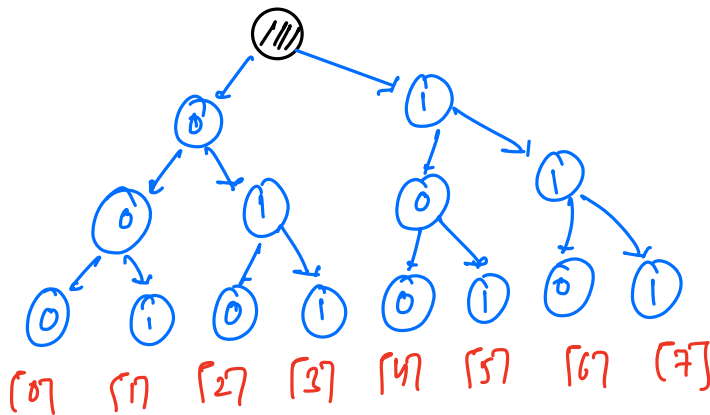


Trie → data-structure
 ↳ prefix-tree

Trie on bit representation

M.S.B
 ↓
 L.S.B



0-7.

| | |
|-----|-----|
| 0 → | 000 |
| 1 → | 001 |
| 2 → | 010 |
| 3 → | 011 |
| 4 → | 100 |
| 5 → | 101 |
| 6 → | 110 |
| 7 → | 111 |

[2³] 8 no's → height = 3

[2⁴] 16 no's → height = 4

[2⁵] 32 no's → height = 5

N no's → height = $\log_2 N$.

[3, 6, 14, 12]

3 → 0010

6 → 0110

14 → 1110

12 → 1010

```

class Node {
    int val;
    Node children[2];
    ↙       ↘
children[0] children[1]
  
```

Q1) Max value of XOR pairs. [A[i] ^ A[j] is maximum]

$$A \rightarrow [9 \ 8 \ 10 \ 7]$$

0 1 2 3

| A | B | A^B |
|---|---|-----|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

idea-1. Consider all the pairs.

$$\begin{array}{r} 9 \rightarrow 1001 \\ 8 \rightarrow 1000 \\ \hline 0001 \end{array}$$

$$\begin{array}{r} 9 \rightarrow 1001 \\ 10 \rightarrow 1010 \\ \hline 0011 \end{array}$$

$$\begin{array}{r} 9 \rightarrow 1001 \\ 7 \rightarrow 0111 \\ \hline 1110 \end{array}$$

$$\begin{array}{r} 8 \rightarrow 1000 \\ 10 \rightarrow 1010 \\ \hline 0010 \end{array}$$

$$\begin{array}{r} 8 \rightarrow 1000 \\ 7 \rightarrow 0111 \\ \hline 1111 \end{array}$$

$$\begin{array}{r} 10 \rightarrow 1010 \\ 7 \rightarrow 0111 \\ \hline 1101 \end{array}$$

Ans = 15

$$\begin{array}{r} A \rightarrow 1011101 \\ B \rightarrow 0100010 \\ \hline \text{XOR} \rightarrow 1111111 \end{array}$$

$$\begin{array}{r} A \rightarrow 1011101 \\ B \rightarrow 1100010 \\ \hline 0111111 \\ \uparrow \uparrow \uparrow \uparrow \uparrow \uparrow \\ 2^5 + 2^4 + 2^3 + 2^2 + 2^1 + 2^0 = 63 \end{array}$$

$$\begin{array}{r} A \rightarrow 1011101 \\ B \rightarrow 0011101 \\ \hline 1000000 \\ \uparrow \\ 2^6 \rightarrow 64 \end{array}$$

