1. Suppose that the following keys are inserted into an initially empty linear-probing hash table, but not necessarily in the order given,

Key	Hash(key)
B	3
G	5
I	3
N	5
O	1
P	5
R	4

And it results in the following hash table:

0	1	2	3	4	5	6
P	R	O	B	I	N	G

Please put check mark (√) all that apply.

	B	G	N	O	R
could have been first key inserted					
could have been last key inserted					
must have been inserted before I					
must have been inserted after I					

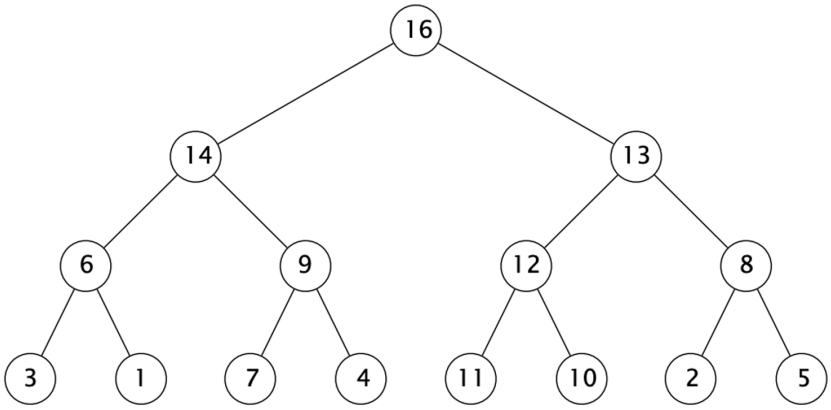
B N
O
B
O R

2. Consider implementing a Deque using a singly linked list, storing the first item in the deque in the first node in the linked list (and the last item in the deque in the last node). Which of the following operations could be implemented to run in constant time in the worst case? Mark all that apply.

    1. `addFirst()`
    2. `addLast()`
    3. `removeFirst()`
    4. `removeLast()`
    5. `isEmpty()`

All operations except removeLast() can be implemented in constant time, by maintaining references to both the first and last nodes in the singly linked list.

3. Consider the following heap.



    a. Which keys in the heap will be involved in a compare operation when we insert 15?

    b. Which keys in the heap will be involved in a compare operation when we delete maximum?

(a) 3 6 14 16

b) 4 5 6 7 9 13 14