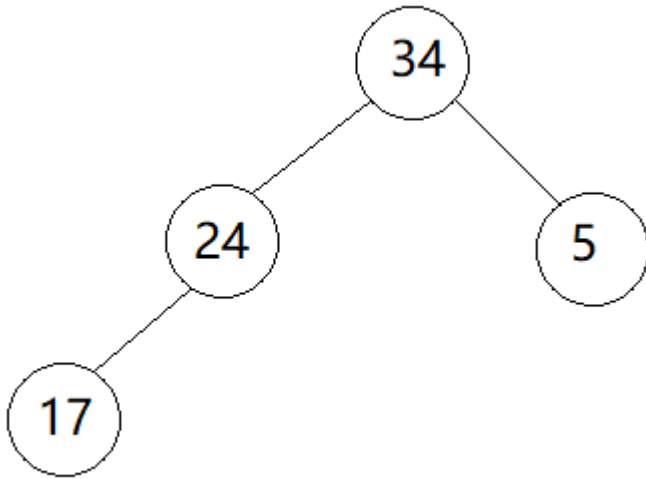


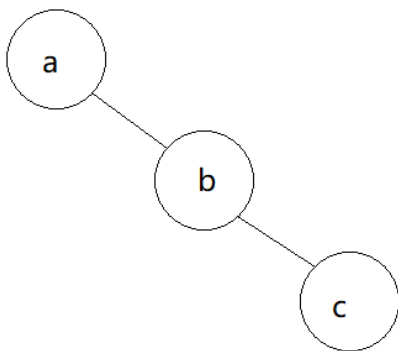
108-1 Data Structure Quiz 2 解答

1.



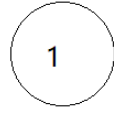
2.

(1)



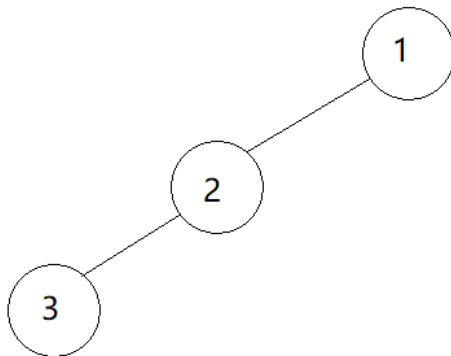
只要是 right skewed binary tree 或 一個 node 的 tree 就算對

(2)



只要一個 node 的 tree 就算對

(3)



只要是 left skewed binary tree 或 一個 node 的 tree 就算對

3.

(1)14

(2) $B(n) = \sum_{i=0}^{n-1} B(i) * B(n-1-i)$

4.

(1)T

(2)F 因為 min heap 一個 node 的 insert 需要  $O(\log_2 n)$ , initializing a min heap with n node 就是插入 n 個 node, 故需要  $O(n \log_2 n)$

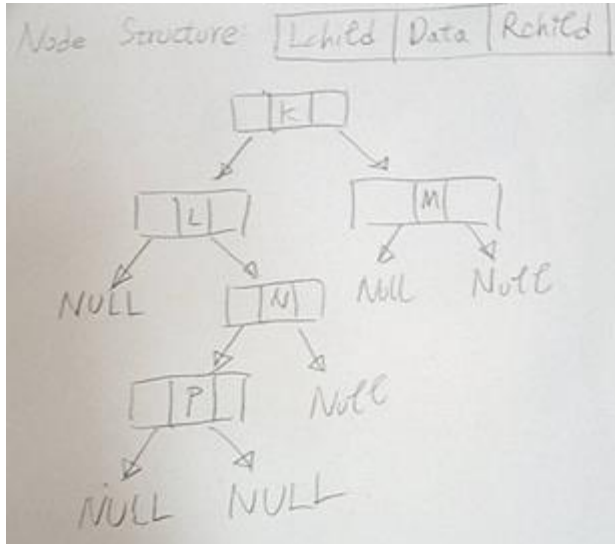
(3)F 與 worst case 一樣也是  $O(n \log_2 n)$

(4)T

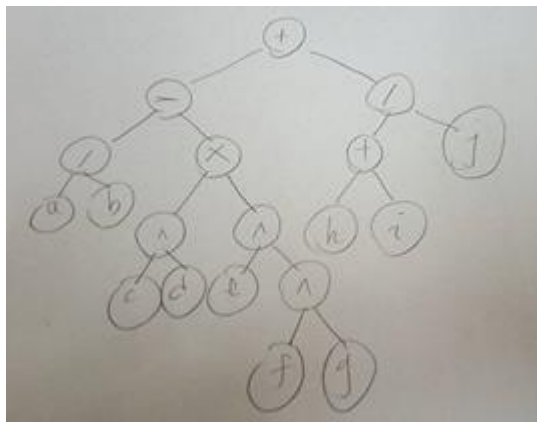
5. (1)

