Ans 1 :

Hints :- must be a ① binary tree. satisfying the ② ordering property..

A2 : The values in the array are not really important for the "shape" of tree.

we have to simulate binsearch over the index range [0, 9).

The tree is recursively constructed as follows :-

Lets say we are searching some element x in array A[lo, hi).

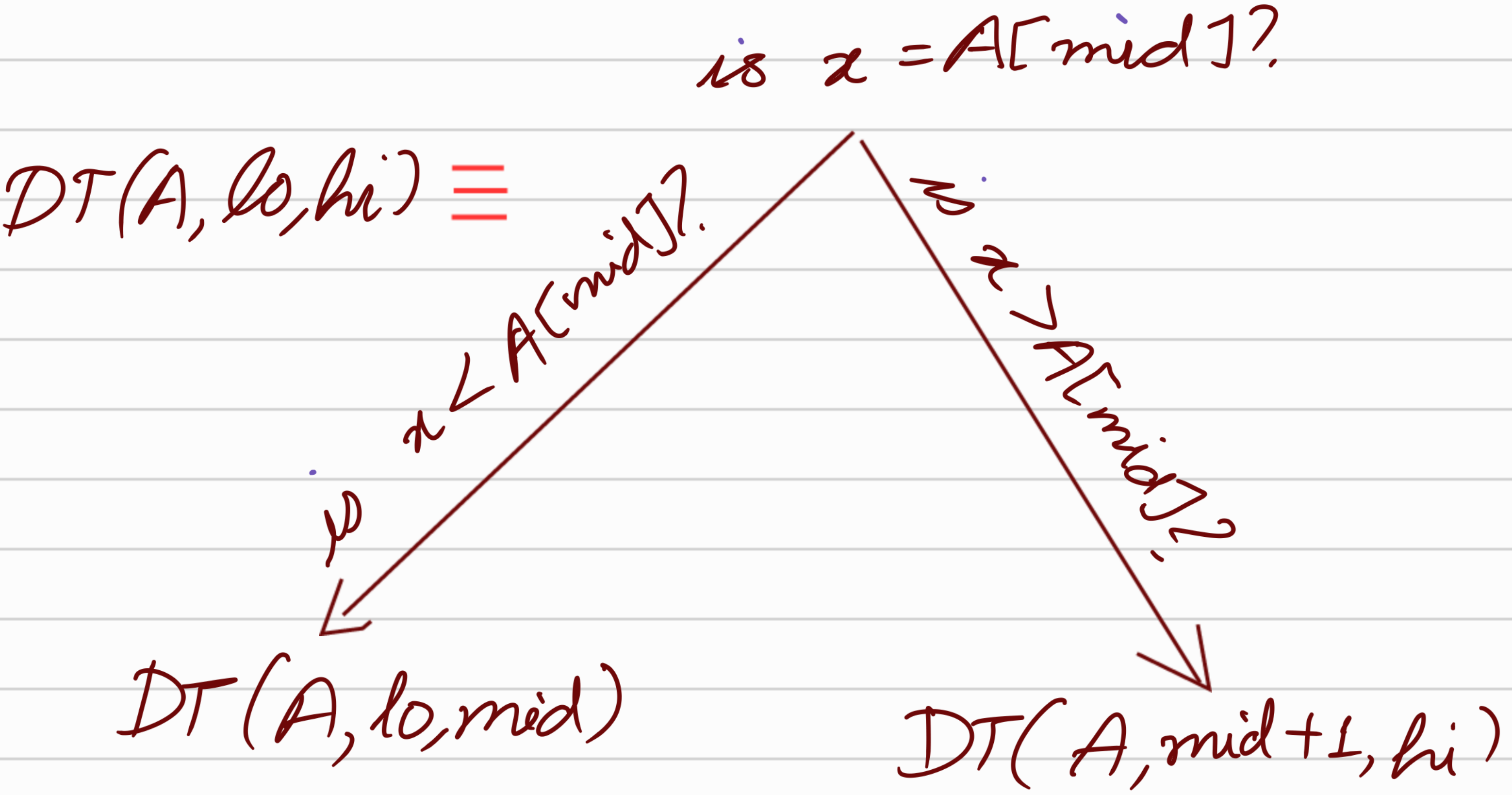
we first compute :

