徑演算法。

2.每次選擇Maxflow最大的點並做Relax。

3.到達sink後，將順向路徑上的edge的Flow加上MaxFlow，逆向則為減。

4.重複直到再也走不到sink。

Pseudo code：

max\_heap <- source

while (max\_heap != null )

u =max\_heap .pop()

if (u == sink)

reset weights

clear max\_heap

max\_heap <- source

u =max\_heap .pop()

for (each edge from u)

if weight of edge != 0

Value[v] = min(Value[u], weight of edge)

max\_heap <- v ;

複雜度O( (VlogV)(logf\*)(E))