$$2^i > 2^0 + 2^1 + 2^2 + \dots + 2^{i-1}$$

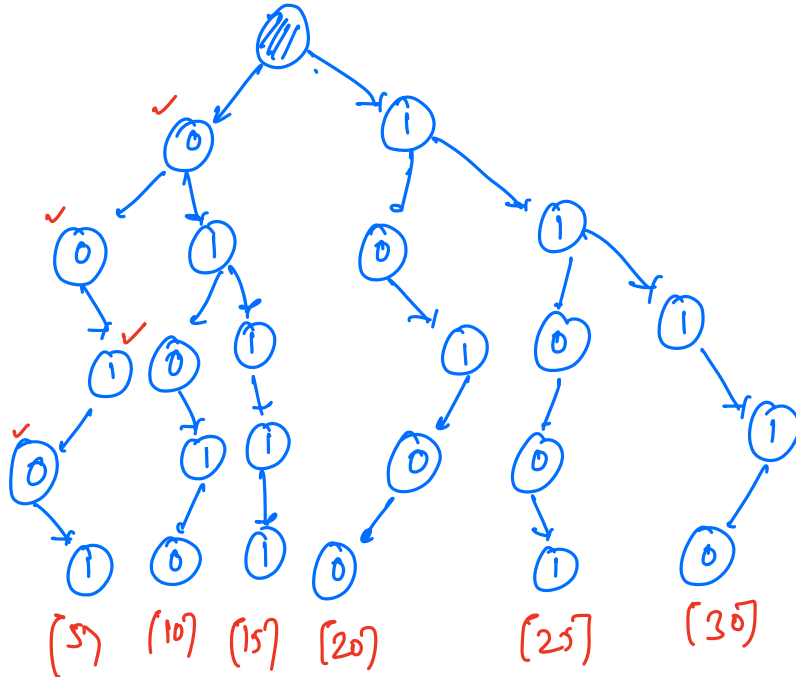
\therefore M.S.B matters

arr[] → [5, 20, 15, 10, 25, 30]

↓ ↓ ↓ ↓ ↓ ↓

Xor value → [28, 30, 27, 30, 28, 27]

xor value $\rightarrow (28, 30, 27, 30, 28, 27)$



$5 \rightarrow 00101$
 $20 \rightarrow 10100$
 $15 \rightarrow 01111$
 $10 \rightarrow 01010$
 $25 \rightarrow 11001$
 $30 \rightarrow 11110$

$S \rightarrow 00101$

demand \rightarrow $\begin{array}{cccc} 1 & 1 & 0 & 1 & 0 \\ \checkmark & \checkmark & \checkmark & \times & \times \\ \hline 1 & 1 & 1 & 0 & 0 \end{array}$

20 → 10100

$$\begin{array}{r} \text{demand} \rightarrow \begin{array}{cccc} 0 & 1 & 0 & 1 \\ \checkmark & \checkmark & \checkmark & \checkmark \\ \hline 1 & 1 & 1 & 0 \end{array} \end{array}$$

15 \rightarrow 01111

$\begin{array}{cccc} & 1 & 0 & 0 & 0 & 0 \\ & \checkmark & \checkmark & \times & \checkmark & \checkmark \\ \text{NOT} \rightarrow & 1 & 1 & 0 & 1 & 1 \end{array}$

10 → 01010

$\text{dmd} \rightarrow \begin{array}{cccc} 1 & 0 & 0 & 1 \\ \checkmark & \checkmark & \checkmark & \times \end{array}$
 $\text{xor} \rightarrow \begin{array}{cccc} 1 & 1 & 1 & 0 \end{array}$

25 → 11001

$\text{And} \rightarrow \begin{matrix} 0 & 0 & 1 & 1 & 0 \\ \vee & \vee & \vee & \times & \times \end{matrix}$
 $\text{Xor} \rightarrow \begin{matrix} 1 & 1 & 1 & 0 & 0 \end{matrix}$

#code ->

→ Find max element in the given arr[]

$x \rightarrow$ no. of bits in max-element, $ans = 0$

Insert all the numbers in trie.

```
for(i=0; i<N; i++){
```

```
    xor = 0, Node curr = root;
```

```
    for(j=x-1; j>=0; j--){
```

```
        if( checkBit(arr[i], j) == true){
```

```
            if(curr.children[0] != NULL){
```

```
                //set jth bit in xor
```

```
                xor = ((1<<j) | xor);
```

```
                curr = curr.children[0];
```

```
            }  
            else{
```

```
                curr = curr.children[1];
```

```
            }  
        }  
    }  
    else{
```

```
        if(curr.children[1] != NULL){
```

```
            //set jth bit in xor
```

```
            xor = ((1<<j) | xor);
```

```
            curr = curr.children[1];
```

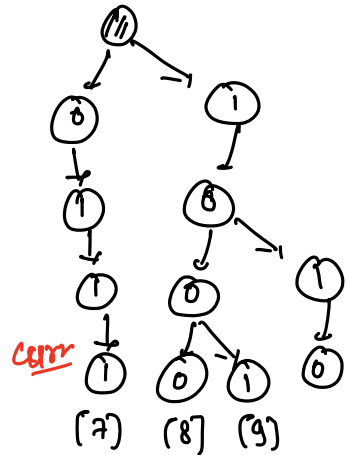
```
        }  
        else{
```

```
            curr = curr.children[0];
```

```
        }  
    }  
    ans = Max(ans, xor);
```

```
return ans;
```

