Data Structures Final Exam

2021/01/11 @ DS class

Presenter: 吳昱宏

Scoring Criteria

除了表格題(time complexity, stable那題)一格兩分外，其餘題目皆為一題五分

考卷上每個V代表五分，每個O代表兩分，請先確認分數有沒有加錯

YOUR SCORE = COUNT(V) * 5 + COUNT(O) * 2

1. Notations 20%

Which of the following equations are correct?

(A) $n! = O(n^n)$ correct, $n!<n^n$

(B) $87n^3 + 870n^4 + 8700n^2 2^n = O(n^2 2^n)$ correct, $2^n>n^3>n^4$

(C) $2^n = O(4^n)$ correct, $4^n>2^n$

(D) $62^n + n^2= \Omega(2^n)$ correct, omega is lower bound, $62^n >2^n$

Answer: (A)(B)(C)(D)

2. Quick Sort

a. What is the average time complexity of quick sort on n numbers? 87%

b. What is the worst time complexity of quick sort on n numbers? 95%

c. If we implement quick sort by always choosing the first component as pivot, for all permutations of {1, 2, 3, 4, 5, 6}, which permutations will exhibit the worst-case behavior of quick sort? 25%

Answer: a. O(nlogn) b. O(n^2) c. 1,2,3,4,5,6 **and** 6,5,4,3,2,1

3. Radix Sort

Suppose you are to sort the following sequence using radix sort:

[18, 203, 16, 30, 123, 521, 63, 528, 210, 216, 941, 55]

Please answer the following questions.

a. At the end of first step, what is the 8th element of the new order? 80%

b. At the end of second step, what is the 8th element of the new order? 80%

Answer: a. [30, 210, 521, 941, 203, 123, 63, 55, 16, 216, 18, 528]

b. [203, 210, 16, 216, 18, 521, 123, 528, 30, 941, 55, 63]

4. Time Complexity / Stable Sort

一格兩分！

	Bubble	Selection	Heap	Merge	Insertion
Worst Case	$O(n^2)$	$O(n^2)$	O(nlogn)	O(nlogn)	$O(n^2)$
Stable	Yes	No 30%	No	Yes	Yes

Why isn’t Selection Sort Stable?

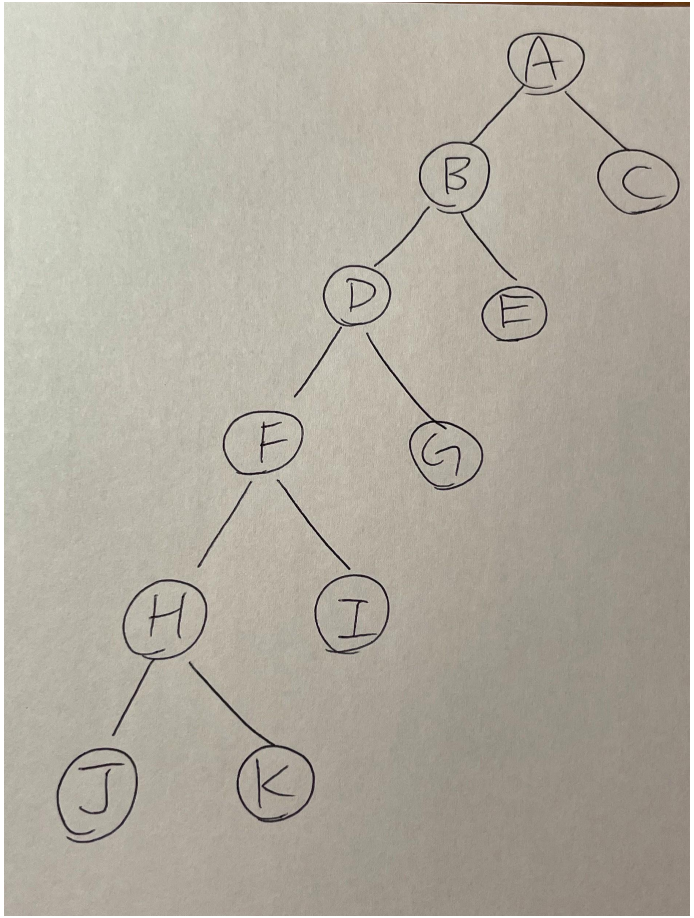
Suppose we are to sort the array `[8a, 8b, 7].` Since the smallest element is 7, we have to swap 7 and 8a. Thus, the new array will be `[7, 8b, 8a]`, which has proved the instability of selection sort.

5. Binary Tree Traversal 95%

In-order = JHKFIDGBEAC

Pre-order = ABDFHJKIGEC

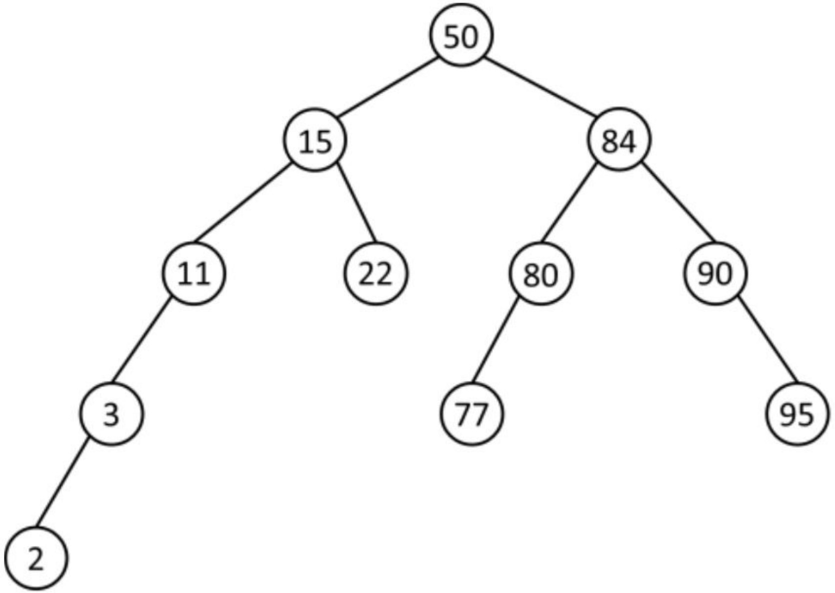
Level-order = ABCDEFGHIJK



6. AVL Tree

a. Given the figure of a binary tree. Is the tree an AVL tree? Why or why not?

80%



No, 11 and 15 are unbalanced.

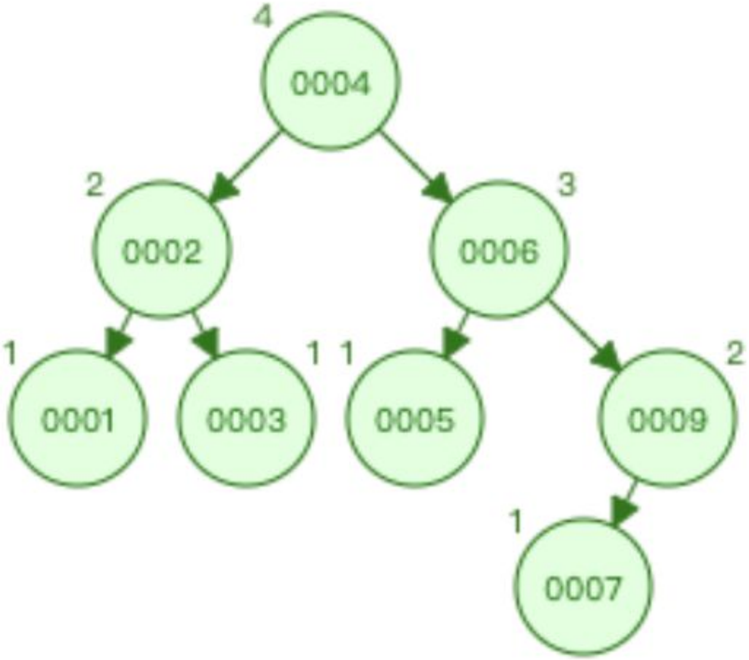
因為這棵樹不平衡 (X)

因為11不平衡 (O)

6. AVL Tree

b. Please insert 2, 1, 4, 5, 9, 3, 6, 7 into an empty AVL tree. What’s the pre-order sequence of the AVL tree? 60%

Pre-order: 42136597



7. BST

How many distinct binary search trees can be created out of 5 distinct keys? 15%

Answer: 42 (14+5+4+5+14)

不會用小畫家，請看黑板:)

Reference: Leetcode #96

8. Dijkstra's Algorithm

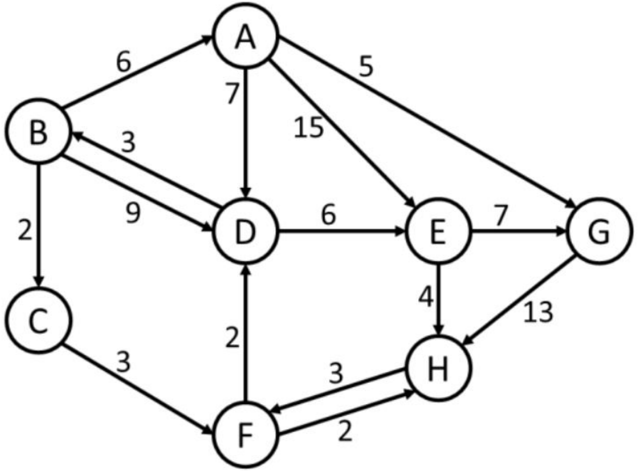
Given a weighted directed graph below, if the Dijkstra's algorithm is applied to find the shortest path starting from vertex A to all other vertices. A is the first vertex added to a set N,which is used to hold nodes that a shortest path has been found. Which of the following statements are true? 90%

(A) F is the last vertex added to set N.

