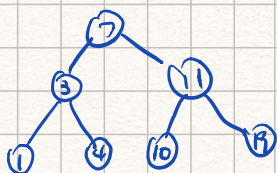


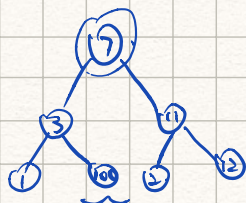
1. a. i. BST for all nodes  $x$ , its left subtree contains nodes smaller than  $x$ .  
 right subtree contains nodes bigger than  $x$ .

ii.



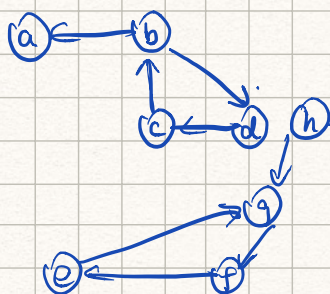
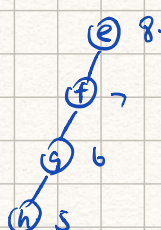
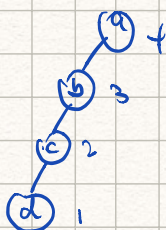
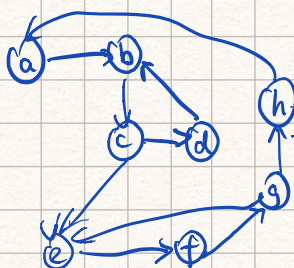
iii.

na

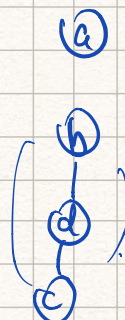


iv. 7, 3, 11, 1, 4, 10, 13.

b.



h



c. Largest Key  $(l, r, x, k)$  ?

if  $(x == \text{nil})$  return  $-\text{INT-MAX}$ ;

return  $\max(\text{LargestKey}(l, r, l(x), k), \text{LargestKey}(l, r, r(x), k), k(x));$

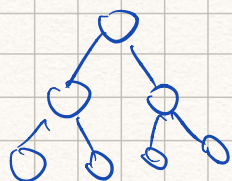
}

d. So every arc is connecting to a new node.

max articulation points

s.t.

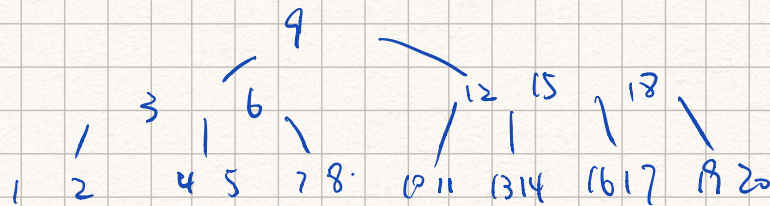
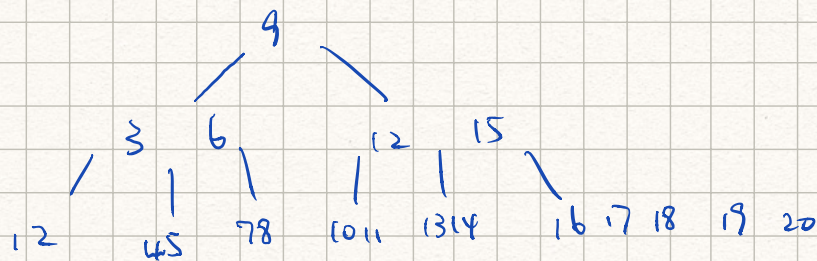
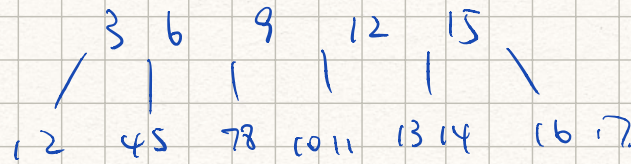
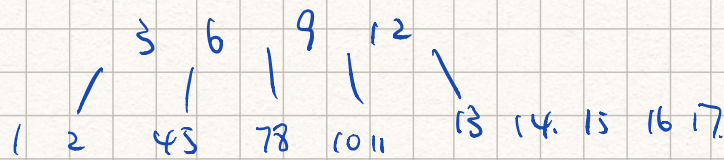
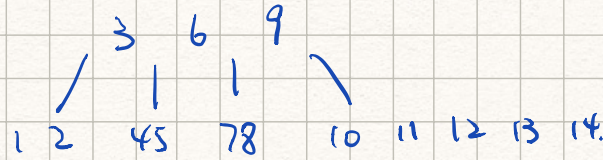
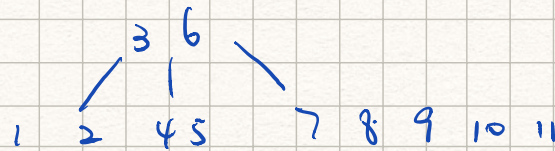
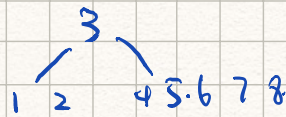
tree





2. 1 -20

order 2 min 2  
none f.



2015 Thany I.

1. f.

for  $i = 1 \sim n$ .  
 $lis[i] = 1$ .

for (the  $j = 0 \sim i$ )

if  $arr[j] > arr[i]$  &&  $lis[i] = lis[j] + 1$ .

$lis[i] = lis[j] + 1$ ;



}

return max\_elem(lis);