[9, 8, 10, 7]



$x = 4$

$ans = 0$

$i = 0$

$9 \rightarrow 1001$

$xor \rightarrow \cancel{0}\cancel{0}\cancel{0}0$
 $ \phantom{\cancel{0}}\phantom{\cancel{0}}\phantom{\cancel{0}}1$

$j = \cancel{3}\cancel{2}\cancel{1}0$
 $ \phantom{\cancel{3}}\phantom{\cancel{2}}\phantom{\cancel{1}}-1$

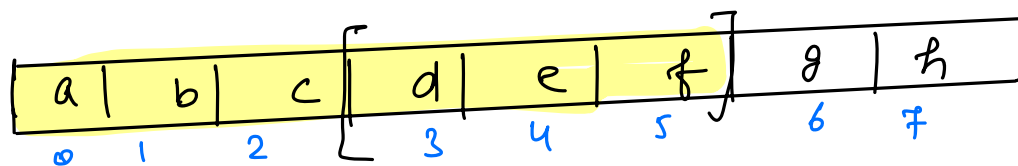
T.C $\rightarrow O(N * x)$
S.C $\rightarrow O(N * x)$

Maximum Subarray XOR

arr = [1, 4, 3]
 0 1 2

(1) → 1
[1, 4] → 5
[1, 4, 3] → 6
[4] → 4
[4, 3] → 7 → ans
[3] → 3

Idea-1 → Consider all the subarrays. T.C → $O(N^2)$



$$\underline{a \wedge b \wedge c \wedge d \wedge e \wedge f}$$
$$\wedge (a \wedge b \wedge c)$$

$$\text{Xor of subarray } [3, 5] = \text{pXor}[5] \wedge \text{pXor}[2]$$

$$\text{Xor of subarray } [i, j] = \text{pXor}[j] \wedge \underline{\text{pXor}[i-1]}$$

idea-2

arr = [1, 4, 3]

1, 4, 3, 5, 6

pXor = [1, 5, 6]

ans = 7

$$\begin{array}{r} 1 \wedge 5 \\ \hline 4 \end{array}$$

$$\begin{array}{r} 1 \wedge 6 \\ \hline 7 \end{array}$$

$$\begin{array}{r} 5 \wedge 6 \\ \hline 3 \end{array}$$

#code:-

ans = 0

```
for (i = 0; i < N; i++) {  
    ans = Max(arr[i], ans)  
}
```

pXor[N];

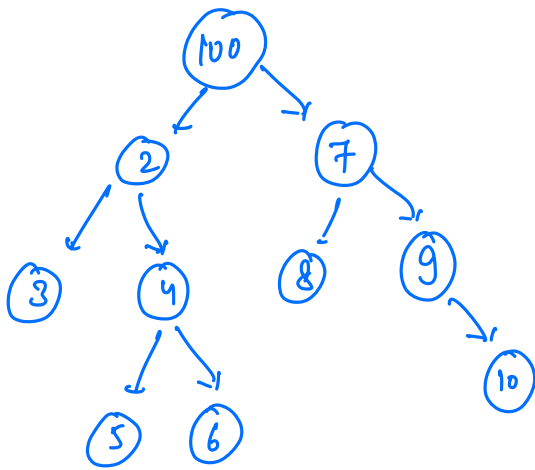
pXor[0] = arr[0];

```
for (i = 1; i < N; i++) {  
    pXor[i] = pXor[i-1] ^ arr[i];  
    ans = Max(ans, pXor[i]);  
}
```