(B) B is the 4th vertex added to set N.

(C) The shortest path from A to E is 13.

(D) The shortest path from A to H is 18.



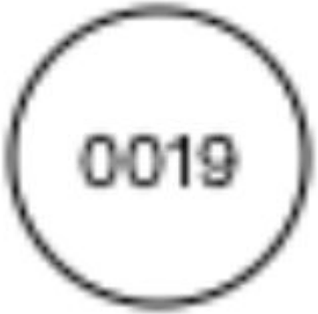
8. Dijkstra's Algorithm

Step	N	Selected	A	B	C	D	E	F	G	H
0		A	0	-	-	7	15	-	5	-
1	A	G	0	-	-	7	15	-	5	18
2	AG	D	0	10	-	7	13	-	5	18
3	AGD	B	0	10	12	7	13	-	5	18
4	AGDB	C	0	10	12	7	13	15	5	18
5	AGDBC	E	0	10	12	7	13	15	5	17
6	AGDBCE	F	0	10	12	7	13	15	5	17
7	AGDBCEFH	H	0	10	12	7	13	15	5	17

9. Heap

a. Please build a min heap by inserting19, 8, 6, 17, 2, 15, 9, 10sequentially. 60%

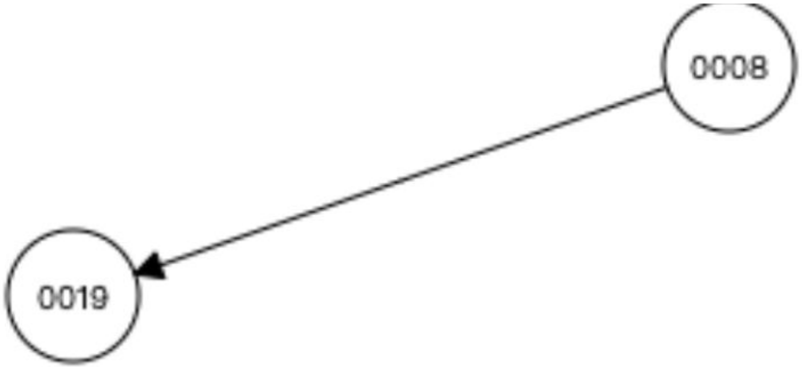
Insert 19:



9. Heap

a. Please build a min heap by inserting 19, 8, 6, 17, 2, 15, 9, 10 sequentially.

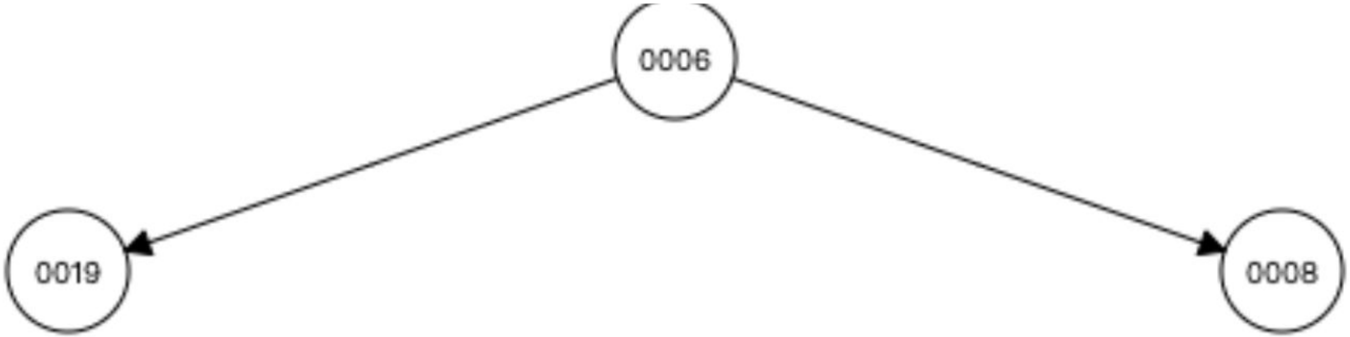
Insert 8: (swap 8&19)



9. Heap

a. Please build a min heap by inserting 19, 8, 6, 17, 2, 15, 9, 10 sequentially.

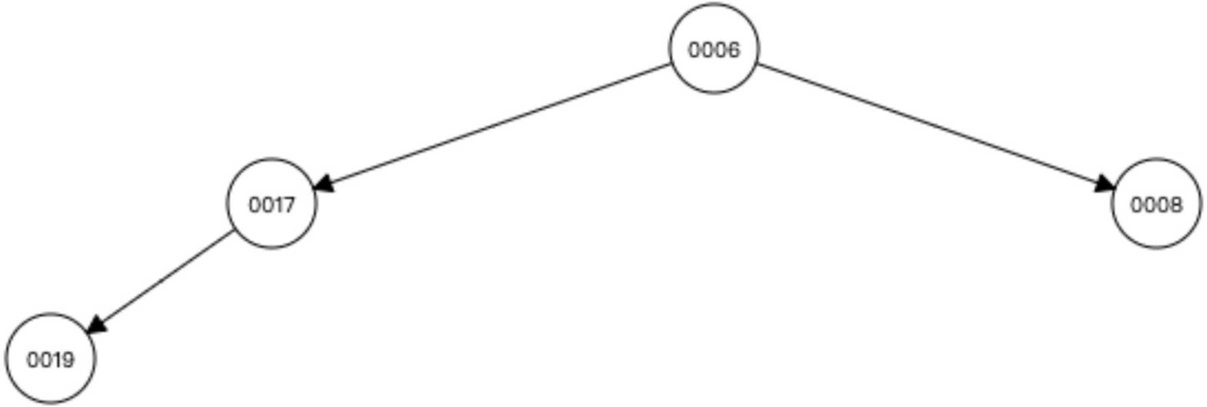
Insert 6: (swap 8&6)



9. Heap

a. Please build a min heap by inserting 19, 8, 6, 17, 2, 15, 9, 10 sequentially.

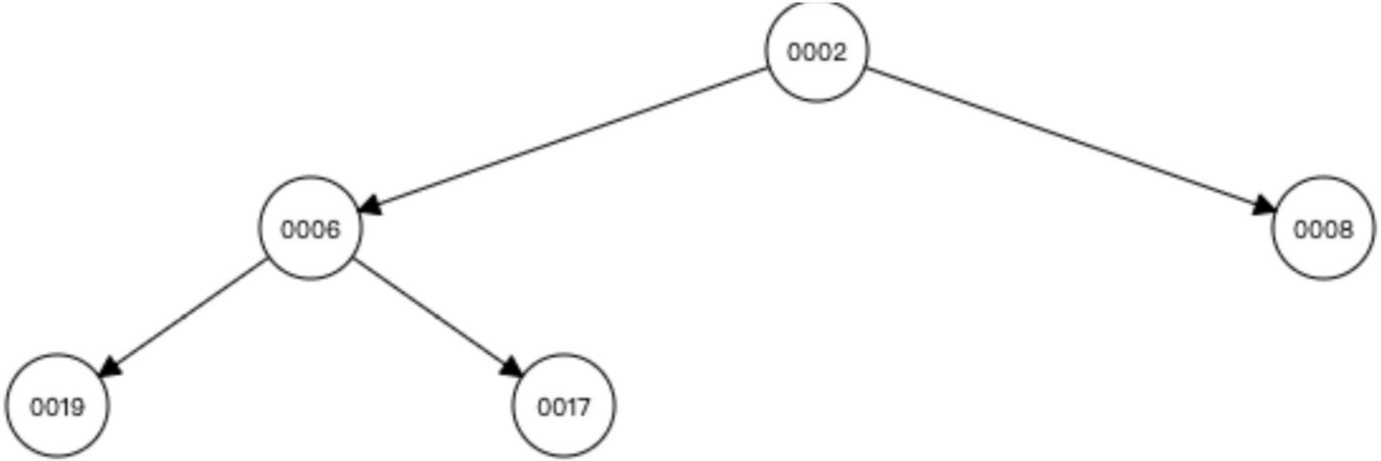
Insert 17: (swap 17&19)



9. Heap

a. Please build a min heap by inserting 19, 8, 6, 17, 2, 15, 9, 10 sequentially.

