Saturday, 18 December 2021 8:52 PM

Q. Subarray with k different integers.

array , (k)

return → no. of good subarrays.

good array → no. of different integers → (k).

[1, 2, 3, 1, 2] ,

(3) → [1, 2, 3]

[1, 2, 1, 2, 3] k = 2

ans → (7) ✓

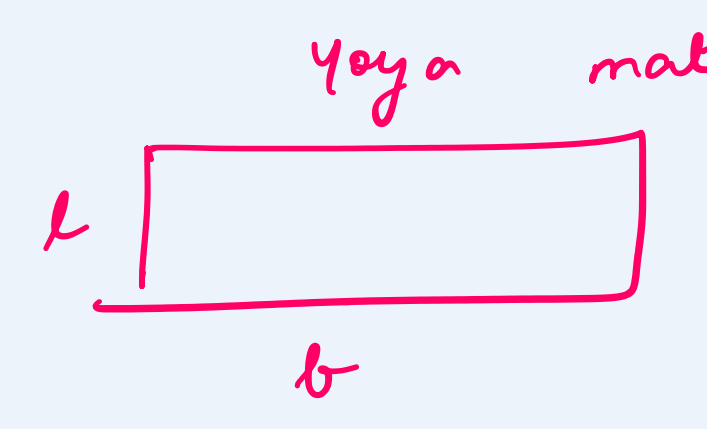
[1, 2] , [1, 2, 1]
[2, 1]   [2, 1, 2]
[1, 2]   [1, 2, 1, 2]
[2, 3]

✓ atmost K distinct ele.

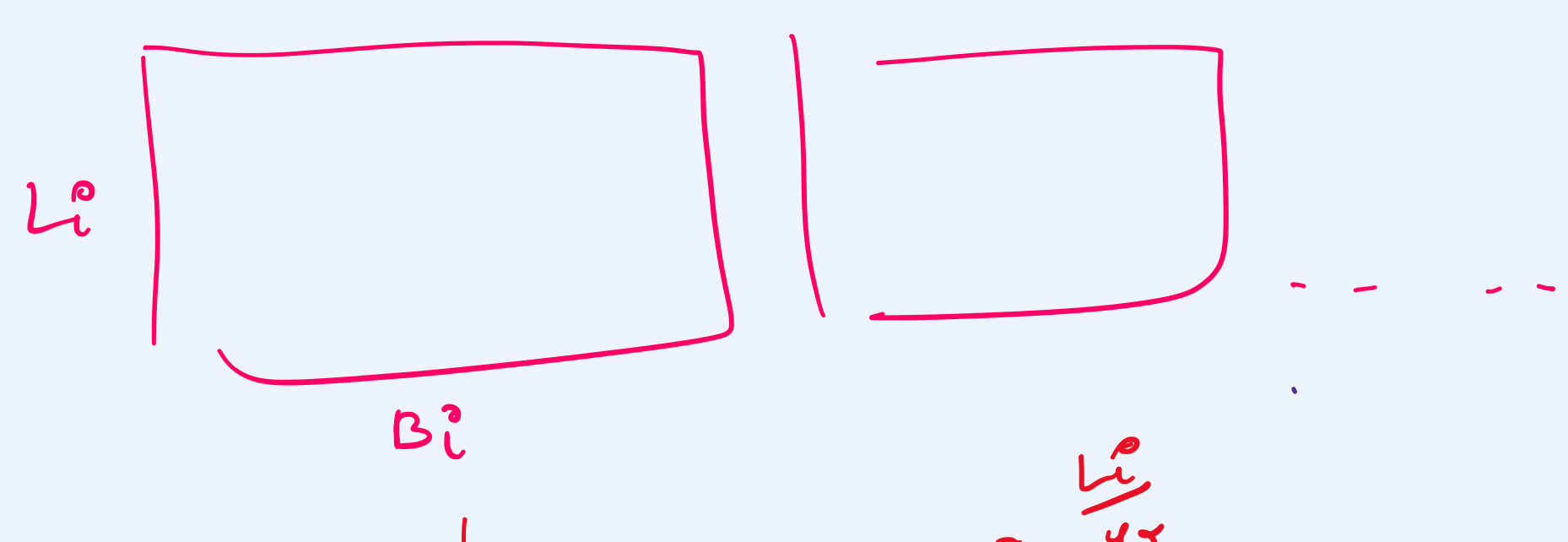
exactly (K) = atmost (K) − atmost (K−1)