$$mid = \frac{lo + hi}{2}$$

We call the decision tree of searching $A[lo, hi)$ as

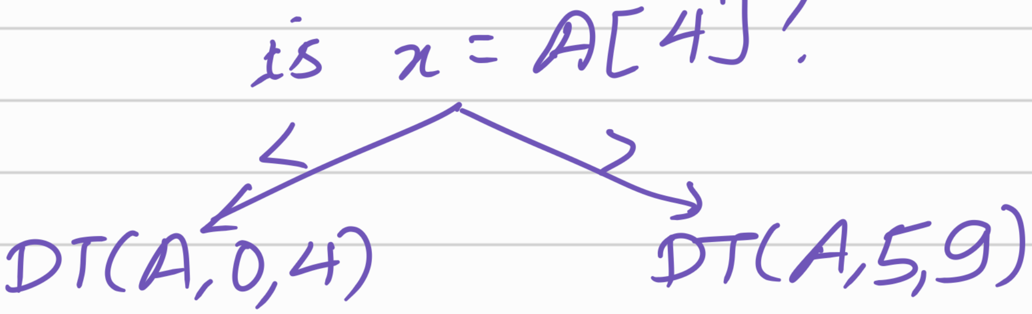
$DT(A, lo, hi)$

we have:-

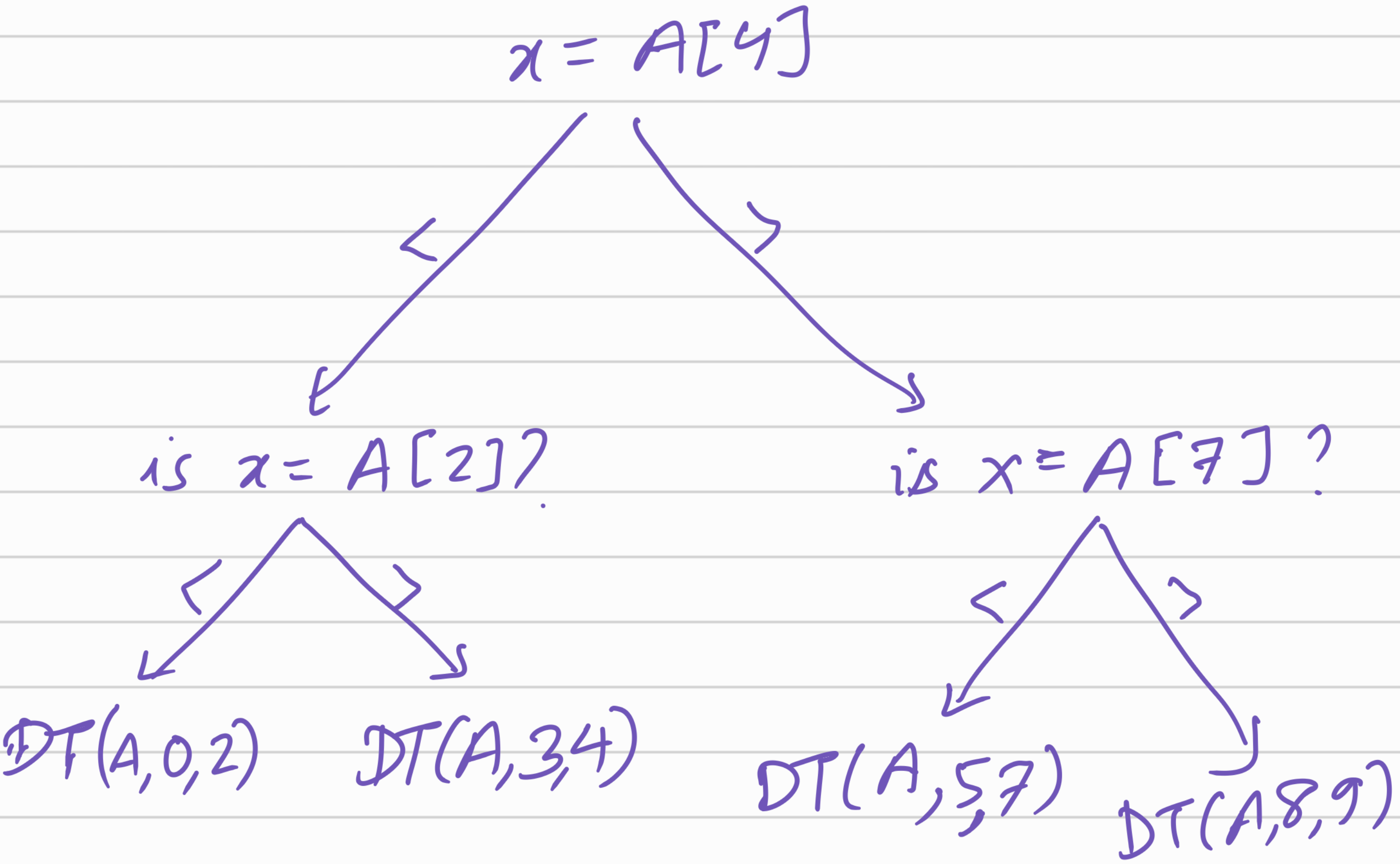


