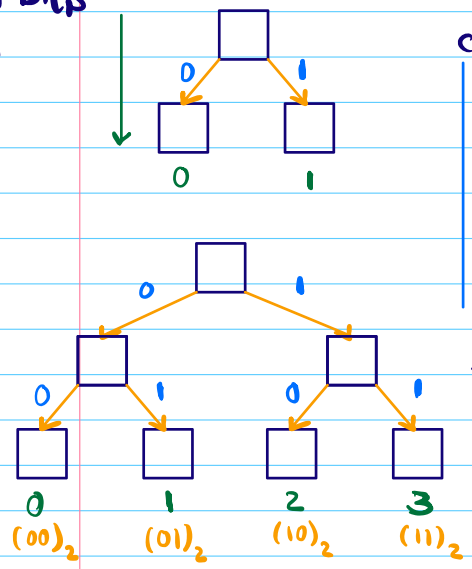
Topics - Tries of Bits
- max XOR Pair
- max XOR subarray
- BT → list (to)

$0 \rightarrow (0)_2$ $1 \rightarrow (1)_2$ $3 \rightarrow (11)_2$

Tries of Bits

0 - 1




```
class Node{
  int data
  Node bit[]
  Node(){
    bit = new Node[2]
  }
}
```

max number → n

$n = 2^H - 1$

1-b

max $(i^{b} \wedge x) = 1$

0 1

max $(0^{b} \wedge x) = 1$

0 1

bit manipulation

$2n = n \ll 1$

$n \gg 1 = n/2$

3210
1010

$((x \gg 2) \& 1)$

$((x \gg i) \& 1)$ → give me bit i from X

P1 Given an array of integers A, find the max value of $A[i] \wedge A[j]$, for all possible i,j $i \neq j$ $x \wedge y == y \wedge x$ $A[i] \wedge A[i]$

ex $A = \{9, 8, 10, 7\}$ (0, 1, 2, 3)

1001 1000 1010 0111

$9 \wedge 8 = 0001$

$7 \wedge 8 = 1111 \rightarrow 15$

bruteforce i, j TC: $O(n^2)$ SC: $O(1)$

$\begin{cases} X \wedge X = 0 & i \neq j \\ X \wedge Y = Y \wedge X & j > i \end{cases}$

```
ans = 0
for i = 0 → n-1        (n-2)
  for j = i+1 → n-1
    ans = max(ans, a[i]^a[j])
```

most significant bit

Note

MSB is important!

① 1000 > 0111 ②

intuition & Sol. in 3 step
- Solve manually
- Solve with trie
- Code

{9, 7, 5, 11, 8}

ex

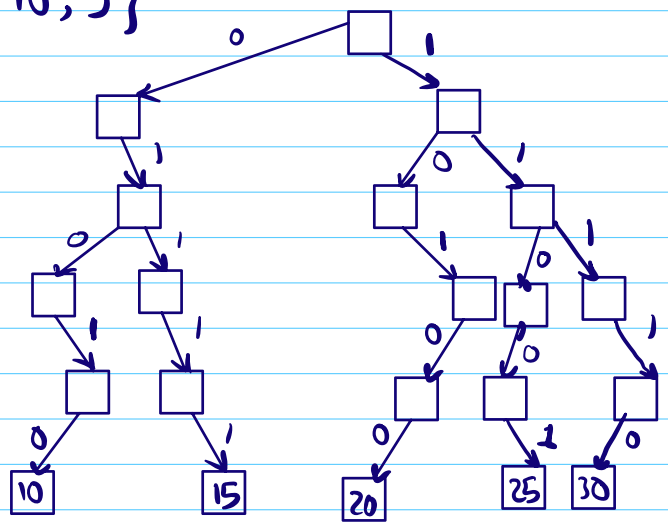
	3	2	1	0
9	1	0	0	1
7	0	1	1	1
5	0	1	0	1
11	1	0	1	1
8	1	0	0	0
	1	1	1	1

{7, 5}

{9, 11, 8}

ex A = {20, 30, 15, 25, 10, 5}

max and		4	3	2	1	0
0	20	1	0	1	0	0
10	30	1	1	1	1	0
27	15	0	1	1	1	1
27 (22)	25	1	1	0	0	1
30	10	0	1	0	1	0
30 (28)	5	0	0	1	0	1