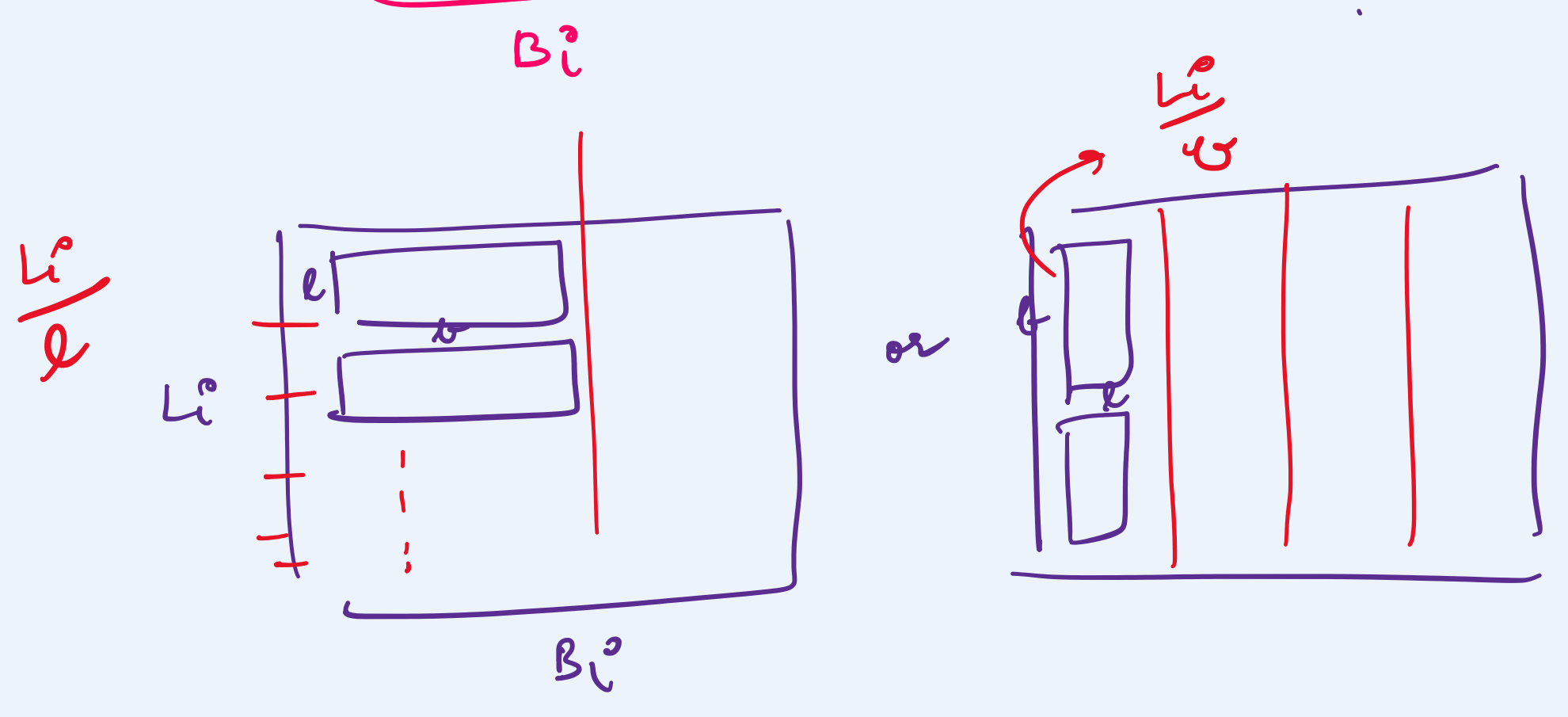
```
subarray with k distinct (A, k) {
    res = 0;
    prefix = 0;
    int [] m = new int [A.length+1];
    for ( int i = 0, j = 0, cnt = 0; i < A.length; i++) {
        if m[A[i]]++ == 0) ++ cnt;
        if (cnt > k) {
            -- m[A[j++]];
            -- cnt;
            prefix = 0;
        }
        while (m [A[j]] > 1) {
            ++ prefix;
            -- m[A[j++]];
        }
        if (cnt == k)
            res += prefix + 1;
    }
    return res;
}
```