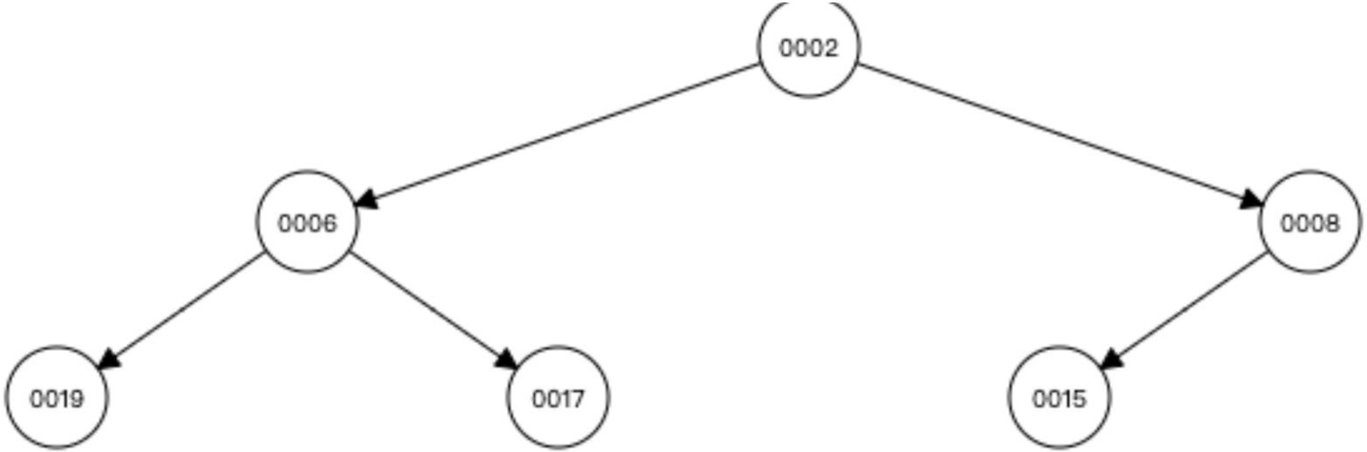
Insert 2: (swap 2&17, 2&6)



9. Heap

a. Please build a min heap by inserting 19, 8, 6, 17, 2, 15, 9, 10 sequentially.

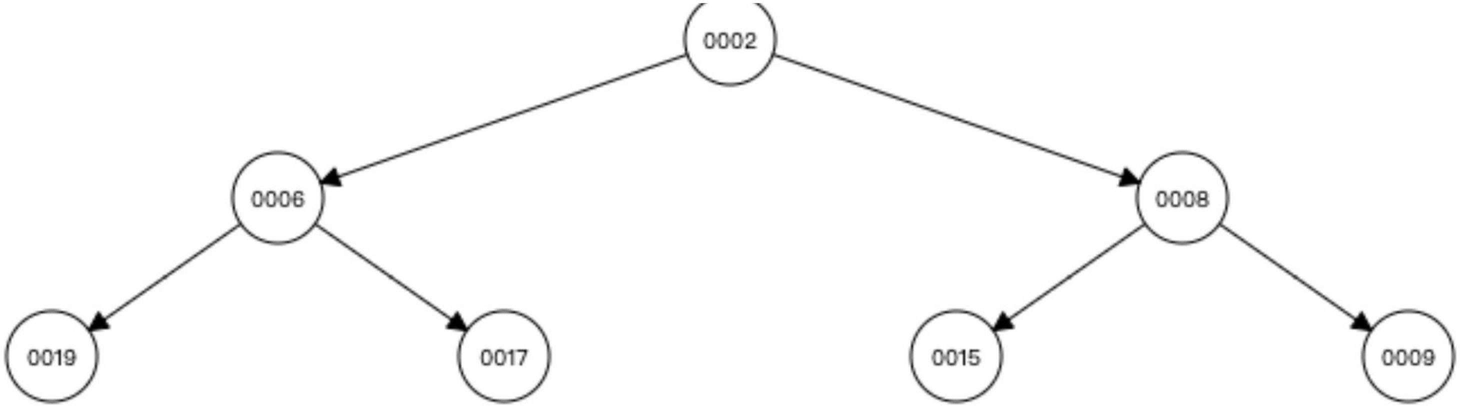
Insert 15:



9. Heap

a. Please build a min heap by inserting 19, 8, 6, 17, 2, 15, 9, 10 sequentially.

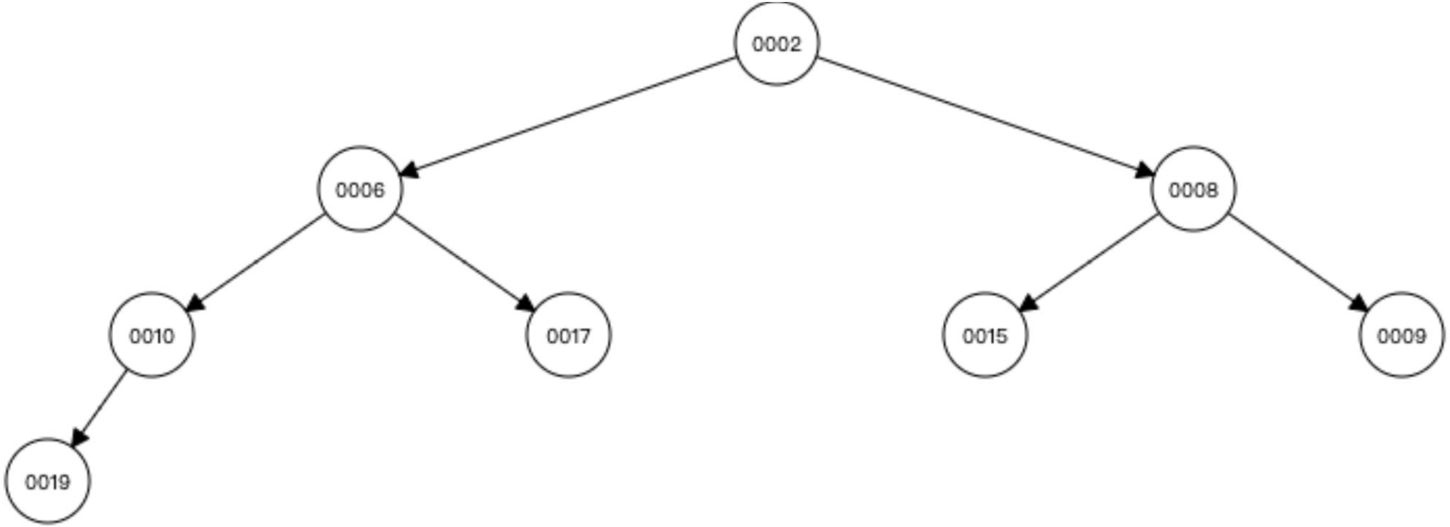
Insert 9:



9. Heap

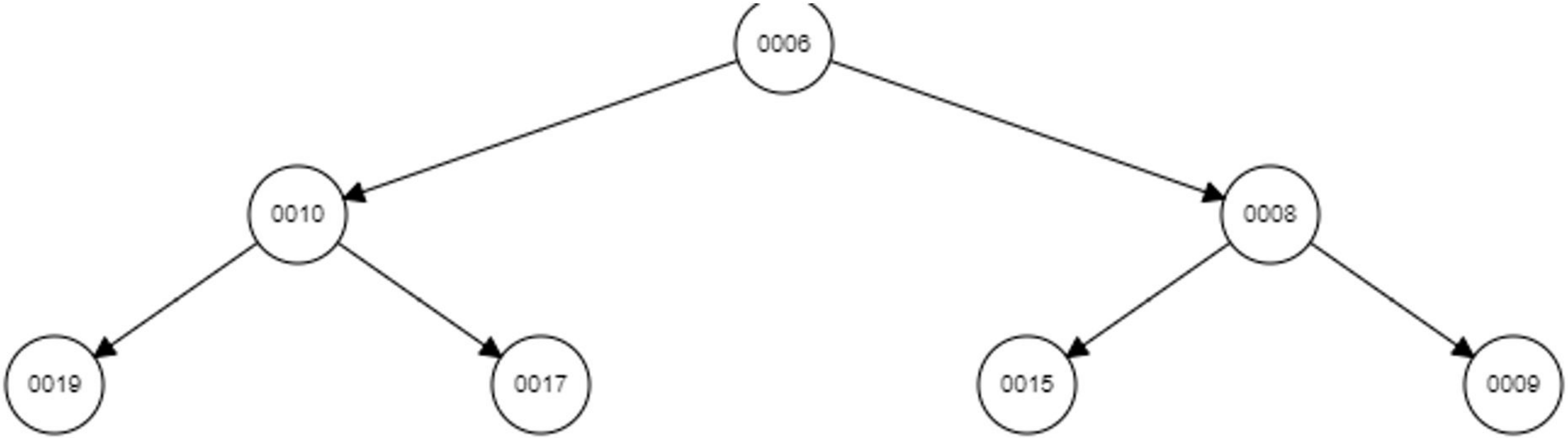
a. Please build a min heap by inserting 19, 8, 6, 17, 2, 15, 9, 10 sequentially.

