Aurelio-Ceejay Guiking

Dr. Wolfgang Bein

CS 302 - 1003

20 March 2018

Assignment 7

* 1. 4.1

root = A

leaves = G, H, I, L, M, K

* 1. 4.2

node A

parent = none

children = B, C

siblings = none

depth = 0

height = 4

node B

parent = A

children = D, E

sibling = C

depth = 1

height = 3

node C

parent = A

children = F

siblings = B

depth = 1

height = 2

node D

parent = B

children = G, H

siblings = E

depth = 2

height = 1

node E

parent = B

children = I, J

siblings = D

depth = 2

height = 2

node F

parent = C

children = K

siblings = none

depth = 2

height = 1

node G

parent = D

children = none

siblings = H

depth = 3

height = 0

node H

parent = D

children = none

siblings = G

depth = 3

height = 0

node I

parent = E

children = none

siblings = J

depth = 3

height = 0

node J

parent = E

children = L, M

siblings = I

depth = 3

height = 1

node K

parent = F

children = none

siblings = none

depth = 3

height = 0

node L

parent = J

children = none

siblings = M

depth = 4

height = 0

node M

parent = J

children = none

siblings = L

depth = 4

height = 0

* 1. 4.5 Show that the maximum number of nodes in a binary tree of height h is 2^(h+1) - 1

Proof by induction

h = 0

2^(h+1)-1 = 1 for h = 0

Therefore true for h = 0

Assume max number of nudes in a binary tree of height is 2^(h+1)-1, for

h = 1, 2, ….. , k

tree T of height k + 1. root at T has a left subtree and a right subtree each of which has height at most K.

At most, 2^(k+1)-1 nodes. Adding the root node gives the maximum number of nodes in a binary tree of height k + 1.

Hence,

2(2^(k+1)-1)+1

2\* 2^(k+1) - 2 + 1

2^((k+1)+1) - 1

2.