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15					
K	L	M		N					P										

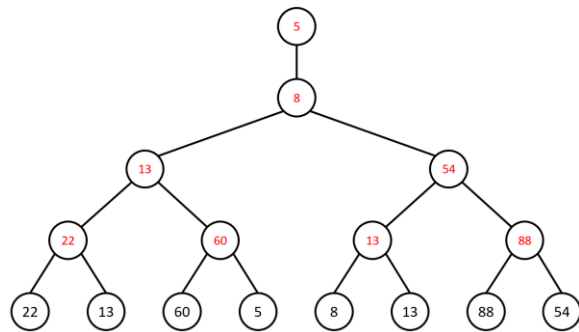
(2)



6.

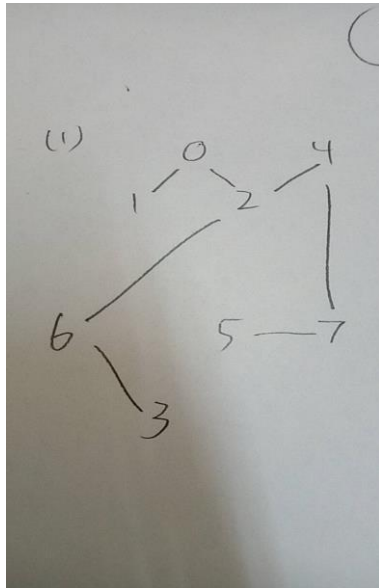


7.

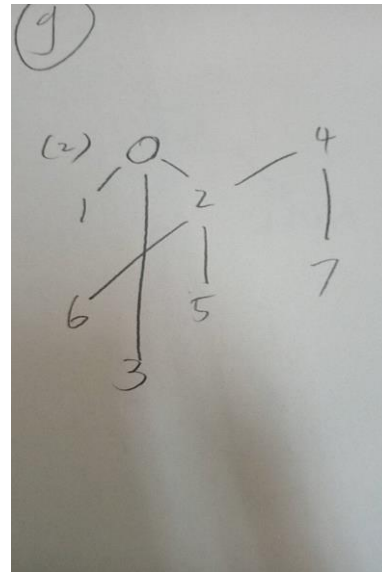


8.

(1)



(2)



9.

$$D^k[i, j] = \begin{cases} w[i, j] & \text{if } k=0 \\ \text{Min}(D^{k-1}[i, j], D^{k-1}[i, k] + D^{k-1}[k, j]) & \text{if } k \geq 1 \end{cases}$$

10.

				[1]	[2]	[3]	[4]	[5]	[6]
init				0	35	$\infty$	$\infty$	$\infty$	$\infty$
1	{1}	2		0	35	80	65	$\infty$	$\infty$
2	{1, 2}	4		0	35	80	65	110	195
3	{1, 2, 4}	3		0	35	80	65	105	195
4	{1, 2, 3, 4}	5		0	35	80	65	105	195

11.

-3

Bellman and Ford algorithm

12.

(1) root->left\_child

(2) root->right\_child

(3) distance[i][j] = cost[i][j]

(4) distance[i][k]+distance[k][j]

(5) distance[i][k]+distance[k][j]

13.

(1)

10, 25, 22, 12, 16, 14

(2)

10, 12, 14, 16, 22, 25

14.

Tree1 3

Tree2 6

15.

(1)

Big  $O(1)$

因為最大值只有可能是在 level1 的兩個 node 中的一個, 因此只要比較兩者大小就好.

(2)

Big  $O(\log n)$

因為插入的新 node 位置在原 tree 的最後一層的最後一個 node 的後面, 所以要調整這個 tree 時要向上逐層比大小, 所以時間複雜度就是這個 tree 的高度