Insert 10: (swap 10 & 19)



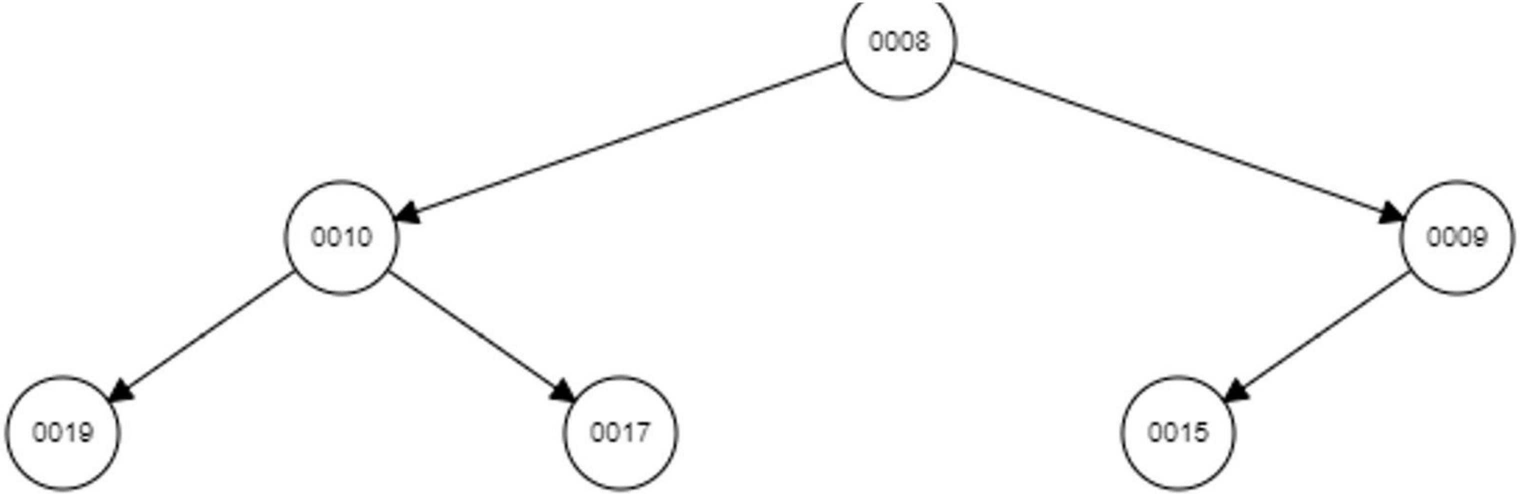
9. Heap

b. Follow the previous question, please perform the delete operation TWICE and show the resultanting min heap for each step (i.e., draw 2 heaps). 60%



9. Heap

Second deletion



9. Heap

c. What are the minimum and maximum numbers of elements in a heap of height h? 85%

Hint: The height of the root node is 1.

Completed binary tree & 期中考題目借屍還魂

Max = $2^h – 1$

Min = 2^{h-1}

10. Directed Graph 80%

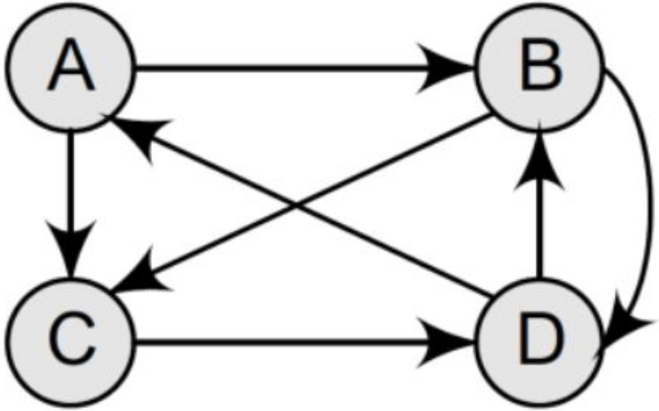
Use sequetial representation!

$$A^1 = \begin{array}{c|cccc} & A & B & C & D \\ \hline A & 0 & 1 & 1 & 0 \\ B & 0 & 0 & 1 & 1 \\ C & 0 & 0 & 0 & 1 \\ D & 1 & 1 & 0 & 0 \end{array}$$

$$A^2 = A^1 \times A^1 = \begin{bmatrix} 0012 \\ 1101 \\ 1100 \\ 0121 \end{bmatrix}$$

$$A^3 = A^2 \times A^1 = \begin{bmatrix} 2201 \\ 1221 \\ 0121 \\ 1113 \end{bmatrix}$$

$$A^4_{4,1} = [1, 1, 1, 3] \text{ X} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = 3 \text{ Ans} = 3 + 1 + 1 = 5$$



11. Double Hashing 50%

S = {16, 3, 35,67,51,1,15,31,19,17}

h1(key) = key mod 16, h2(key) = 1 + (key mod 15)

Insert 16: h(16, 0) = (h1(16) + 0 × h2(16)) mod 16 = 16%16 = 0

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
16															

11. Double Hashing

S = {16, 3, 35,67,51,1,15,31,19,17}

h1(key) = key mod 16, h2(key) = 1 + (key mod 15)

Insert 3: h(3, 0) = (h1(3) + 0 × h2(3)) mod 16 = 3%16 = 3

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
16			3												

11. Double Hashing

S = {16, 3, 35,67,51,1,15,31,19,17}

h1(key) = key mod 16, h2(key) = 1 + (key mod 15)

Insert 35: h(35, 0) = (h1(35) + 0 × h2(35)) mod 16 = 35%16 = 3 (collision!)

h(35, 1) = (h1(35) + 1 × h2(35)) mod 16 = 35%16 + 1X(1 + 35%15)= 9

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
16			3						35						

11. Double Hashing

S = {16, 3, 35,67,51,1,15,31,19,17}

h1(key) = key mod 16, h2(key) = 1 + (key mod 15)

Insert 67: h(67, 0) = (h1(67) + 0 × h2(67)) mod 16 = 67%16 = 3 (collision!)

h(67, 1) = (h1(67) + 1 × h2(67)) mod 16 = 67%16 + 1X(1 + 67%15) = 11

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
16			3						35		67				

11. Double Hashing

S = {16, 3, 35,67,51,1,15,31,19,17}

h1(key) = key mod 16, h2(key) = 1 + (key mod 15)

Insert 51: h(51, 0) = (h1(51) + 0 × h2(51)) mod 16 = 51%16 = 3 (collision!)

h(51, 1) = (h1(51) + 1 × h2(51)) mod 16 = 51%16 + 1X(1 + 51%15) = 10

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
16			3						35	51	67				

11. Double Hashing

S = {16, 3, 35,67,51,1,15,31,19,17}

h1(key) = key mod 16, h2(key) = 1 + (key mod 15)

Insert 1: h(1, 0) = (h1(1) + 0 × h2(1)) mod 16 = 1%16 = 1 (終於沒collision啦！)

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
16	1		3						35	51	67				

11. Double Hashing

S = {16, 3, 35,67,51,1,15,31,19,17}

h1(key) = key mod 16, h2(key) = 1 + (key mod 15)

Insert 15: h(15, 0) = (h1(15) + 0 × h2(15)) mod 16 = 15%16 = 15 (又沒collision了)

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
16	1		3						35	51	67				15

11. Double Hashing

S = {16, 3, 35,67,51,1,15,31,19,17}

h1(*key*) = *key* mod 16, h2(*key*) = 1 + (*key* mod 15)

