Homework 5 杜業誠 E24105038 資訊114甲

1)

⇒ Assume that there exists a minimum spanning tree T' of $G(V,E)$ that does not contain the edge e with the minimum weight but contain the edge e' as the cut. edge between S and $V-S$.

↳ Consider T as the spanning tree of $G(V,E)$ that does include the edge e

↳ To construct T from T'

⇒ remove edge e'. Breaks T' into two components

⇒ Add edge e. Reconnects

⇒ So, $T = T' - \{e'\} \cup \{e\}$

↳ T is a spanning tree

⇒ $w(T) = w(T') - w(e') + w(e)$

since $\boxed{w(e) < w(e')}$

$\boxed{\therefore w(T) < w(T')}$

↳ Since $w(T) < w(T')$, this contradicts the assumption that T' is a minimum spanning tree.

∴ Therefore, every minimum spanning tree in $G(V,E)$ must contain the edge e with the minimum weight, as it is necessary to maintain the minimum total weight to form a MST. #

---

2) ① adjacency list : $\Theta(V+E)$

② adjacency matrix : $\Theta(V^2)$ #

① ↳ exploring each vertex ⇒ $\Theta(V)$

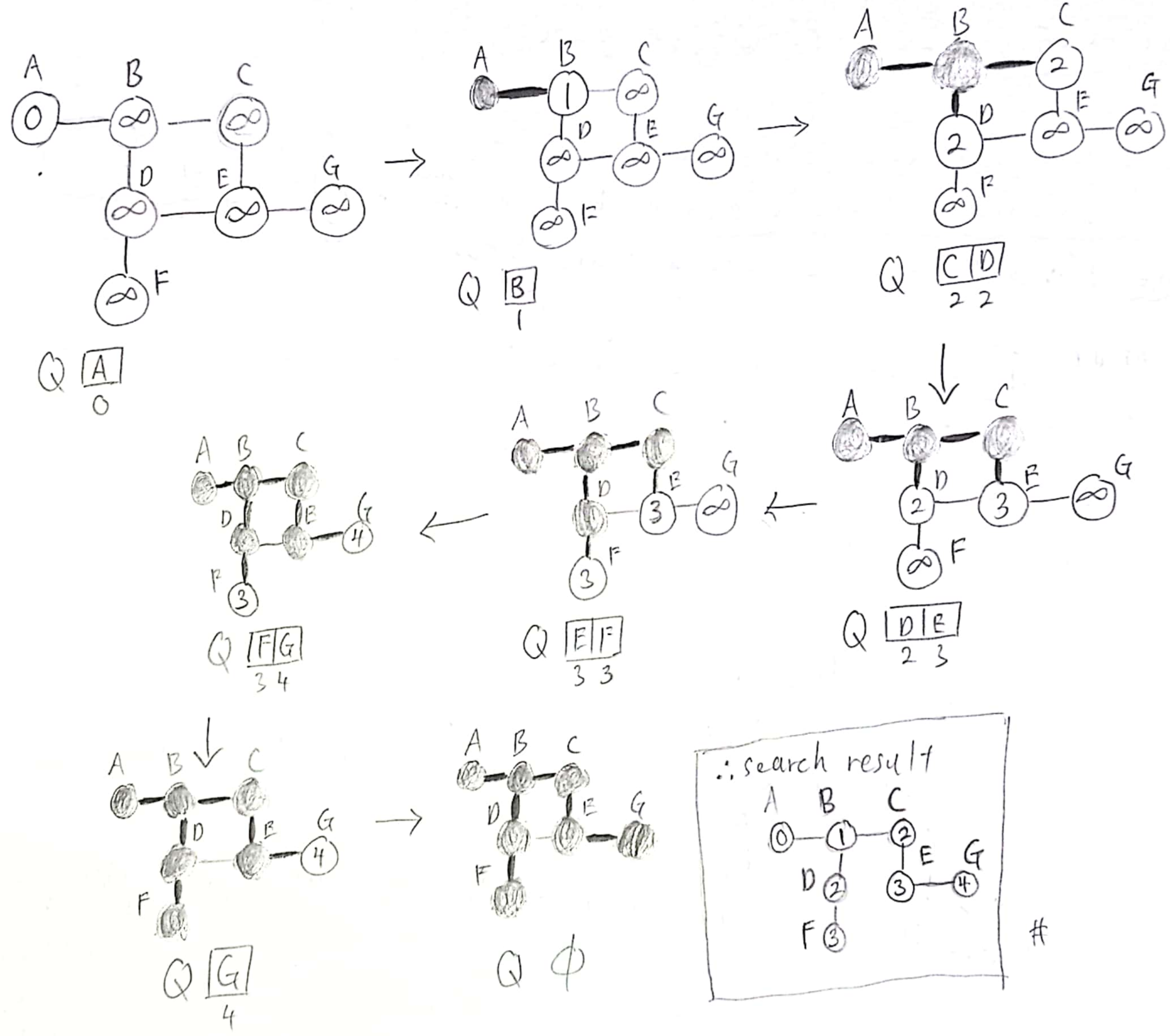
↳ exploring each adjacent edge ⇒ $\Theta(E)$ $\boxed{\therefore \Theta(V+E)}$

