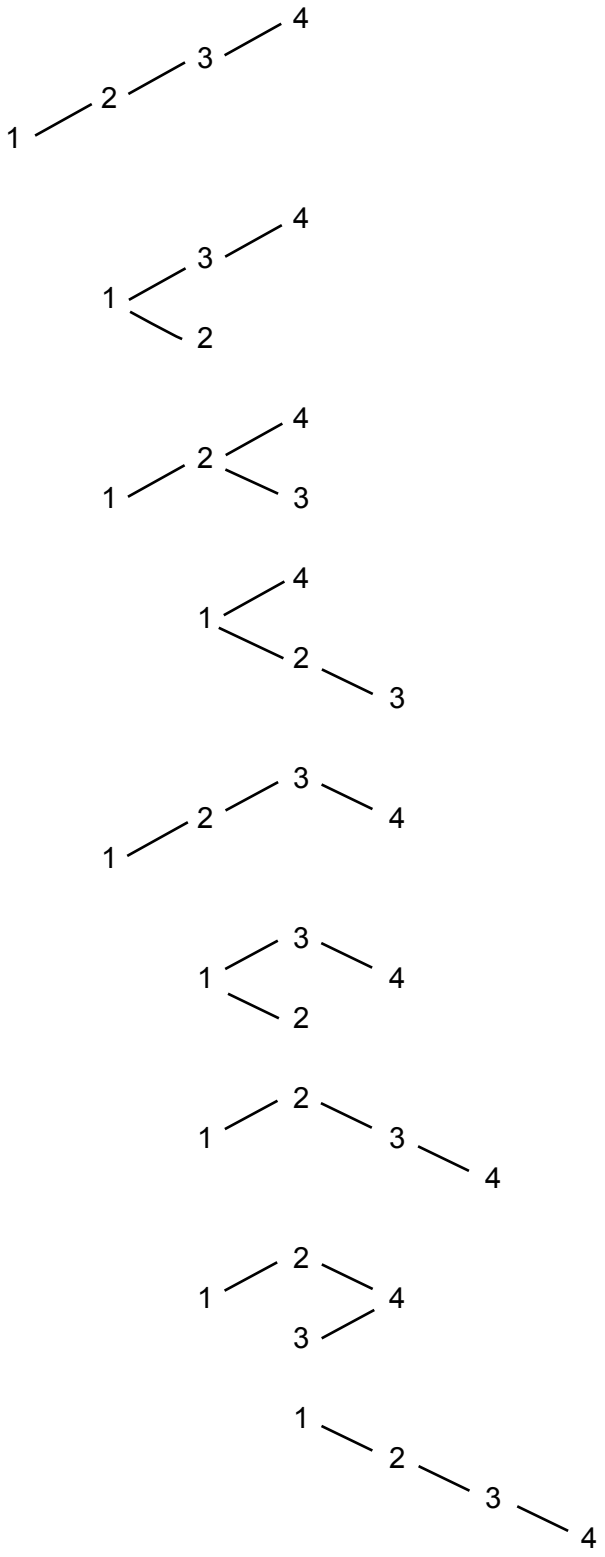
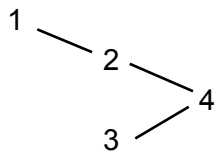
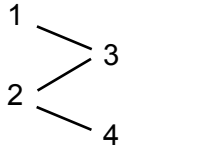
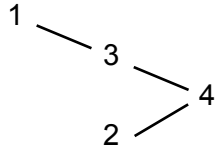
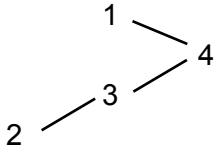
5.1:

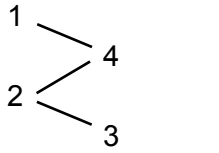




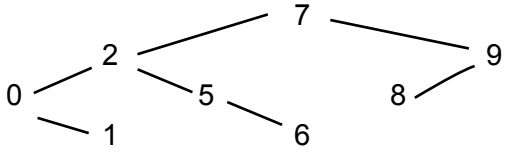




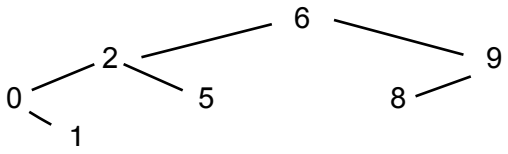




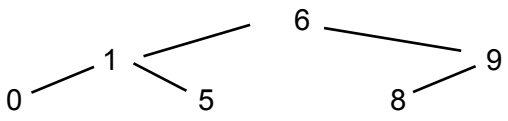
5.2:



5.3:



~~1~~



5.4: Yes.

5:

n = 3 T(3) = 1.50203704834e-05 seconds)
n = 4 T(4) = 1.00135803223e-05 seconds)
n = 5 T(5) = 1.09672546387e-05 seconds)
n = 6 T(6) = 1.50203704834e-05 seconds)
n = 7 T(7) = 1.81198120117e-05 seconds)
n = 8 T(8) = 2.09808349609e-05 seconds)
n = 9 T(9) = 2.28881835938e-05 seconds)
n = 10 T(10) = 2.78949737549e-05 seconds)
n = 11 T(11) = 2.88486480713e-05 seconds)
n = 12 T(12) = 3.69548797607e-05 seconds)
n = 13 T(13) = 4.10079956055e-05 seconds)
n = 14 T(14) = 4.19616699219e-05 seconds)
n = 15 T(15) = 4.6968460083e-05 seconds)

The time cost of function call is increase with the increasing size of input.

6: Time: $O(n \lg n)$, Space: $O(1)$, constant. The time is the same as merge sort. The space is less than merge sort.

7:
```
procedure COMPUTED_VALUE(){
begin:
        a = first()
        val = impressive_A(a.height)
        a.computed_value = val
        a = next()
end
}
```

time complexity: $O(1)$
space complexity: $O(n)$

8:
```
procedure INSERT(val){
a, b : STACK
begin:
        for all n in a
                if n < val
                        b.push(n)
        b.push(n)
        for all n in a
                if n > val
                        b.push(n)
        return b
```

```
end
}

time complexity: O(1), constant
space complexity: O(n)

procedure REMOVE(val){
a, b : STACK
begin:
        for all n in a
                if n! = val
                        b.push(n)
        return b
end
}

time complexity: O(1), constant
space complexity: O(n)
9:      procedure RSYNC(){
n = 1
begin:
        a:= SERVER
        b:= CLIENT
        while b(n , n+511) != null
                if a(n, n+511) != null
                        if a(n, n+511) != b(n, n+511)
                                return n
end
}
```