Insert 31: h(31, 0) = (h1(31) + 0 × h2(31)) mod 16 = 31%16 = 15 (collision QQ)

h(31, 1) = (h1(31) + 1 × h2(31)) mod 16 = [31%16 + 1X(1 + 31%15)] %16= 1 (哭啊)

h(31, 2) = (h1(31) + 2 × h2(31)) mod 16 = [31%16 + 2X(1 + 31%15)] %16= 3 (哭阿門)

h(31, 3) = (h1(31) + 3 × h2(31)) mod 16 = [31%16 + 2X(1 + 31%15)] %16= 5 (噫！我中啦)

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
16	1		3		31				35	51	67				15

11. Double Hashing

S = {16, 3, 35,67,51,1,15,31,19,17}

h1(key) = key mod 16, h2(key) = 1 + (key mod 15)

Insert 19: h(19, 0) = (h1(19) + 0 × h2(19)) mod 16 = 19%16 = 3 (collision QAQ)

h(19, 1) = (h1(19) + 1 × h2(19)) mod 16 = 19%16 + 1X(1 + 19%15) = 8

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
16	1		3		31			19	35	51	67				15

11. Double Hashing

S = {16, 3, 35,67,51,1,15,31,19,17}

h1(key) = key mod 16, h2(key) = 1 + (key mod 15)

Insert 17: h(17, 0) = (h1(17) + 0 × h2(17)) mod 17 = 17%16 = 1 (collision 😢)

h(17, 1) = (h1(17) + 1 × h2(17)) mod 16 = 17%16 + 1X(1 + 17%15) = 4

Final Answer: (沒寫-1沒關係)

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
16	1		3	17	31			19	35	51	67				15

12. Stack 80%

(5%) Please transfer the following infix expression to its postfix form:

a ÷ (b + c × d) × (e - f) + g

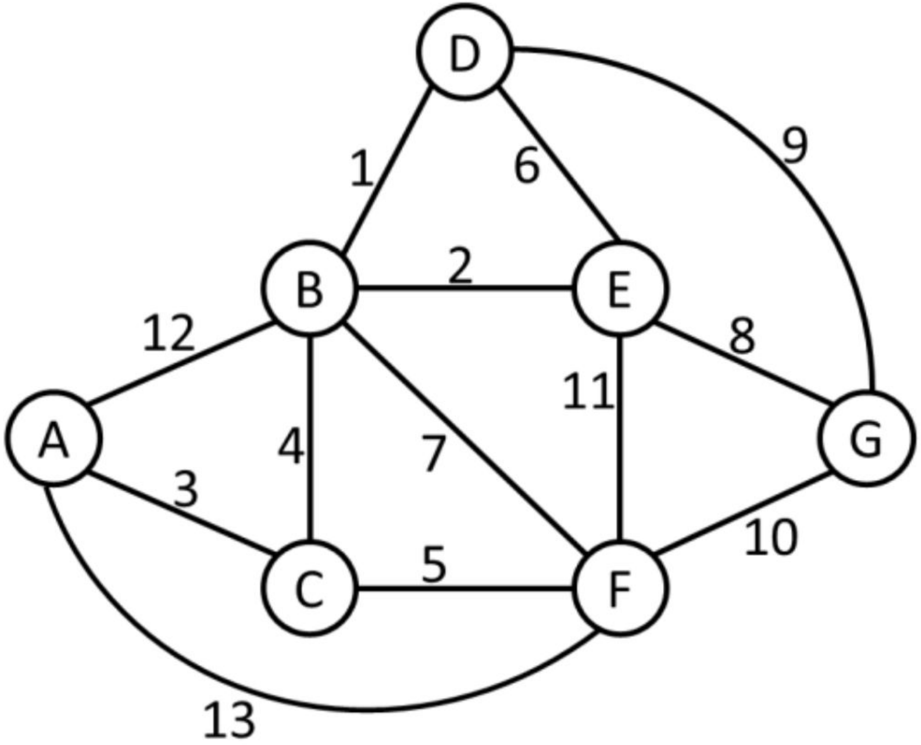
Answer: a b c d × + ÷ e f - × g +

13. MST & Graph Traversal

a. Prim’s algorithm 70%

B -> D(1) -> E(2) -> C(4) -> A(3)

-> F(5) -> G(8)



13. MST & Graph Traversal

b. Krustal’s Algorithm 70%

Step 1:

F = {{A}, {B}, {C}, {D}, {E}, {F}, {G}}

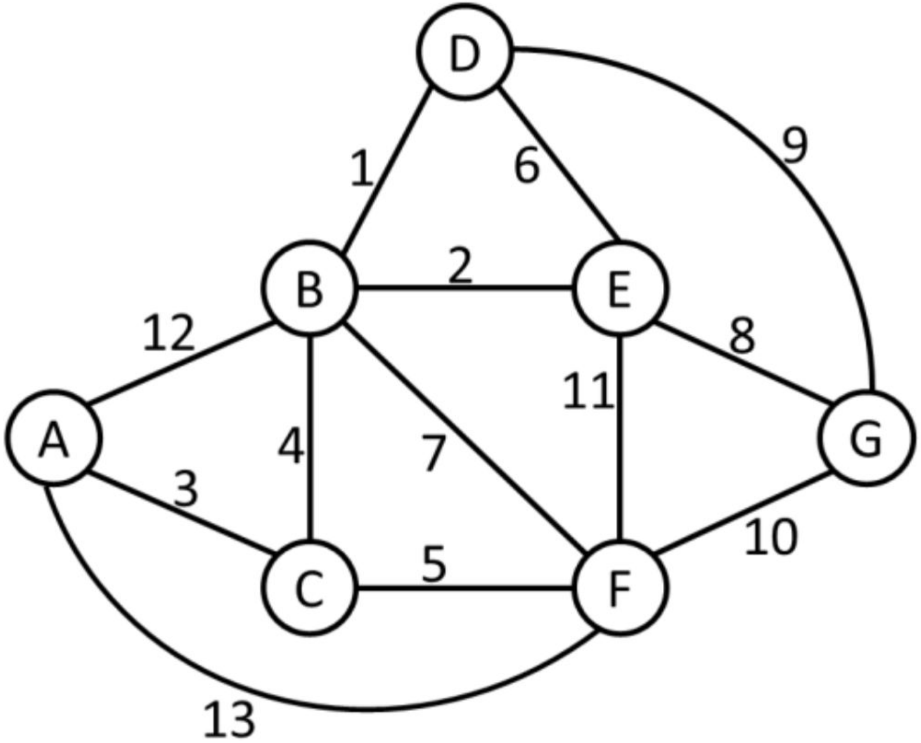
MST = {}

Q = {(B, D), (B, E), (A, C), (B, C),

(C, F), (D, E), (B, F), (E, G),

(D, G), (G, F), (E, F), (A, B),

(A, F)}



13. MST & Graph Traversal

b. Krustal's Algorithm

Step 2:

F = {{A}, {B}, {C}, {D}, {E}, {F}, {G}}

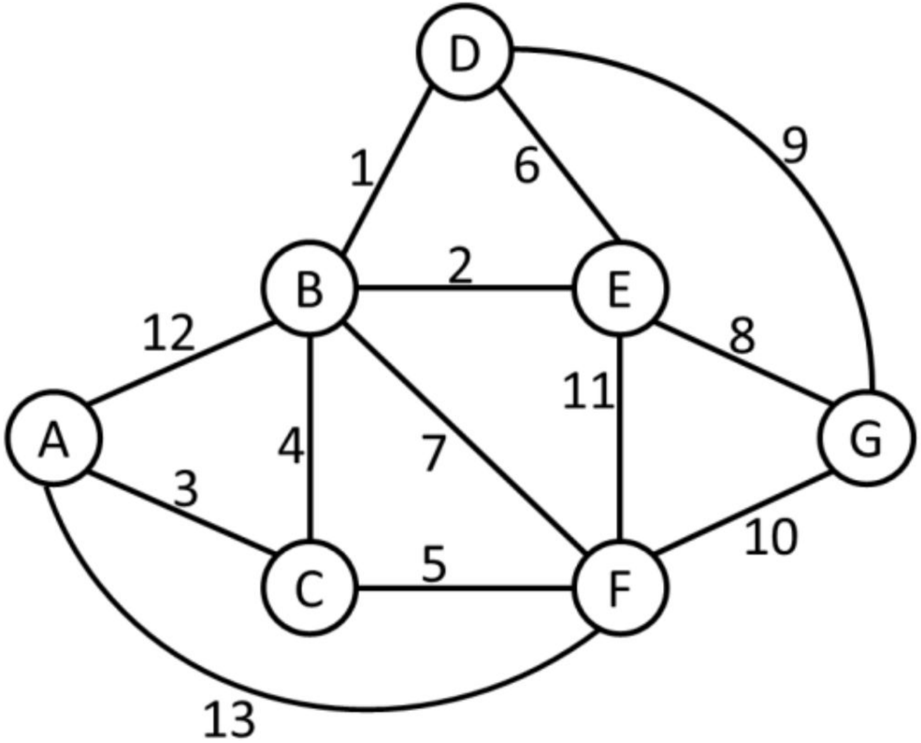
MST = {(B, D)}

Q = {~~(B, D)~~, (B, E), (A, C), (B, C),

(C, F), (D, E), (B, F), (E, G),

(D, G), (G, F), (E, F), (A, B),

(A, F)}



13. MST & Graph Traversal

b. Krustal's Algorithm

Step 3:

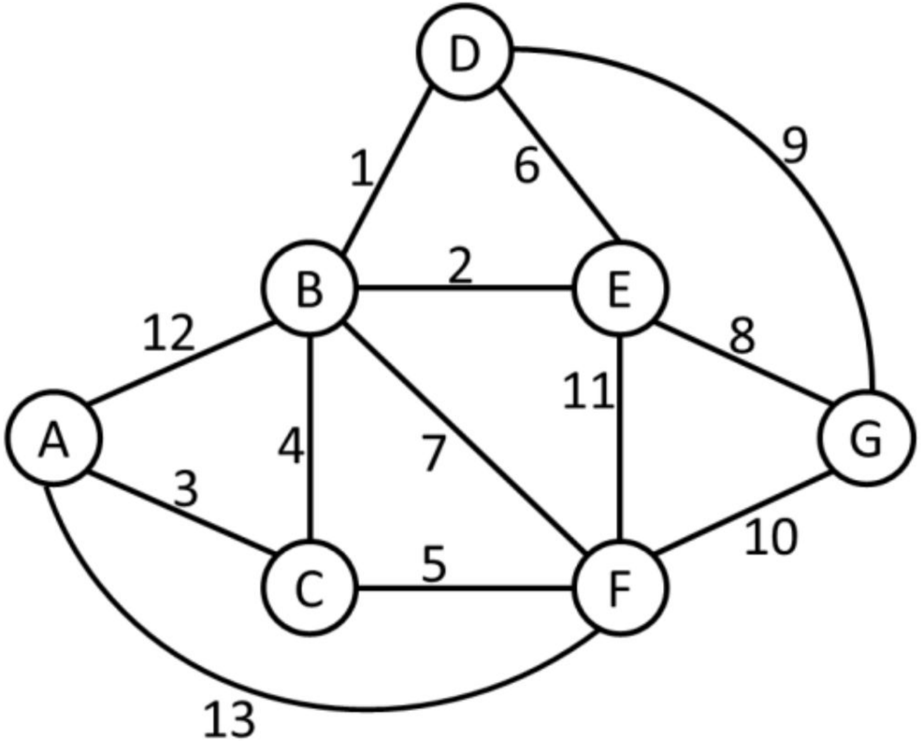
F = {{A}, {B, D, E}, {C}, {F}, {G}}

MST = {(B, D), (B, E)}

Q = {~~(B, E)~~, (A, C), (B, C), (C, F),

(D, E), (B, F), (E, G), (D, G),

(G, F), (E, F), (A, B), (A, F)}



13. MST & Graph Traversal

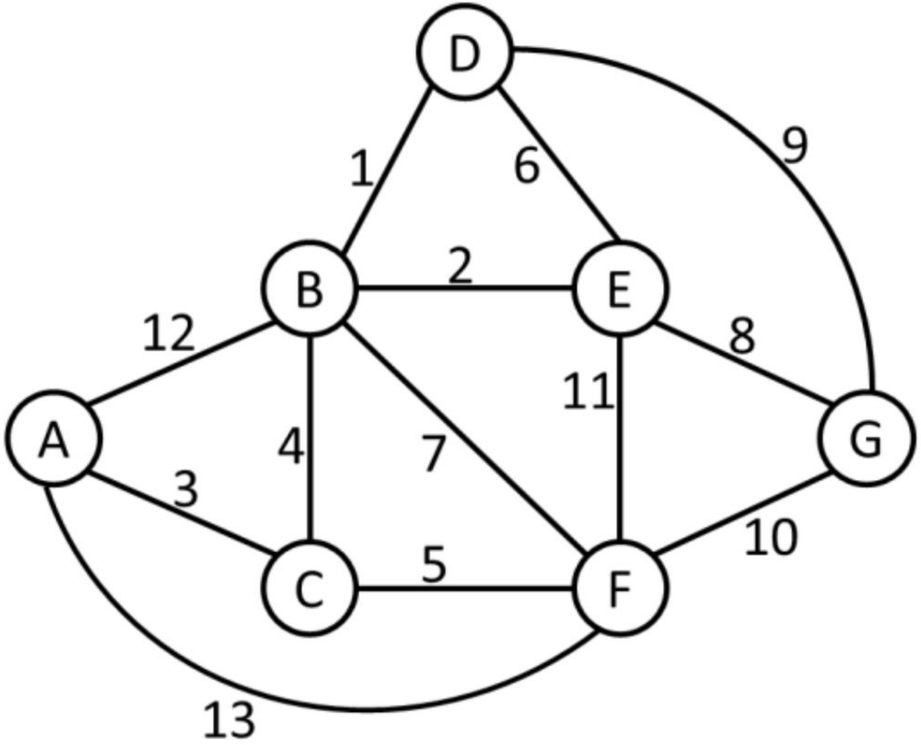
b. Krustal's Algorithm

Step 4:

F = {{A, C}, {B, D, E}, {F}, {G}}

MST = {(B, D), (B, E), (A, C)}

Q = {~~(A, C),~~ (B, C), (C, F), (D, E),
(B, F), (E, G), (D, G), (G, F),
(E, F), (A, B), (A, F)}



13. MST & Graph Traversal

b. Krustal's Algorithm

Step 5:

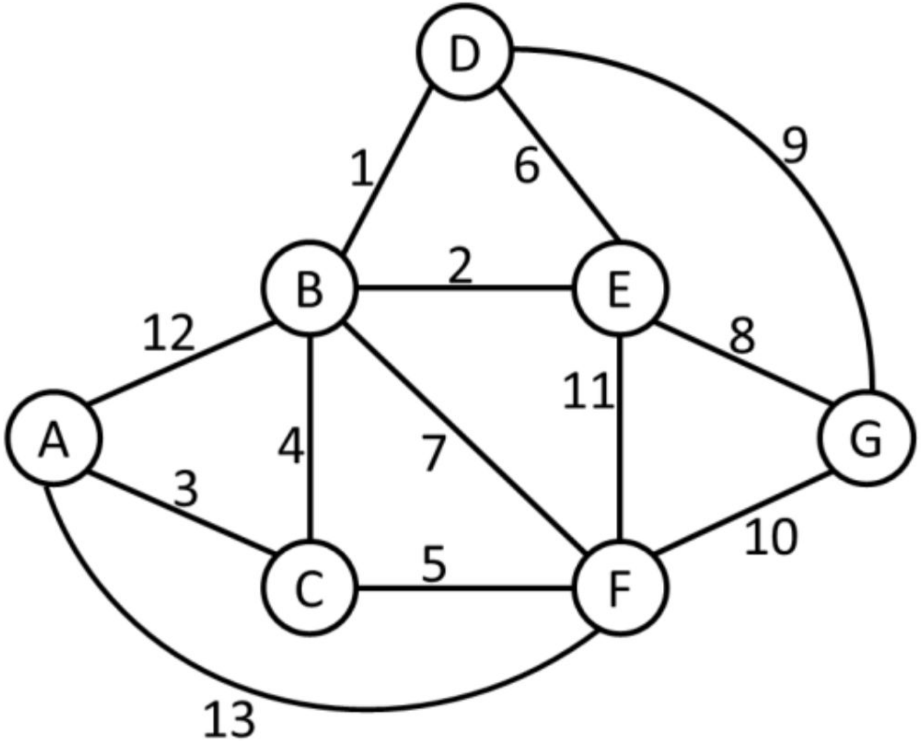
F = {{A, C, B, D, E}, {F}, {G}}

MST = {(B, D), (B, E), (A, C), (B, C)}

Q = {~~(B, C)~~, (C, F), (D, E), (B, F),

(E, G), (D, G), (G, F), (E, F),

(A, B), (A, F)}



13. MST & Graph Traversal

b. Krustal's Algorithm

Step 6:

F = {{A, C, B, D, E, F}, {G}}

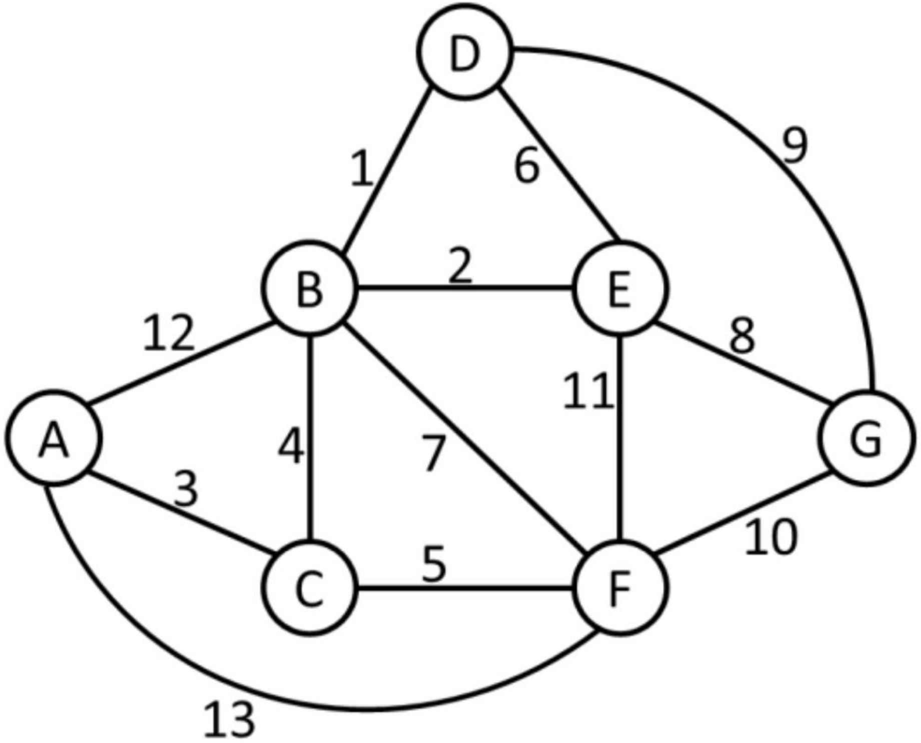
MST = {(B, D), (B, E), (A, C), (B, C)

(C, F)}

Q = {~~(C, F)~~, (D, E), (B, F), (E, G),

(D, G), (G, F), (E, F), (A, B),

(A, F)}



13. MST & Graph Traversal

b. Krustal’s Algorithm

Step 7: (dismiss)

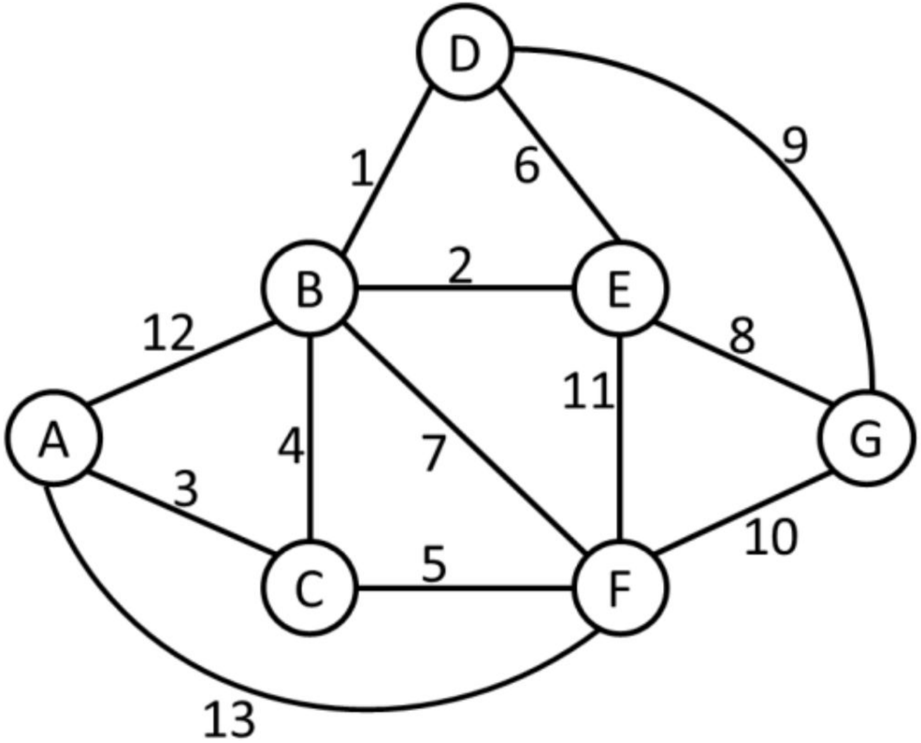
F = {{A, C, B, D, E, F}, {G}}

MST = {(B, D), (B, E), (A, C), (B, C)

(C, F)}

Q = {~~(D, E)~~, (B, F), (E, G), (D, G),

(G, F), (E, F), (A, B), (A, F)}



13. MST & Graph Traversal

b. Krustal's Algorithm

Step 8: (dismiss)

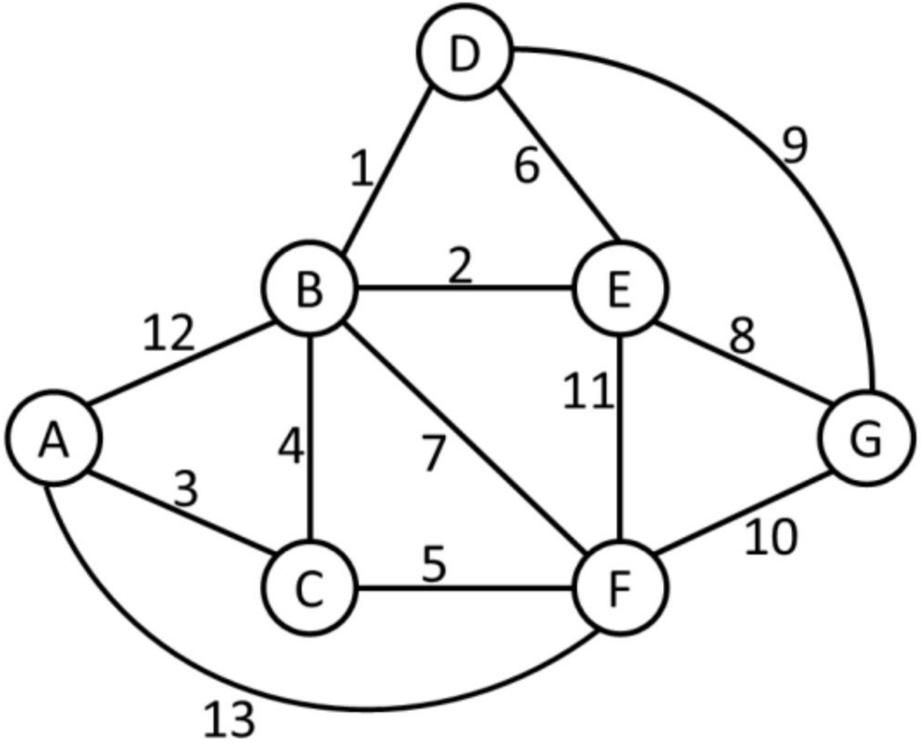
F = {{A, C, B, D, E, F}, {G}}

MST = {(B, D), (B, E), (A, C), (B, C)

(C, F)}

Q = {~~(B, F)~~, (E, G), (D, G), (G, F),

(E, F), (A, B), (A, F)}



13. MST & Graph Traversal

b. Krustal's Algorithm

Step 9: (DONE)

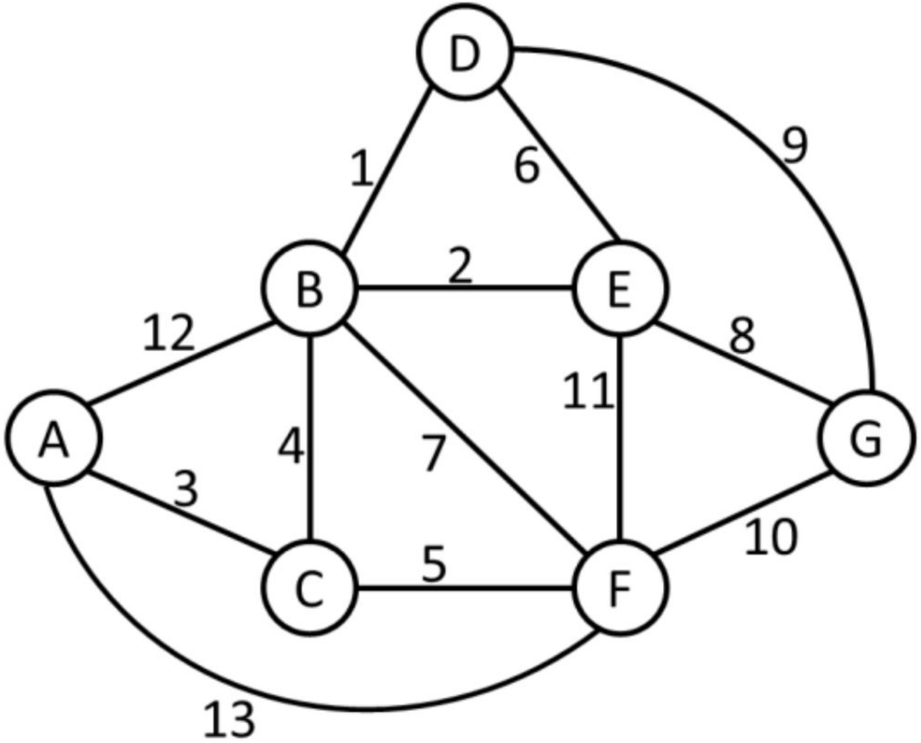
F = {{A, C, B, D, E, F, G}}

MST = {(B, D), (B, E), (A, C), (B, C)