② ↳ exploring each vertex ⇒ $\Theta(V)$

↳ exploring each adjacent vertex : for each vertex, we need to check the presence of an edge in the adjacency matrix, which takes $\Theta(V)$ for each row. Since there are V vertices (rows). Hence, $\Theta(V^2)$.

$\boxed{\therefore \Theta(V^2)}$

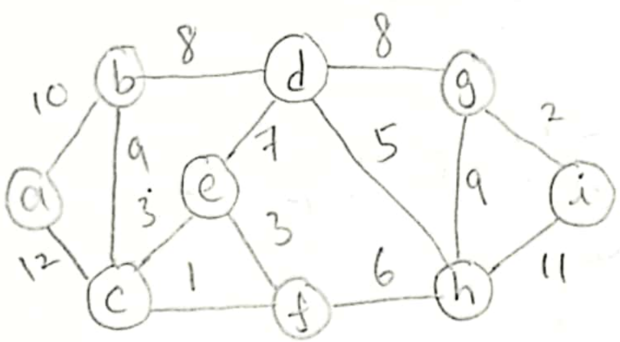
3)


A
B
C
D
E
F
G
Q A
0
Q B
1
Q C D
2 2
Q D E
2 3
Q E F
3 3
Q F G
3 4
Q G
4
Q ϕ
∴ search result
#

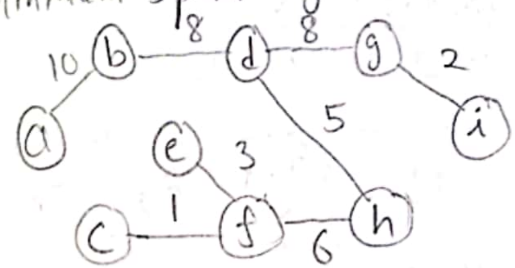
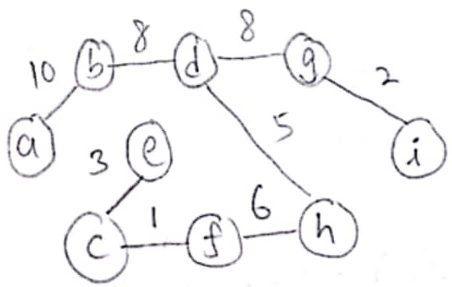
4) ⇒ Minimum spanning tree is a spanning tree whose weight is minimum over all spanning trees in an undirected graph $G=(V,E)$. #

⇒ $\boxed{\text{false}}$ #

↳ Assume in every graph, there is only one minimum spanning tree. (assumption)

⇒ example: 

↳ Minimum spanning tree is:

⇒  or 

↳ In this example, there is more than one MST. You can either choose edge (e, f) or edge (c, e).

⇒ ∴ This leads to the contradiction with the assumption states that in every graph, there is only one minimum spanning tree.

∴ Ans: $\boxed{\text{false}}$ #

5) ⇒ (f,i) = 3 : chosen

(a,b) = 5 : chosen

(e,i) = 6 : chosen

(e,h) = 7 : chosen

(b,h) = 8 : chosen

(a,d) = 8 : chosen

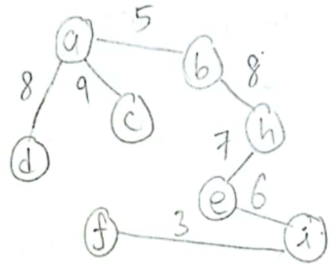
(a,c) = 9 : chosen

(c,e) = 10 : reject

(c,f) = 12 : reject

(d,f) = 14 : reject

∴ Minimum Cost Spanning Tree is:

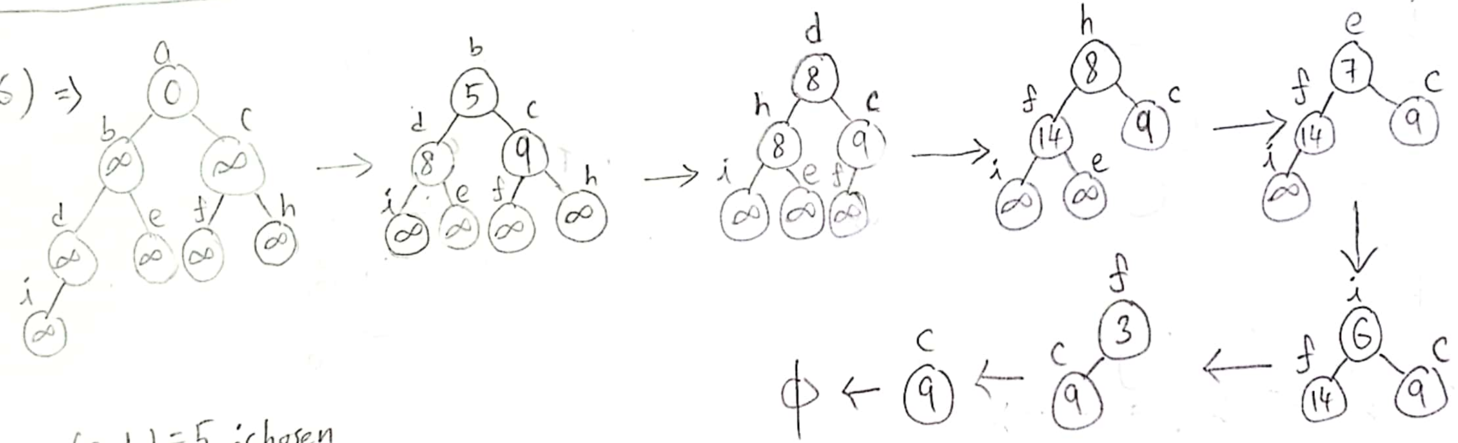


∴ minimum cost = 3+5+6+7+8+8+9
= 46

#

---

6) ⇒



⇒ (a,b) = 5 : chosen

(a,d) = 8 : chosen

(b,h) = 8 : chosen

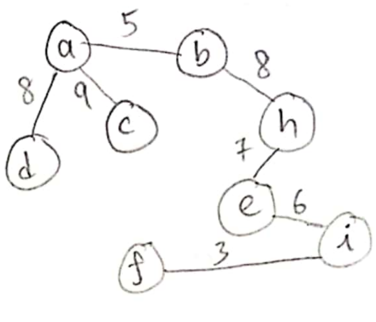
(h,e) = 7 : chosen

(e,i) = 6 : chosen

(i,f) = 3 : chosen

(a,c) = 9 : chosen

⇒ Minimum Cost Spanning Tree:

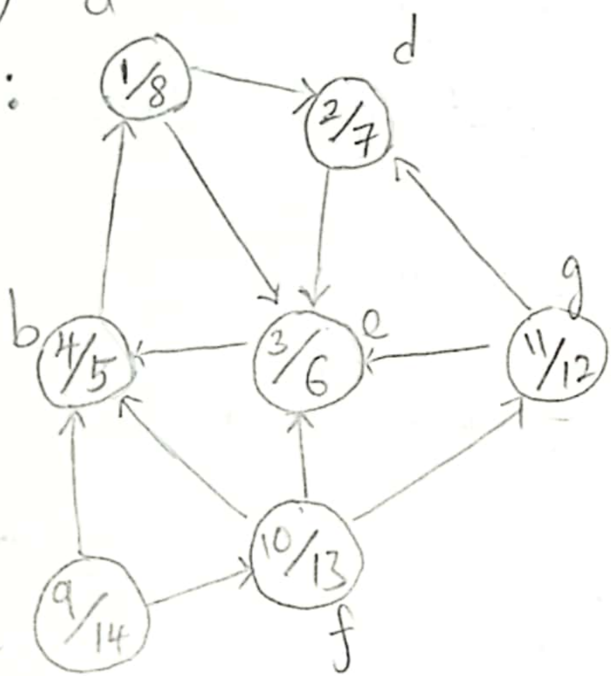


⇒ minimum cost
= 5+8+8+7+6+3+9
= 46

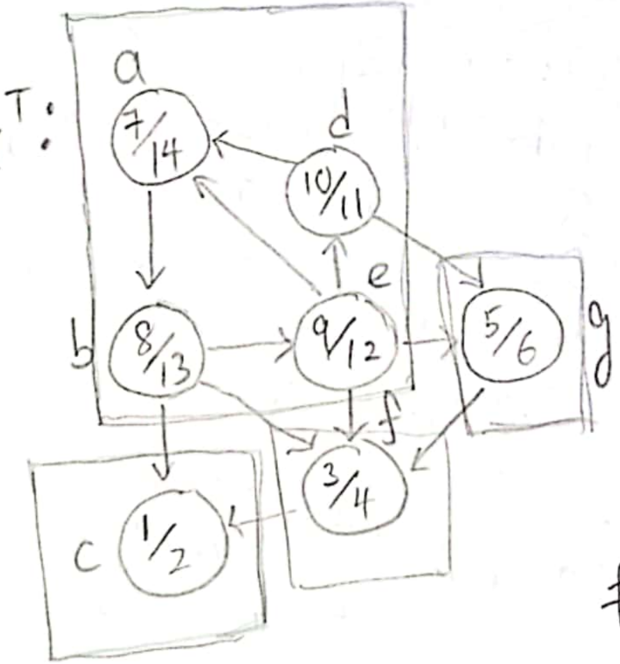
#

7)

G:



G^T:



⇒ 4 strongly connected components

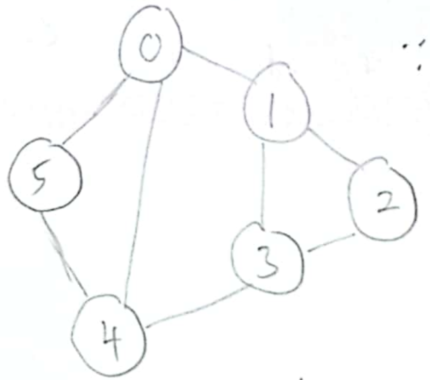
↳ ① a, b, d, e

② c

③ f

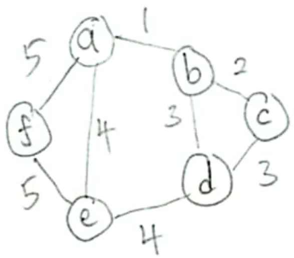
\# ④ g

8)



$\because$ cost(node u and node v) = max(u, v)

$\Rightarrow$ Let the vertex be a~f



$\Rightarrow$ (a,b) = 1 : chosen

(b,c) = 2 : chosen

(c,d) = 3 : chosen

(b,d) = 3 : reject

(d,e) = 4 : chosen

(a,e) = 4 : reject

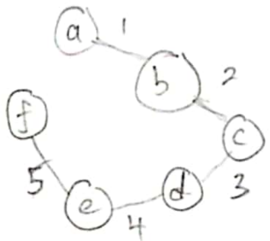
(e,f) = 5 : chosen

(a,f) = 5 : reject

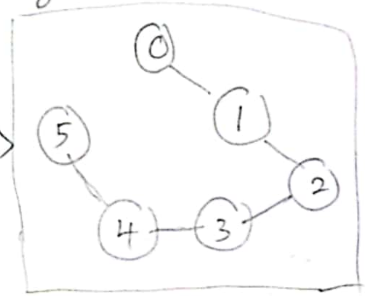
$\therefore$ minimum cost = 1+2+3+4+5

= 15

$\Rightarrow$ minimum spanning tree is :



$\Rightarrow$



#

---

9) a) T, The topological sort of an arbitrary directed graph $G = (V, E)$ can be computed using DFS and time complexity of DFS is $\Theta(V+E)$ in linear time.

b) F, Depth-First Search algorithm works for both directed and undirected graph.

c) T, Negative edges does not affect Prim's Algorithm, because it always adds a safe (light) edge into the subset of MST. Hence, Prim's Algorithm still works correctly when there are negative.

10) => To prove that a connected graph with distinct edge costs has exactly one unique minimum-cost spanning tree, we need to prove two things:

① There exists at least one minimum-cost spanning tree

② Any two minimum-cost spanning trees of the graph are identical

① ↳ Since the graph is connected, there is at least one spanning tree.

↳ Cut Property: For any cut in a graph, if the minimum cost edge of the cut is included in a spanning tree, that tree is a minimum-cost spanning tree.

② => Suppose there are two different minimum-cost spanning trees, T1 and T2, since both trees are minimum-cost spanning trees, they must have the same total cost, but since the cost of edges are all distinct, there must be at least one edge (e) that is present in T1 but not in T2 and let u, v be the vertices on either end of edge e and let P be the unique path between v and u in T2.

↳ Since e is not in T2, the path P in T2 must use other edges. Let x be the first edge encountered on path P that is not in T1. Since P is a unique path, removing edge x from P and adding edge e to it would create cycle in T1, which contradicts the fact that T1 is a minimum-cost spanning tree (does not have cycle).

∴ Therefore, the assumption that there are two distinct minimum-cost spanning tree is a contradiction. Hence, any two minimum-cost spanning trees of the graph must be identical.

⇒ ∴ If the costs of all edges in a given connected graph are distinct, then the graph has exactly one unique minimum-cost spanning tree.

(proved) #