Q. Ramdev



N yoga halls



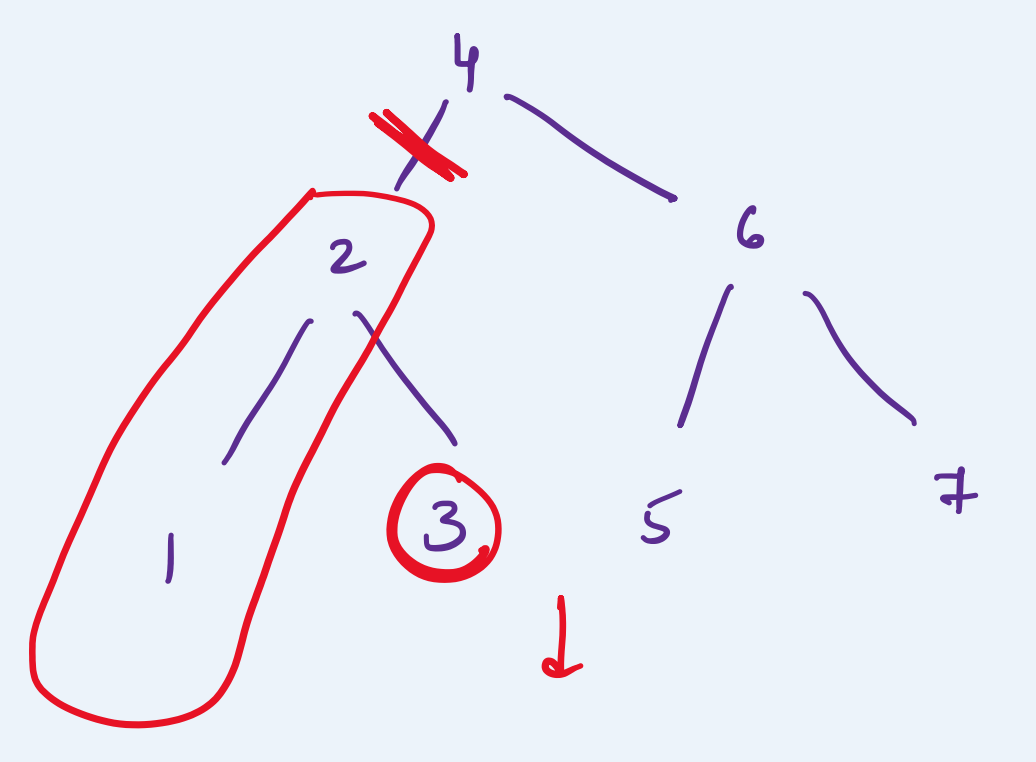


$$\sum_{i=0}^{n-1} \left(\frac{Li}{l} \times \frac{Bi}{b} \times \frac{Li}{b} \times \frac{Bi}{l} \right)$$