Code

```
void insert(root, X){
    cur = root
    for(i=4; i>=0; i--){
        b = ((X>>i)&1)
        if(cur.bit[b]==null){
            cur.bit[b] = new Node();
        }
        cur = cur.bit[b]
    }
    cur.data = X
}
```

O(b) ~ O(1)

```
int search(root, X){
    for(i=4; i>=0; i--){
        b = ((X>>i)&1)
        t = 1-b
        if(cur.bit[t] != null)
            cur = cur.bit[t]
        else
            cur = cur.bit[b]
    }
    ret X ^ cur.data
}
```

ret max XOR in trie so far

a[i]

O(b) ~ O(1)

fortunate

unfortunate

```
main(){
    ans = a[0] ^ a[1]
    insert(root, a[0])
    insert(root, a[1])
    for(i=2; i<n; i++){
        ans = Max(ans, search(root, a[i]))
        insert(root, a[i])
    }
    ret ans
}
```

O(n)

TC: O(n)

SC: O(b x n) ~ O(n)

bit man

P2 Given an integer array, find max subarray XOR of the given array.

ex $A = \{1, 6, 2\}$

1 0 → 1,6 → $1 \wedge 6 = 7$ → ans

6 0 → 6,2 → $2 \wedge 6 = 4$

2 0 → 1,6,2 → $1 \wedge 6 \wedge 2 = 5$

SC: O(1)

TC: $O(n^3)$

```
for i=0 → n
  for j=i+1 → n
    XOR(a[i], ..., a[j])  O(n)
```

---

prefix sum

$P[i] = a[i] + P[i-1]$ ①

$\forall i, j$, $sum(a[i], \dots, a[j]) = \begin{cases} P[j] - P[i-1] \\ \text{if } i=0 \rightarrow sum = P[j] \end{cases}$ ②

def. prefix XOR

$a[i] \wedge a[i-1] \wedge \dots a[0]$

$P[i] = a[i] \wedge P[i-1]$

$XOR(a[i] \wedge \dots \wedge a[j]) =$

$a[i] \wedge a[i+1] \wedge \dots \wedge a[j] \wedge$
$a[0] \wedge a[1] \wedge a[2] \wedge \dots \wedge a[i-1] \wedge$
$a[0] \wedge a[1] \wedge a[2] \wedge \dots \wedge a[i-1]$