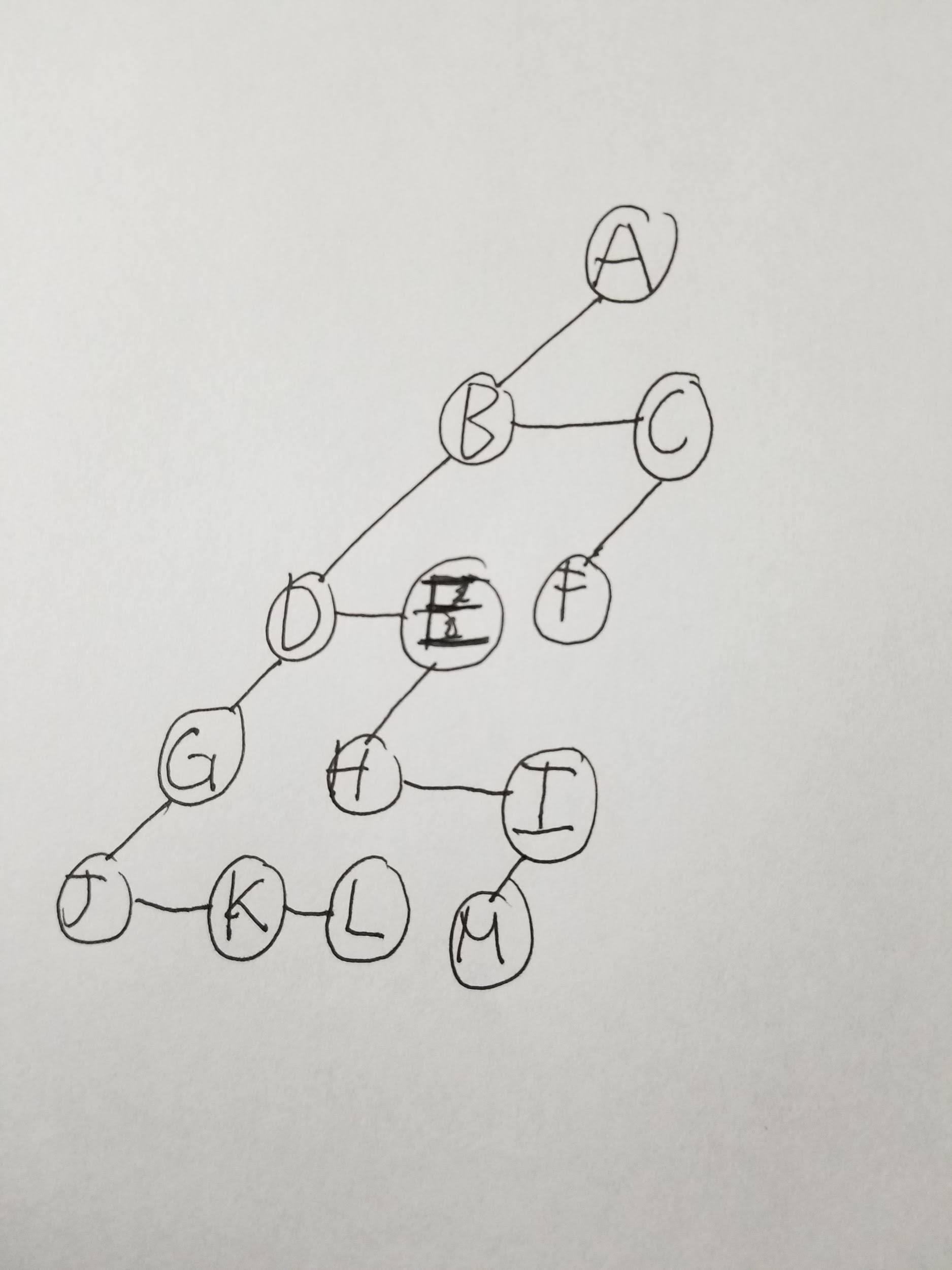
1. Preorder

A, B, D, G, J, K, L, E, H, I, M, C, F

1. Postorder

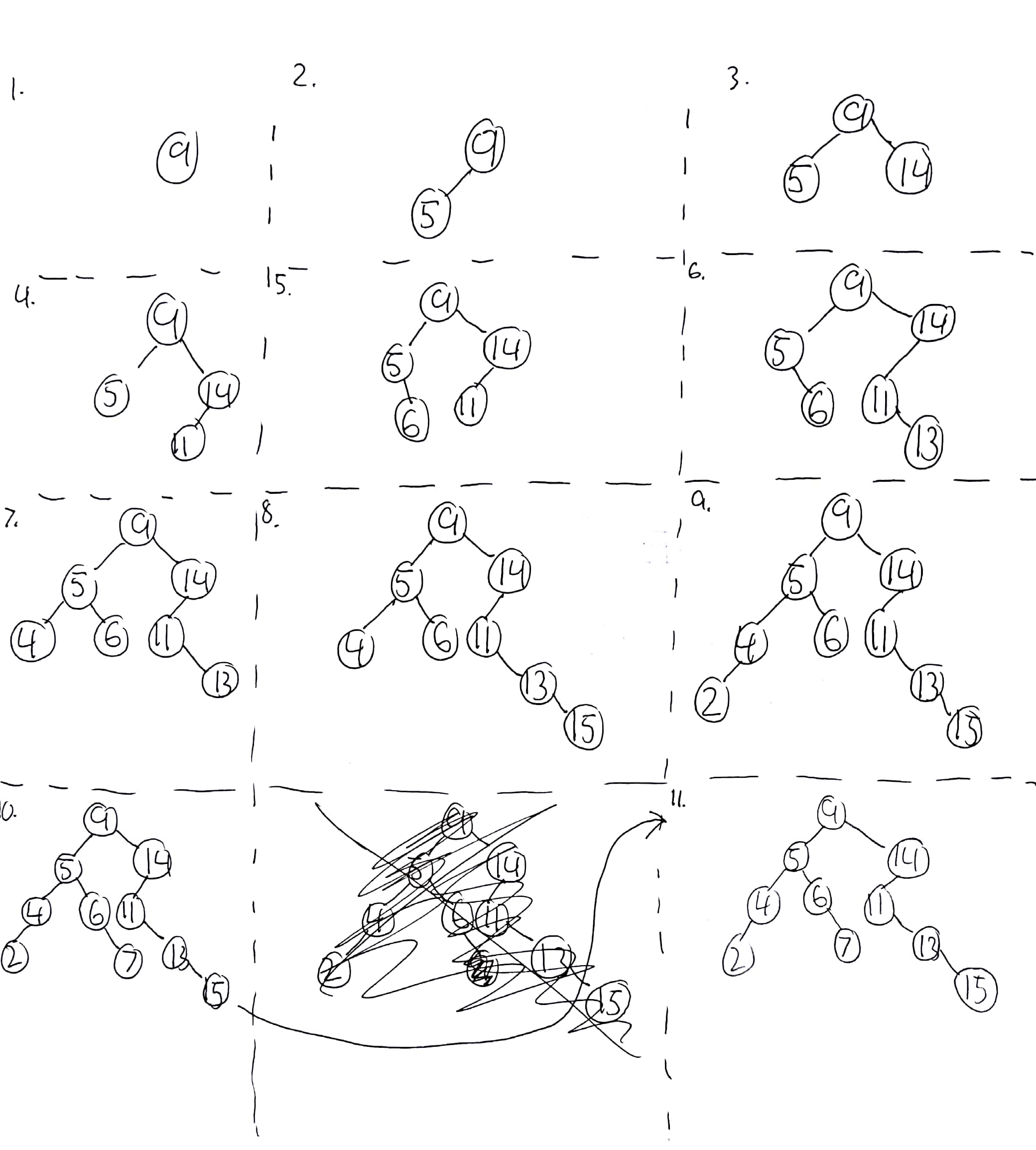
J, K, L, G, D, H, M, I, E, B, F, C, A

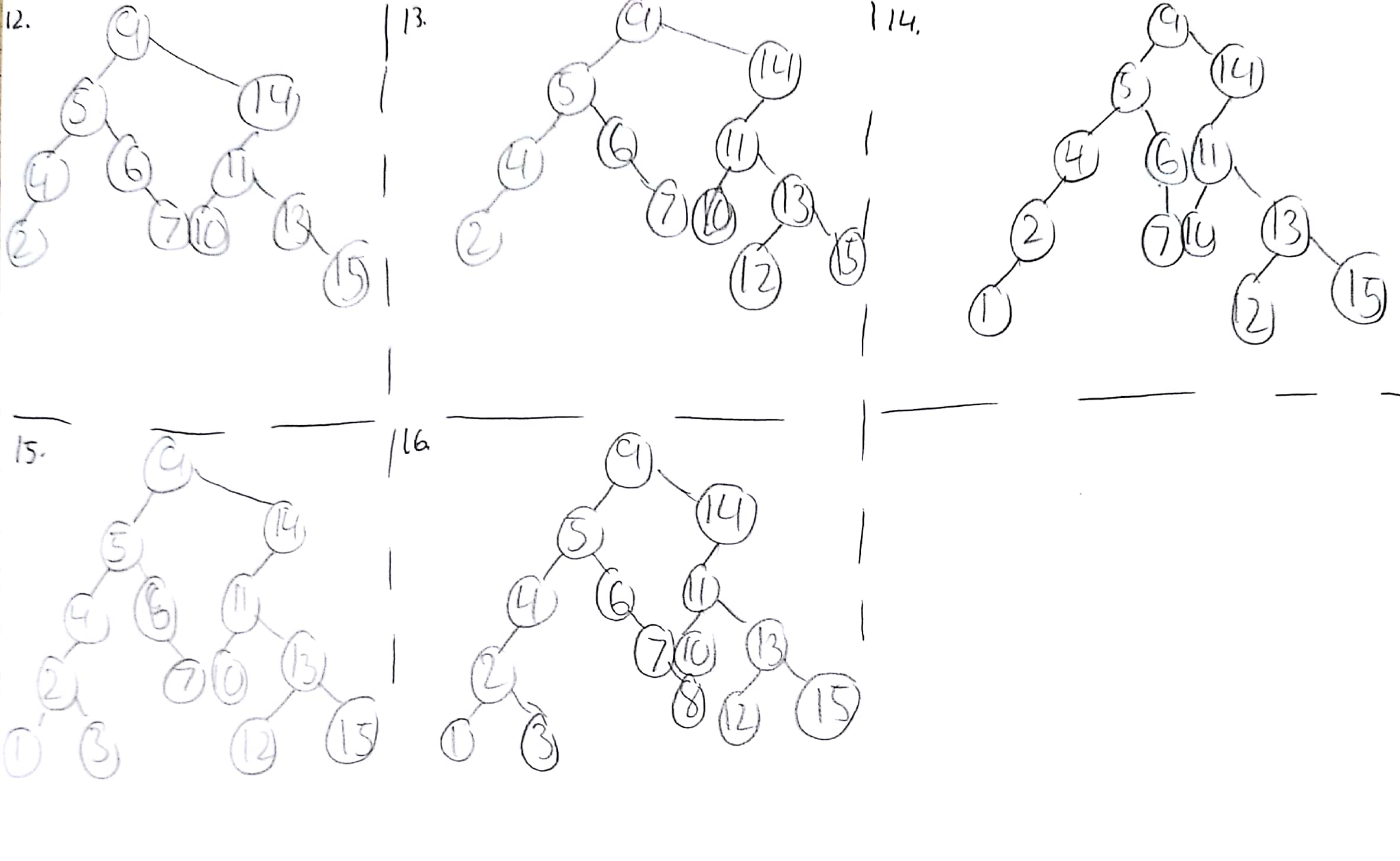
1. Left-child right-sibling