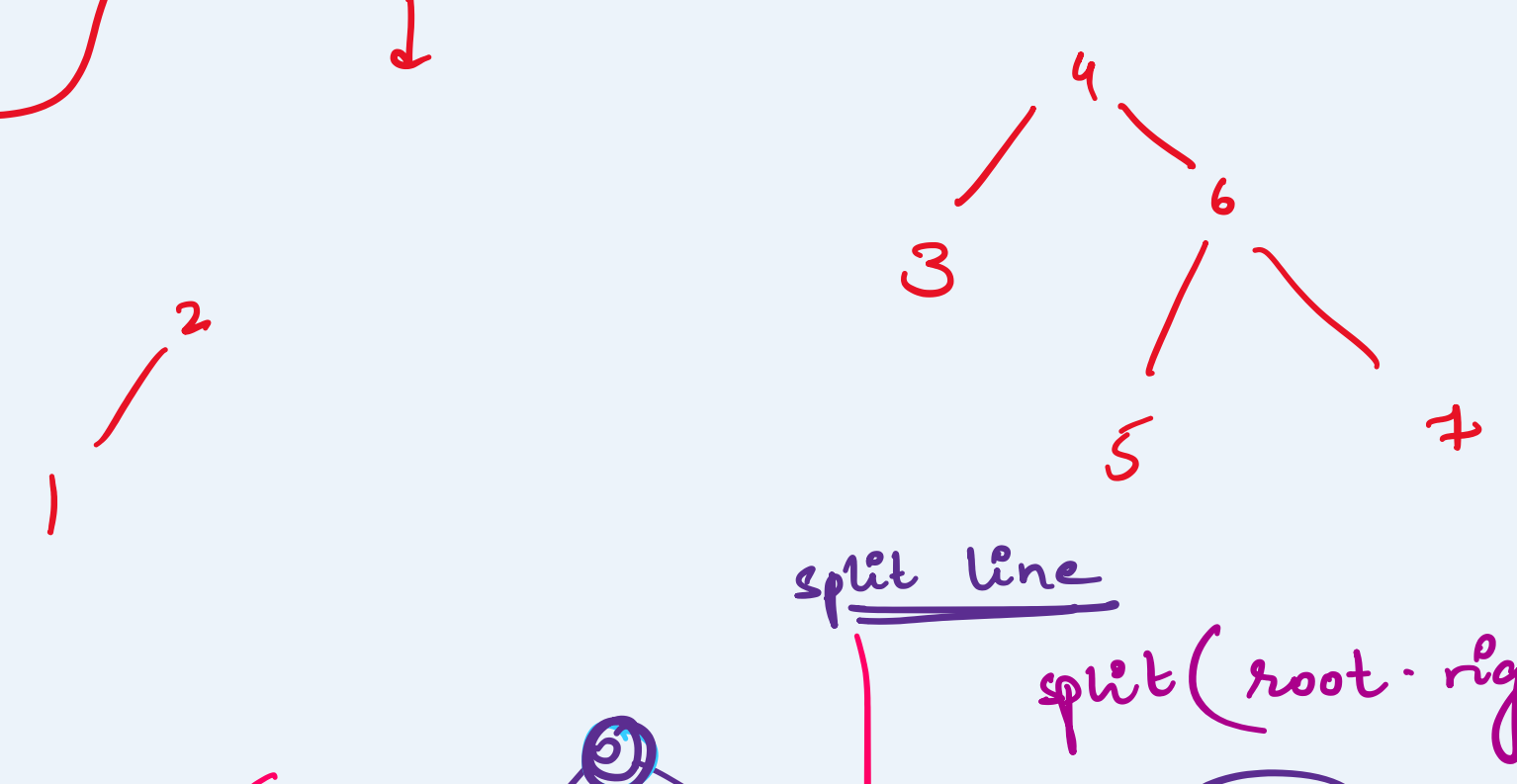
mat → l, b
result = 0

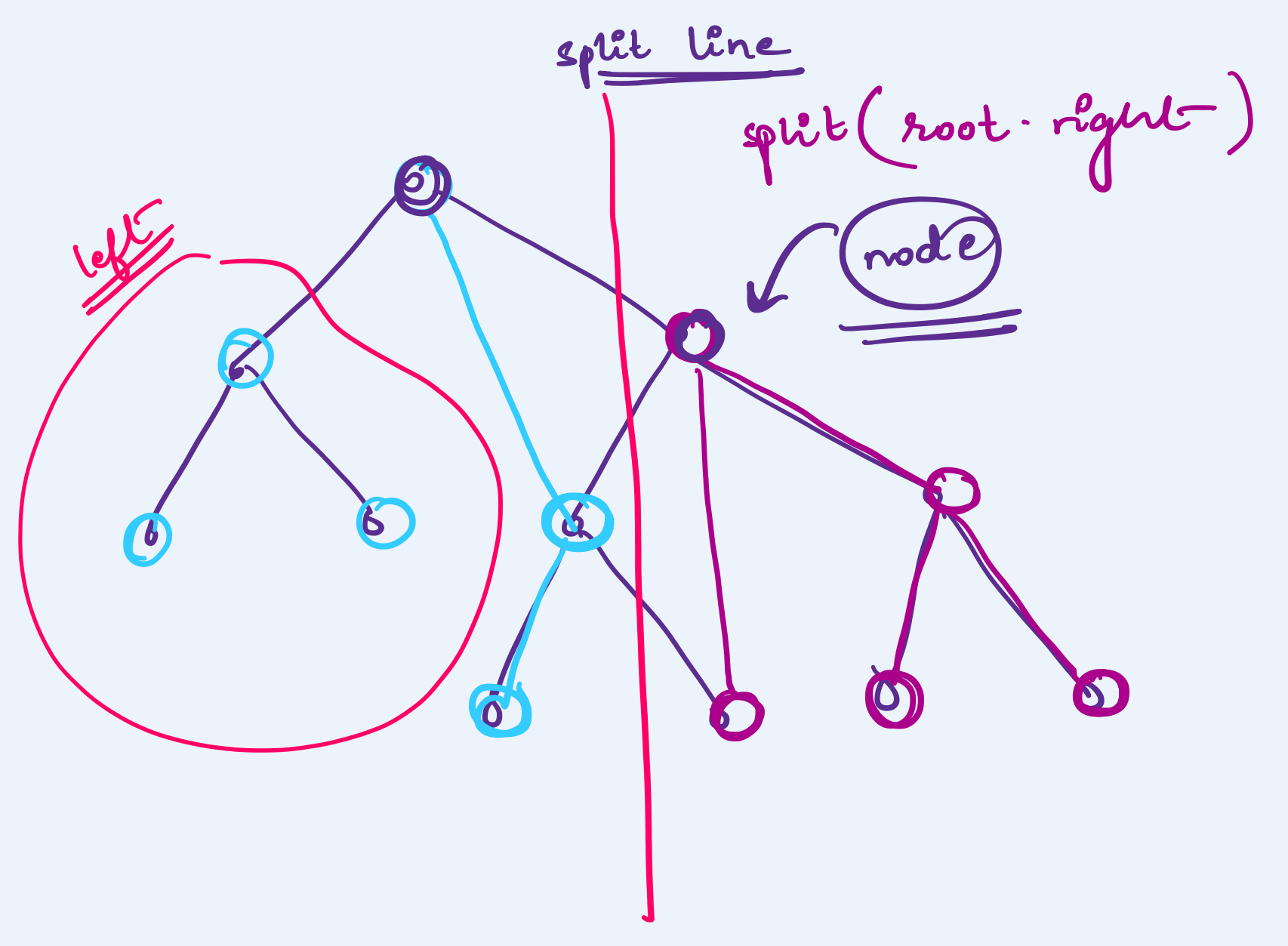
```
for ( i=0; i<n; i++) {
    int Li = input();
    int Bi = input();
    result += max( ( Li/l )x( Bi/b ) , ( Li/b ) x ( Bi/l ))
```