$= P[j] \wedge P[i-1]$ → $i=0$? ⚠

① $X \wedge X = 0$

② $0 \wedge Y = X \wedge X \wedge Y = Y$

$XOR(a[i] \dots a[j]) = P[j] \wedge P[i-1]$

edge case $i=0$

maximize == maximize

1- Calculate PXOR TC

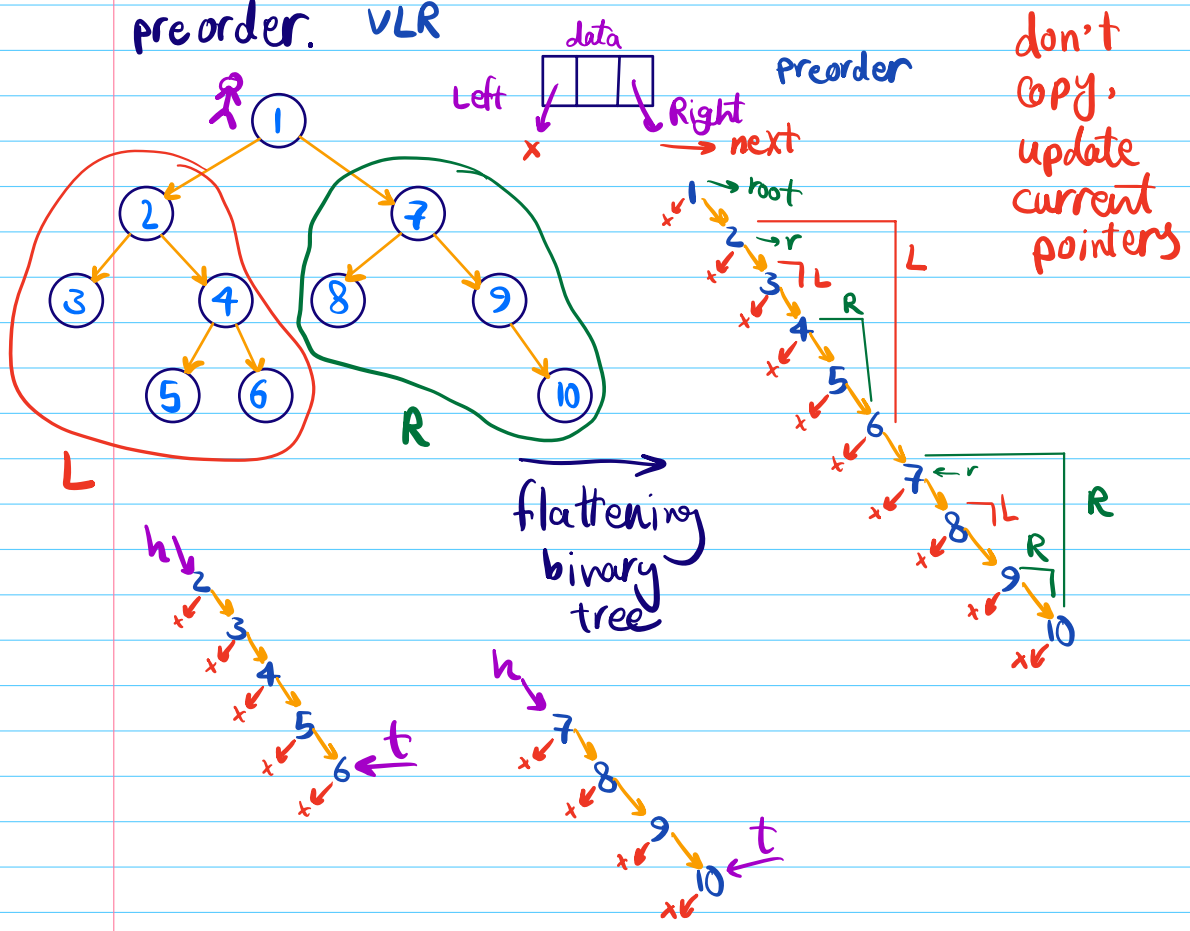
2- find $P[i \wedge \dots \wedge j]$ SC

PSXOR → same nest loop

TC: $O(n + n^2) \sim O(n^2)$ SC: $O(n)$ codesassignment

95% similar PS max sub array

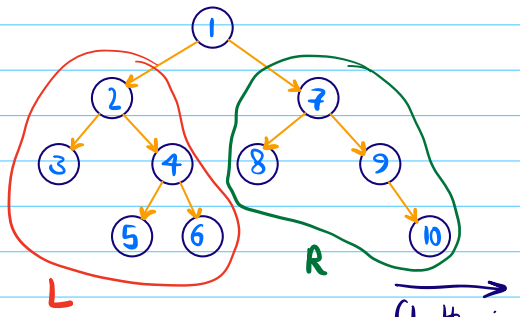
P3 Convert a given binary tree to linked list in preorder. VLR



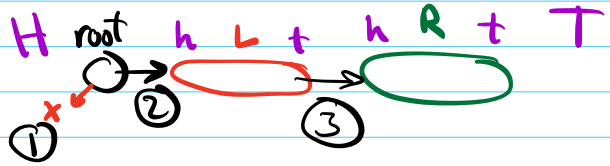
```
(head,tail) flatten(root){                                   Pair <head,tail>
    if(root==null) ret (head:null, tail:null)
    // root
Pair ← L = flatten(root.left)        L
       R = flatten(root.right)       R
    root.left = null ①
    if(L.head==null && R.head==null){
①     ret (root, root)
    }
    else if(L.head==null){
②     root.right = R.head
      ret (root, R.tail)
    }
    else if(R.head==null){
③     root.right = L.head
      ret (root, L.tail)
    }
    else{
④     root.right = L.head ②
      L.tail.right = R.head ③
      ret (root, R.tail)
            head      tail
    }
}
```

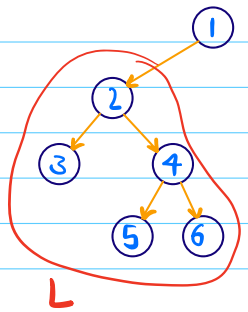
4 different cases

	L.head	R.head	
isnull?	T	T	①
	T	F	②
	F	T	③
	F	F	④



flattening binary tree





TC: O(n)

SC: O(height)