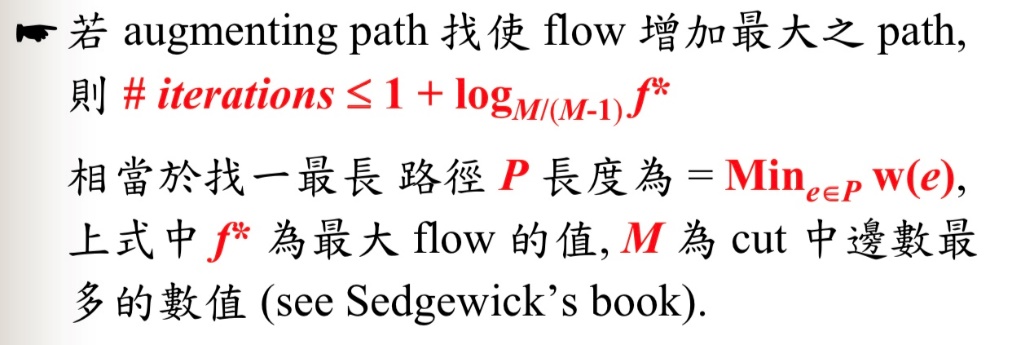
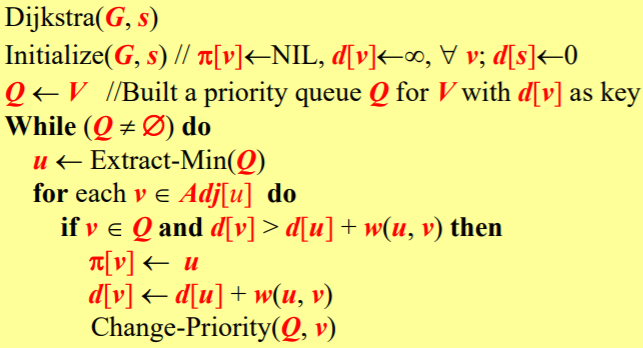
Give There are two extended ways used to find the augmenting path that we have mentioned in class (refers to slides p.14, Unit 10), please design an efficient algorithm with the argument of second method to find the augmenting path. Argue that your algorithm is correct and also analyze the time-complexity





max(d[u],w(u,v))

d[v]>max(d[u],w(u,v))

G input graph

Gf flow graph

Gr residual graph

while(true)

P = dijkstra(Gr)找最大瓶頸及路線

if P == null

Gf 找到max flow

break

for each (u,v,b) є P

if Gr(u,v).forward\_edge&&Gr(u,v).forward\_edge!=0

Gf(u,v) += b

if Gr(u,v). backward\_edge&&Gr(u,v). backward\_edge!=0

Gf(v,u) -= b

for each (u,v,f) in Gf:

(u,v,x) є Gr = (v,u,G(u,v).c-f)

(v,u,x) є Gr = (v,u,f)

如果找的到augmenting path 那瓶頸>0

那flow就會增加且dijkstra 是用 Gr找因此不會超過容量

Time Complexity

O(log F) 跑的次數( F 為max flow )

O(|E|log|V|) dijkstra(priority queue 用 heap)

Total :O(log F|E|log|V|)