Q. Split BST



target = (2)





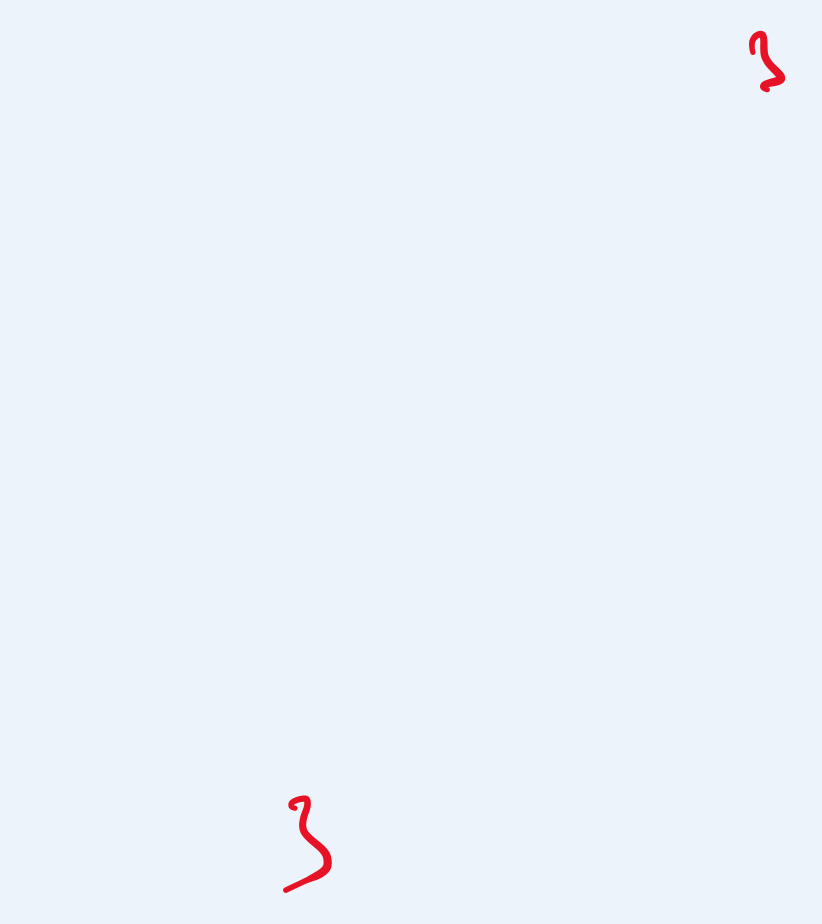

```
node splitBST ( root, v) {
    if (root == null)
        return new node[] {null, null};
    else if (root.val <= v)
        node [] bns = splitBST (root.right, v);
        root.right = bns[0];
        bns [0] = root;
        return bns;
    } else {
        bns = splitBST ( root.left, v);
        root.left = bns[1];
        bns [1] = root;
        return bns;
    }
}
```

Q. K-th smallest prime fraction

arr = [1, 2, 3, 5]
1 → prime no.

(k) = (3)

1/5 , 1/3 , 2/5 , 1/2 , 3/5 , 2/3



ans → (2/5)

(0, 1) , (0, 2), (0, 3) ~~(0, 4)~~

Binary search