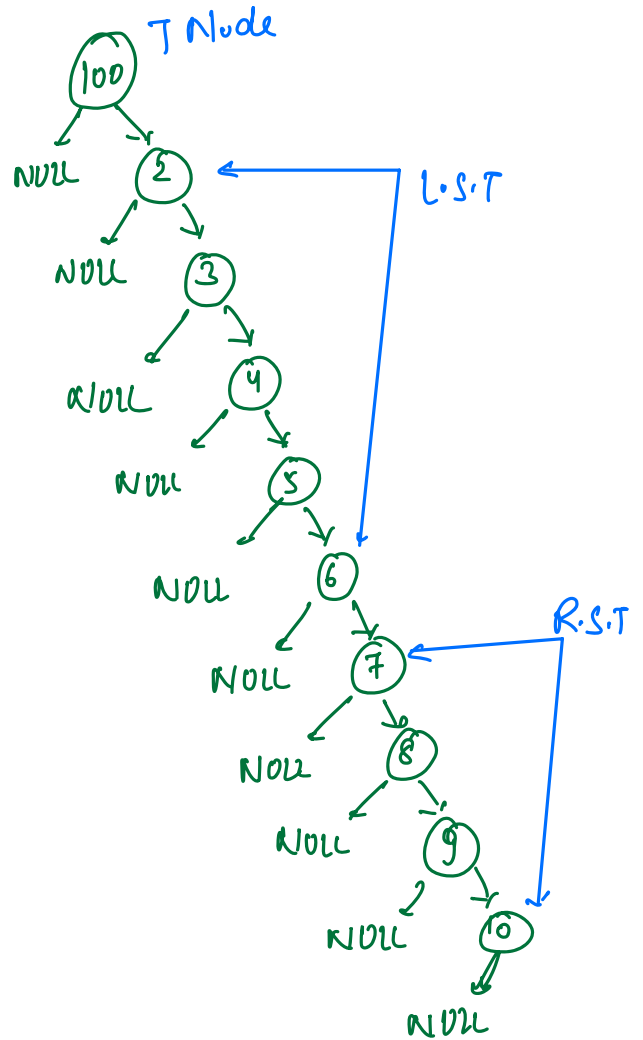
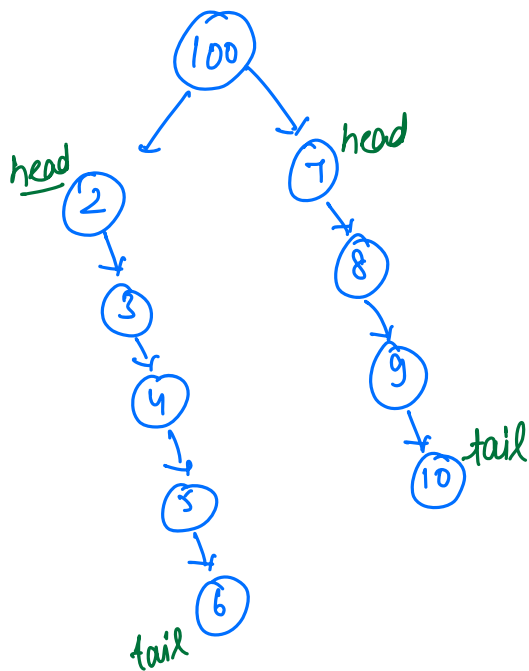
T.C $\rightarrow O(N \cdot \log N)$
S.C $\rightarrow O(N+2)$

- find maximum xor value pair in the pXor[]
(similar to prev question)

Q1 Convert the given binary tree to linked-list in pre-order manner.



update
pointers



```
pair {  
    Node head;  
    Node tail;  
}
```

#code. →

```
pair flatten ( Node root) {
```

```
    if (root == NULL) { return new Pair{NULL, NULL}; }
```

```
    pair lp = flatten(root.left);
```

```
    pair rp = flatten(root.right);
```

```
    if (lp.head == NULL && rp.head == NULL) {
```

```
        {
            return { root, root };
        }
```

```
    else if ( lp.head == NULL ) {
```

```
        {
            return ( root, rp.tail );
        }
```

```
    else if ( rp.head == NULL ) {
```

```
        {
            root.right = lp.head;
            return ( root, lp.tail );
        }
```

```
    else {
```

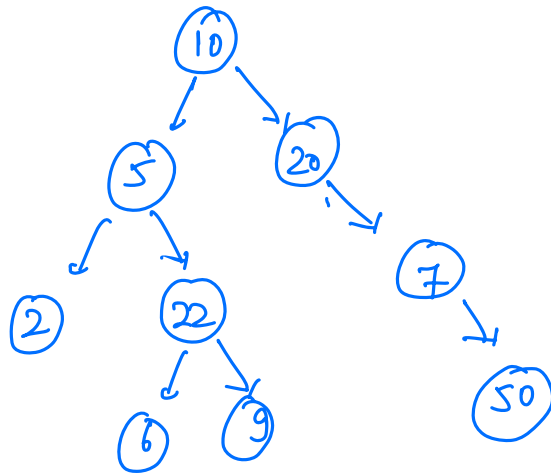
```
        {
            root.left = NULL;
            lp.tail.right = root.right;
            root.right = lp.head;
            return ( root, rp.tail );
        }
```

$\left[\begin{array}{l} T.C \rightarrow O(N) \\ S.C \rightarrow O(H) \end{array} \right]$

⇒ dry-run #todo

Q: Given a B.S.T where exactly 2 nodes are swapped.

