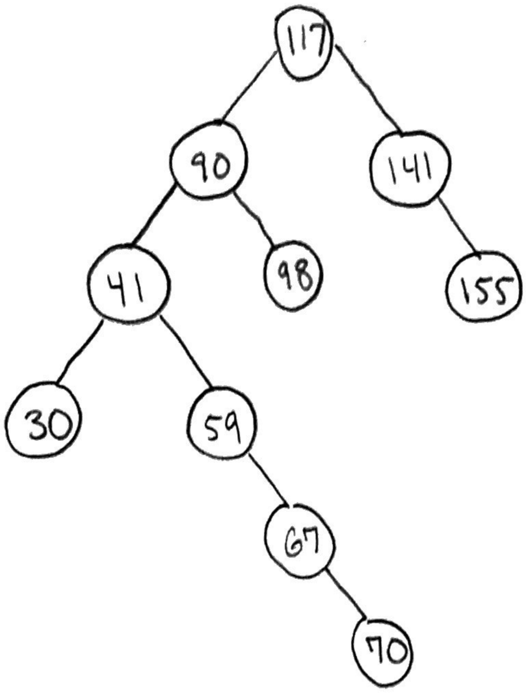
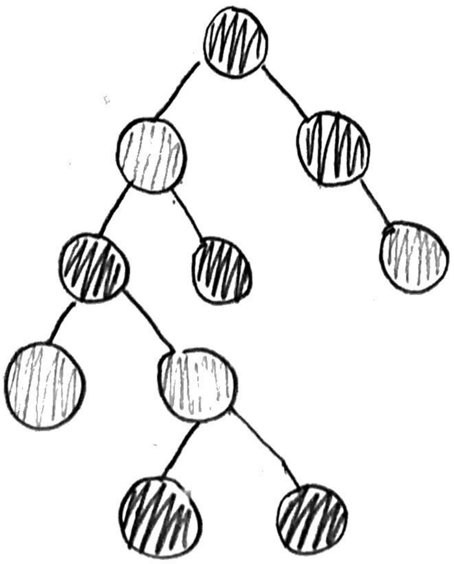
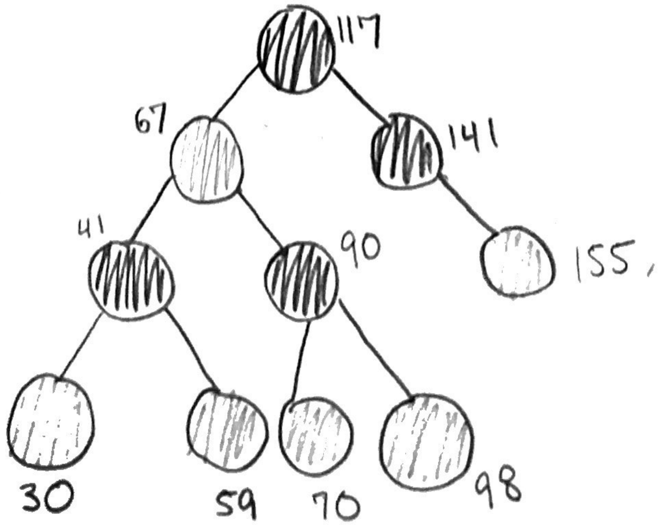
Q3

a)



e)





	unsorted sing	sorted sing	unsorted doub	sorted doub
search	O(n)	O(n)	O(n)	O(n)
insert	O(1)	O(n)	O(1)	O(n)
delete	O(n)	O(n)	O(1)	O(1)
successor	O(1)	O(1)	O(1)	O(1)
predecessor	O(n)	O(n)	O(1)	O(1)
min	O(n)	O(1)	O(n)	O(1)
max	O(n)	O(n)	O(n)	O(1)

* search requires traversing every #, sorted double could be $O(\frac{n}{2})$ max

* inserting unsorted could just put element at end, sorted requires traversal for the correct spot to keep it sorted

* you'd have to traverse the list for node's previous node for single before deleting, you'd have the previous with dll

* finding the next element would be O(1) for any linked list

* you'd need to traverse the sll to find previous element, dll has previous pointer so O(1)

* min/max would be O(1) for sorted but O(n) for unsorted