(C, F), (E, G)}

Q = {~~(E, G)~~, (D, G), (G, F),

(E, F), (A, B), (A, F)}



13. MST & Graph Traversal

b. Krustal's Algorithm

Step 9: (DONE)

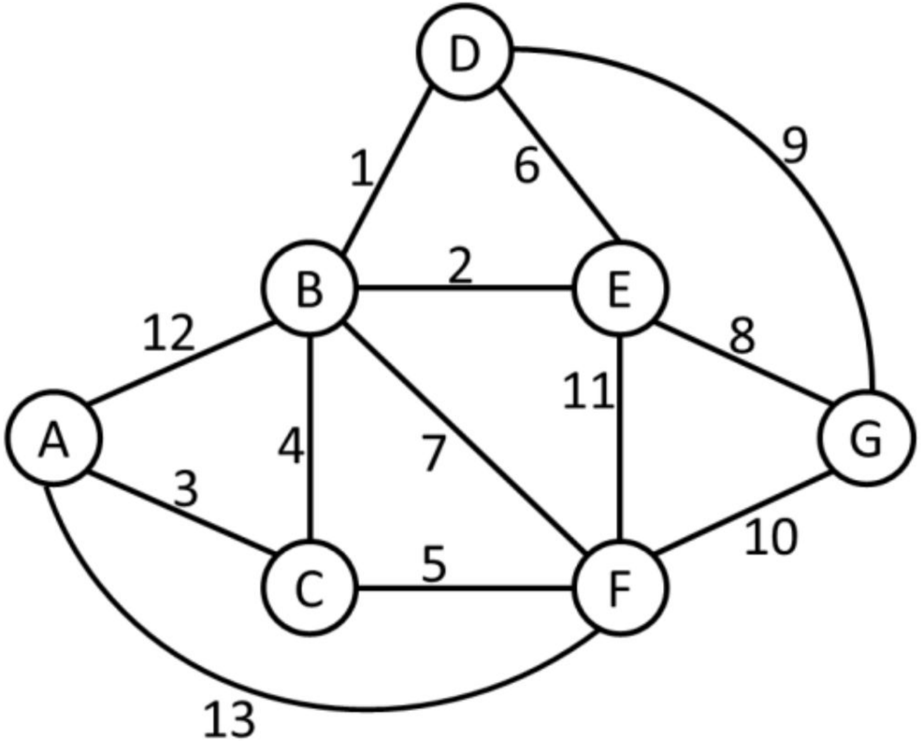
F = {{A, C, B, D, E, F, G}}

MST = {(B, D), (B, E), (A, C), (B, C)

(C, F), (E, G)}

Q = {~~(E, G)~~, (D, G), (G, F),

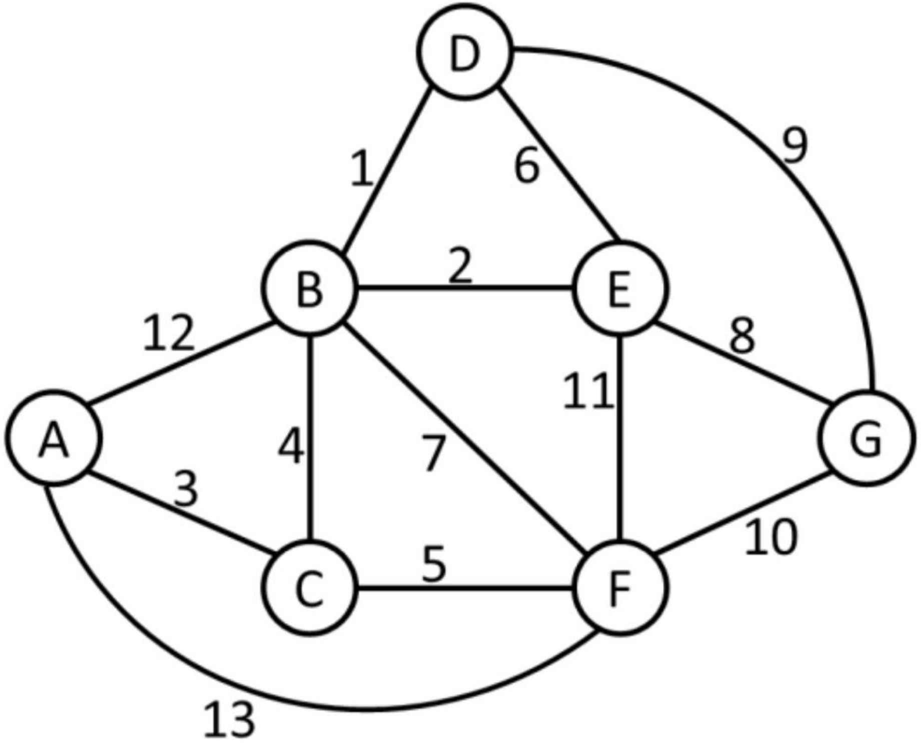
(E, F), (A, B), (A, F)}



13. MST & Graph Traversal

c. DFS from vertex E 30%

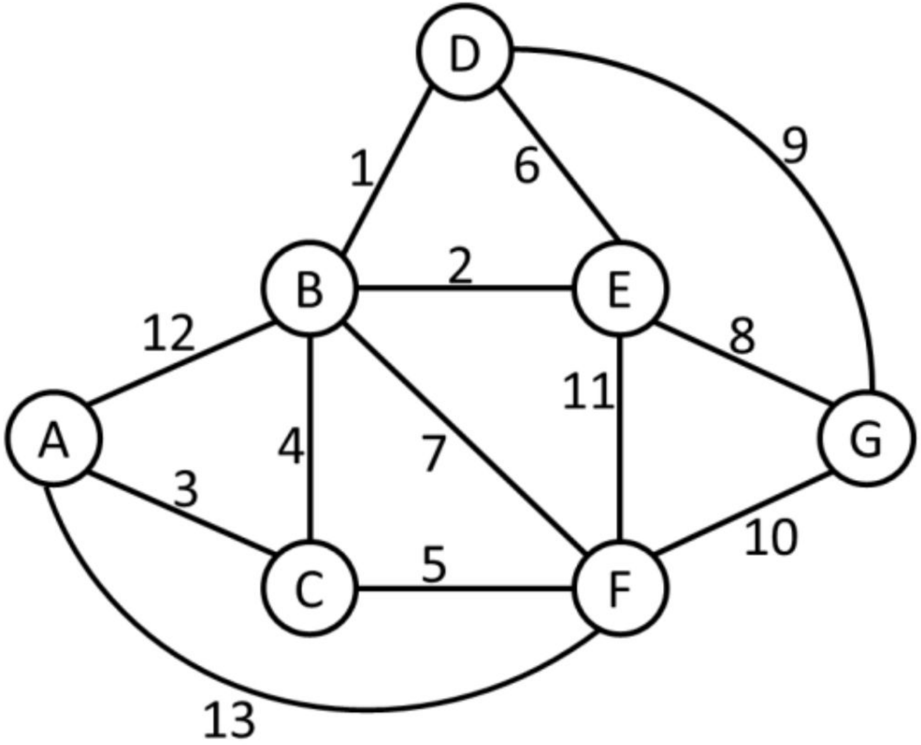
E -> B -> D -> G -> F -> C -> A



13. MST & Graph Traversal

d. BFS from vertex E 40%

E -> B -> D -> G -> F -> C -> A



Statistics

120~130	1 (0.9%)
110~119	7 (6.4%)
100~109	10 (9.2%)
90~99	17 (15.7%)
80~89	17 (15.7%)
70~79	14 (12.9%)
60~69	12 (11.1%)
Below 60	30 (27.8%)

Average: 76

Median: 78

Reminder

1. Homework #4 #5仍可補交
2. 答案卷務必交回!!!!

Questions?