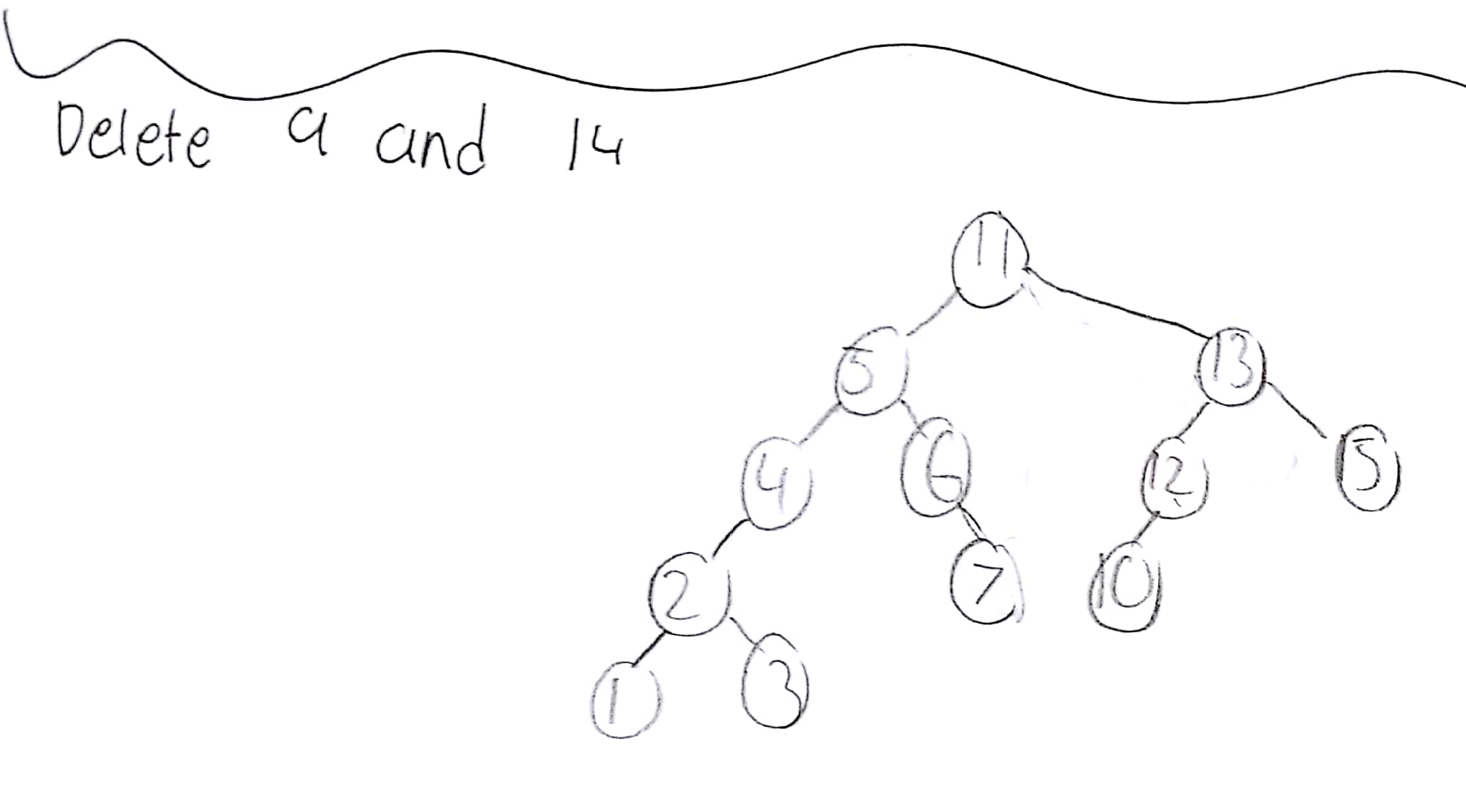
3.

a. Inserting the elements one by one





b. Deleting



4. Height of a Binary Search Tree which accommodates 1000 nodes

1. worst case

n, which is 1000 nodes

height = 1000

1. best case

log\_2(1000) ~ roughly 9.9

best case at height = 10

5. Answer is in the book. Look at avl.pdf

6.

7.

1. log\_2(100) = roughly 6.6

Max height < 7

worst case and best case = 7

b. log\_2(500) = roughly 8.9

Max height < 9

worst case and best case = 9

8.

1. log\_2(300) = roughly 8.2

Max height < 9

worst case = 9

b.

best case = 9