$A = \{1, 2, \ldots, 9\}$

Suppose the array given to us is $A[0, 9)$ {having 9 elements}.



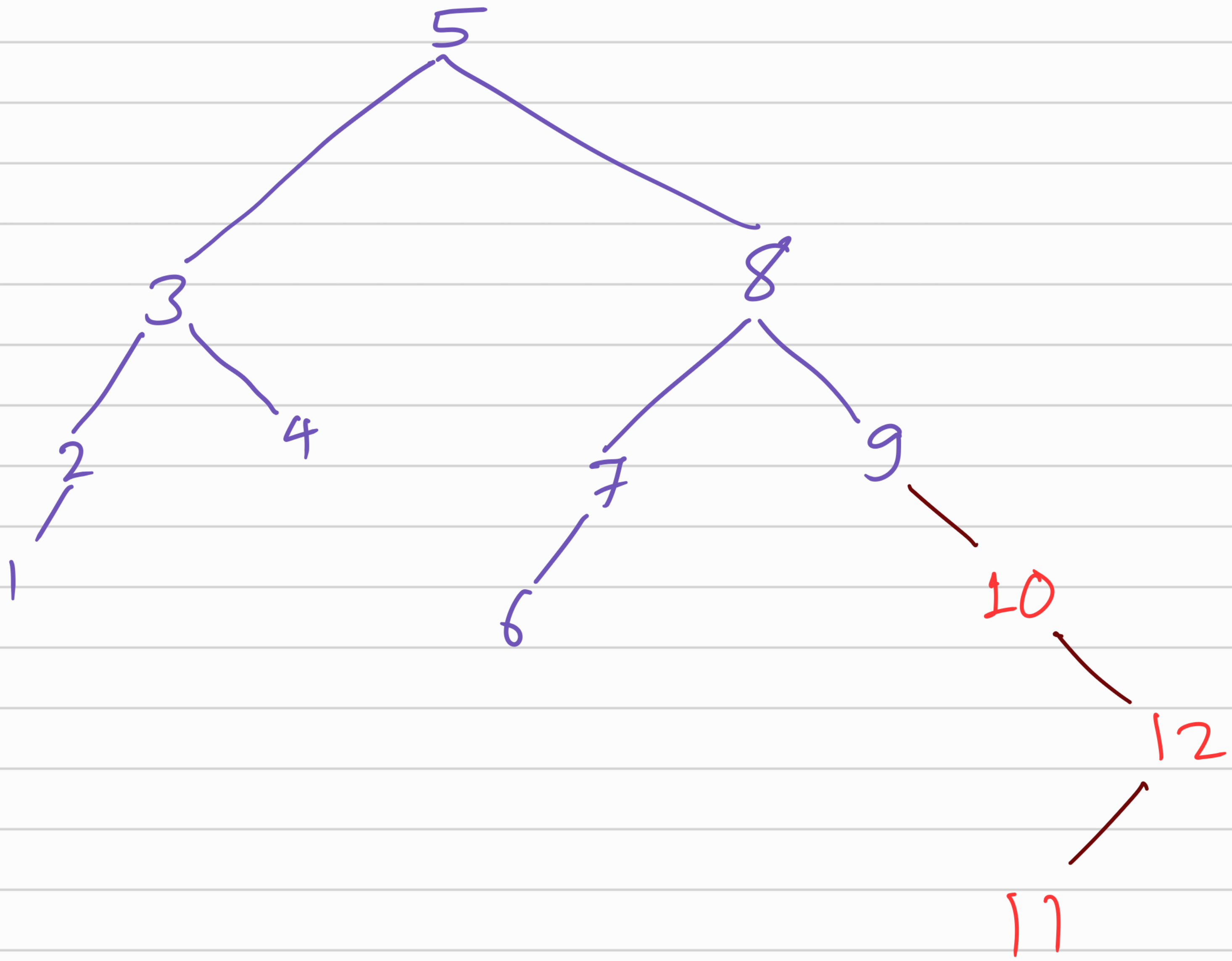
Keep on expanding the tree recursively :-