Find the two nodes. [All nodes are distinct]

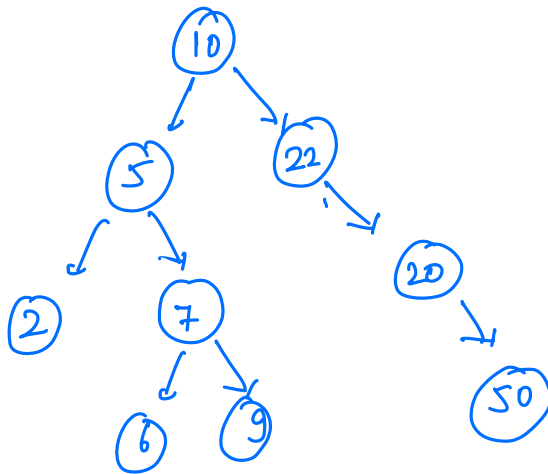


In-Order $\rightarrow [2, 5, 6, 22, 9, 10, 20, 7, 50]$

2 dips.

1st dip \rightarrow 1st element

2nd dip \rightarrow 2nd element

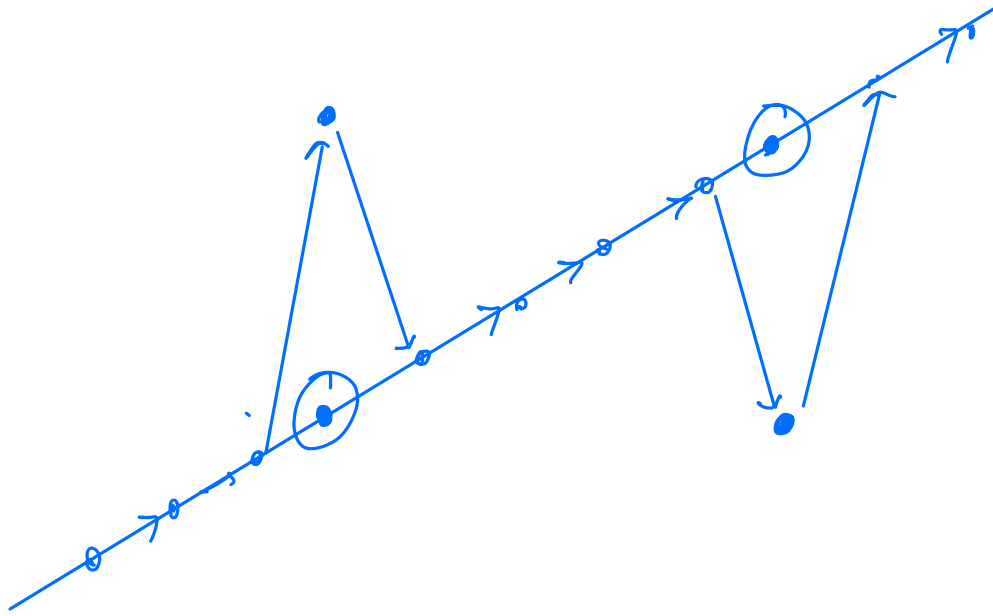


In-Order $\rightarrow [2, 5, 6, 7, 9, 10, 22, 20, 50]$

1 dip.

Consider both the elements from that dip.

T.C $\rightarrow O(N)$
S.C $\rightarrow O(N) \rightarrow O(H+)$



in first dip - greater element is swapped element.
 in second dip - smaller element is swapped element.

```
void traversal (Node root) {
```

```
    // base condn
```

```
    traversal (root->left)
```

```
    traversal (root->right)
```

```
    ?
```

B.M + Trie

$\left[\begin{array}{l} \rightarrow \text{Auto-complete} \\ \rightarrow \text{Doubts on trees.} \end{array} \right] \underline{\text{P.S.S.}}$