Keep on expanding to get the full tree.

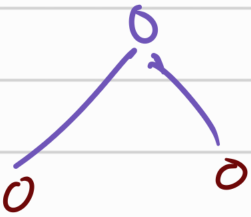
The base case is a leaf of DT(A, i, i+1) for any i.

Ans 3 :- After all the insertions :-



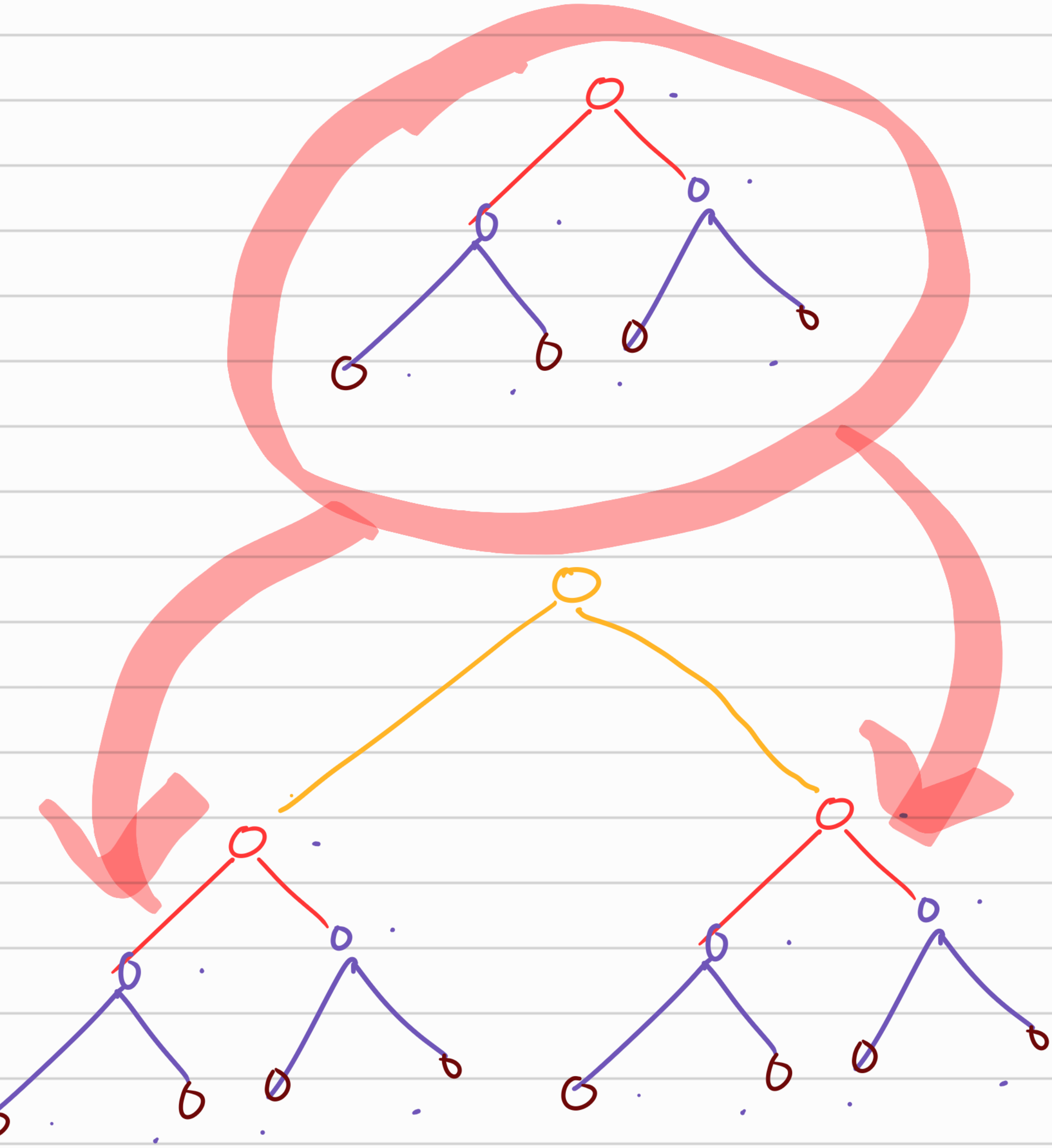
Ans 4:

1 single node

3 

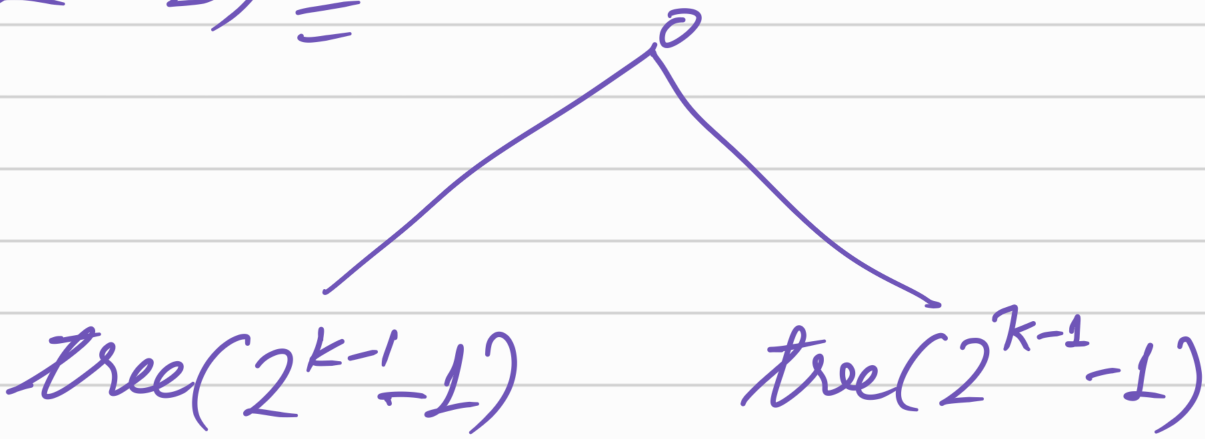
7

15



In general :-

tree $(2^k-1) \equiv$



Ans 5: Hint

Exchange left with right in the code for answer. Justify the soln. using ordering property of recursive structure of the tree.