Day 3

23 November 2022 19:34

1	2	3

1	2	3

1	3	2

2	1	3

2	3	1

3	1	2

3	2	1

1 2 3

(1, 2, 3) (2, 3)

2 1 3

(1, 2, 3)

| 1 | 2 | 3

1 2 3

(2, 3)

1 3 2

2 1 3

(1, 3)

2 3 1

3 1 2

(1, 2)

3 2 1

[1, 2, 3]

1 — —

2 — —

3 — —

1	2	3
1	3	2
2	1	3
2	3	1
3	2	1
3	1	2

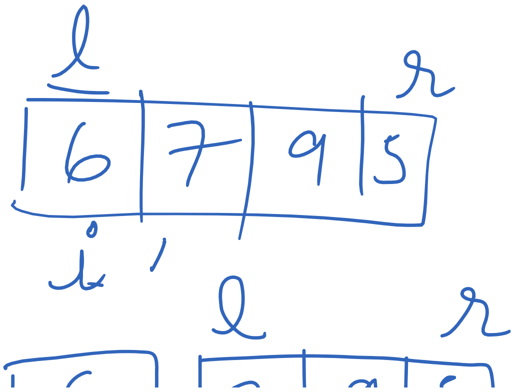
6

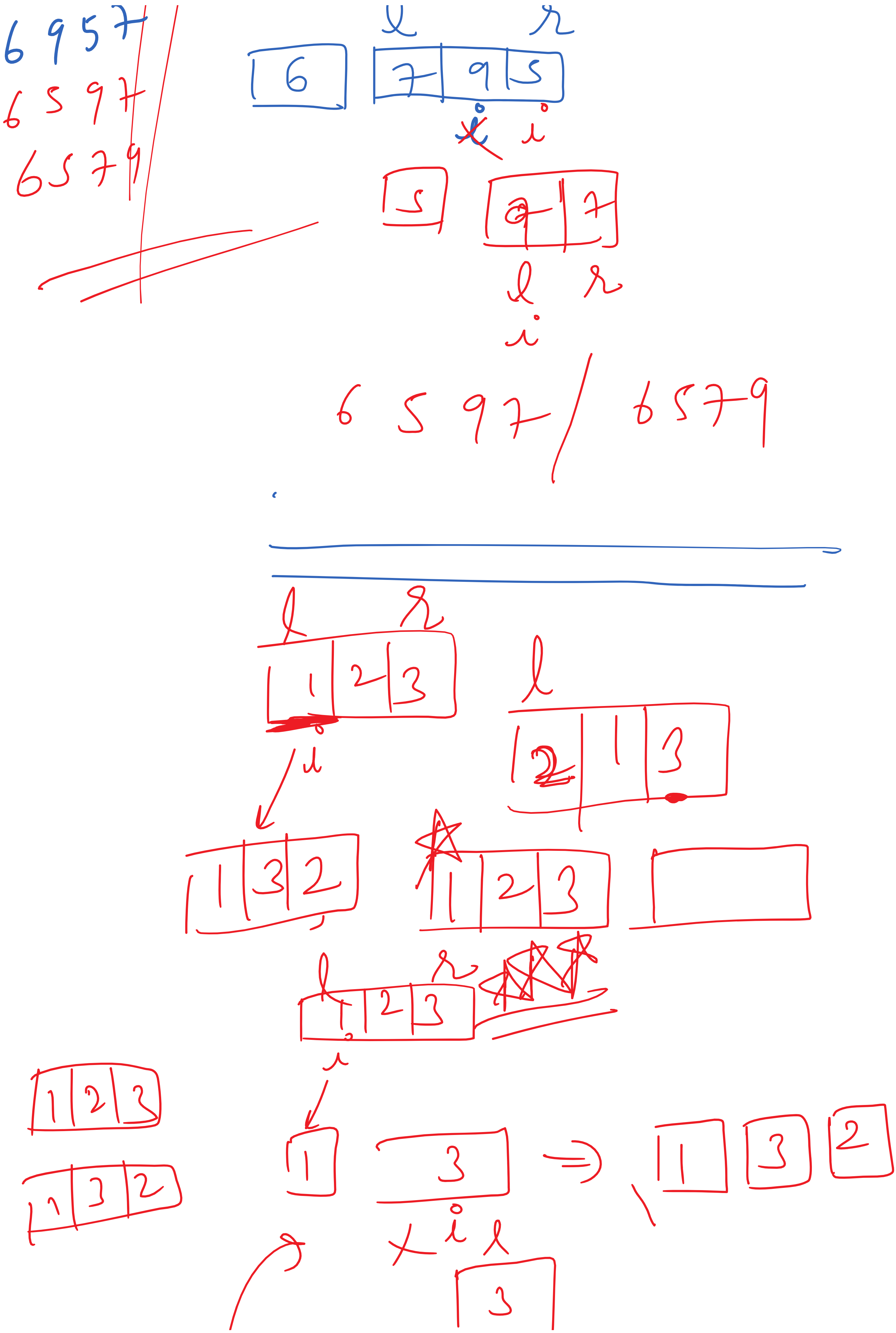
6795

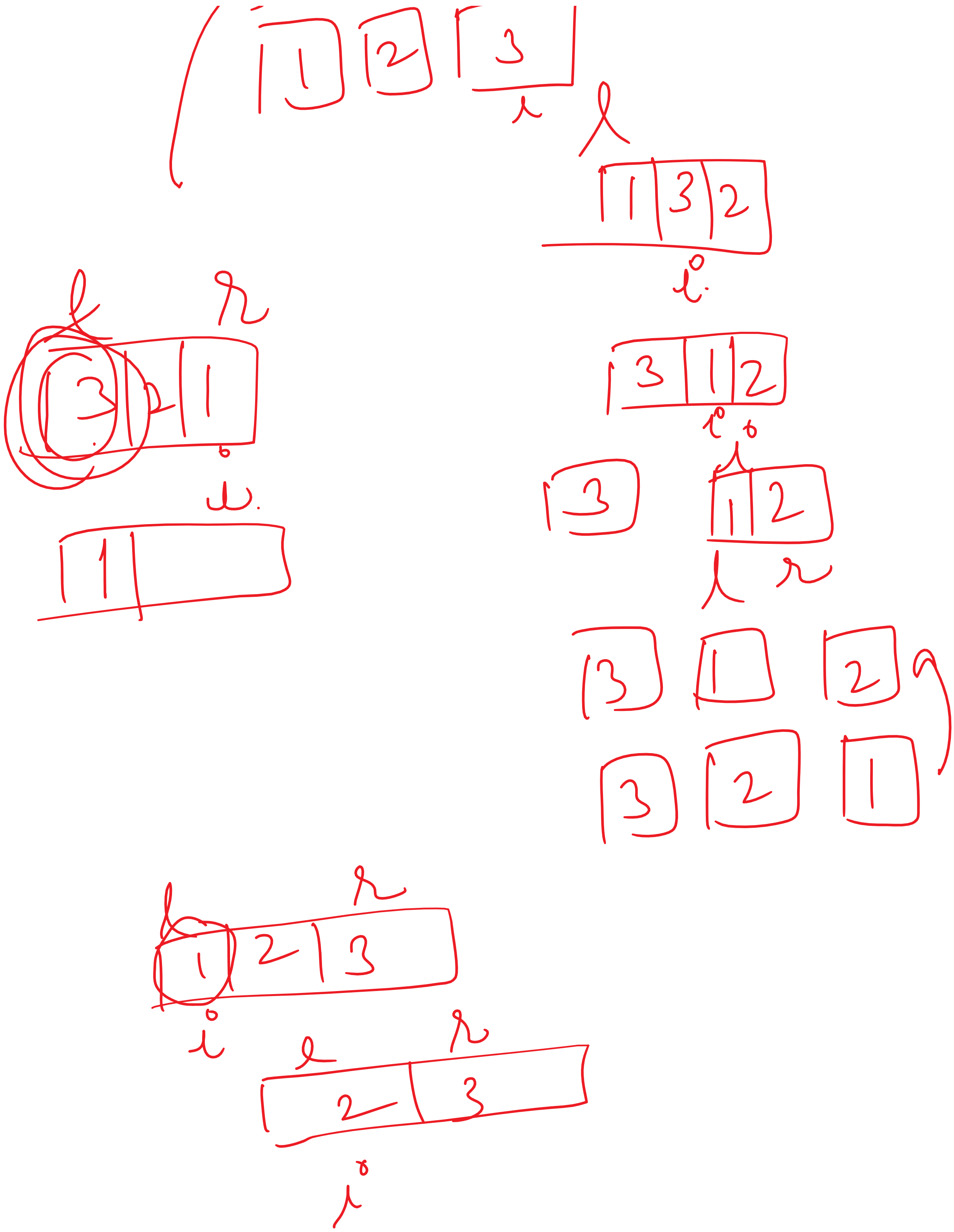
6759

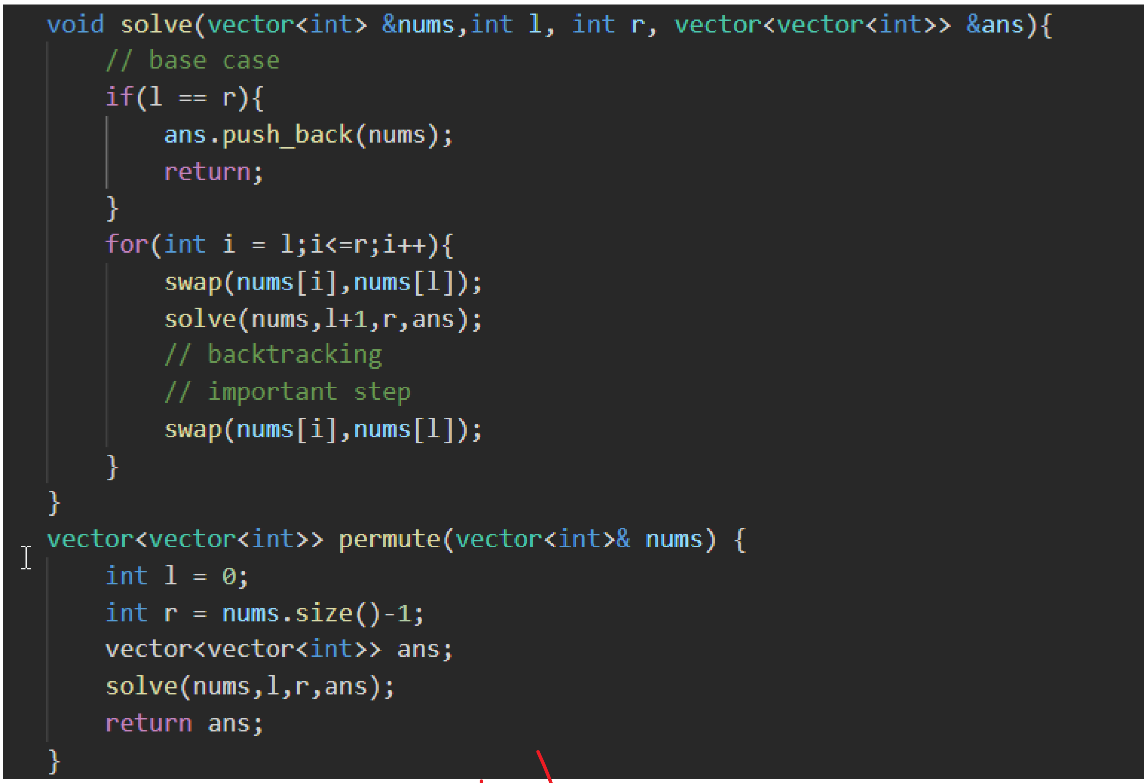
6975

6957



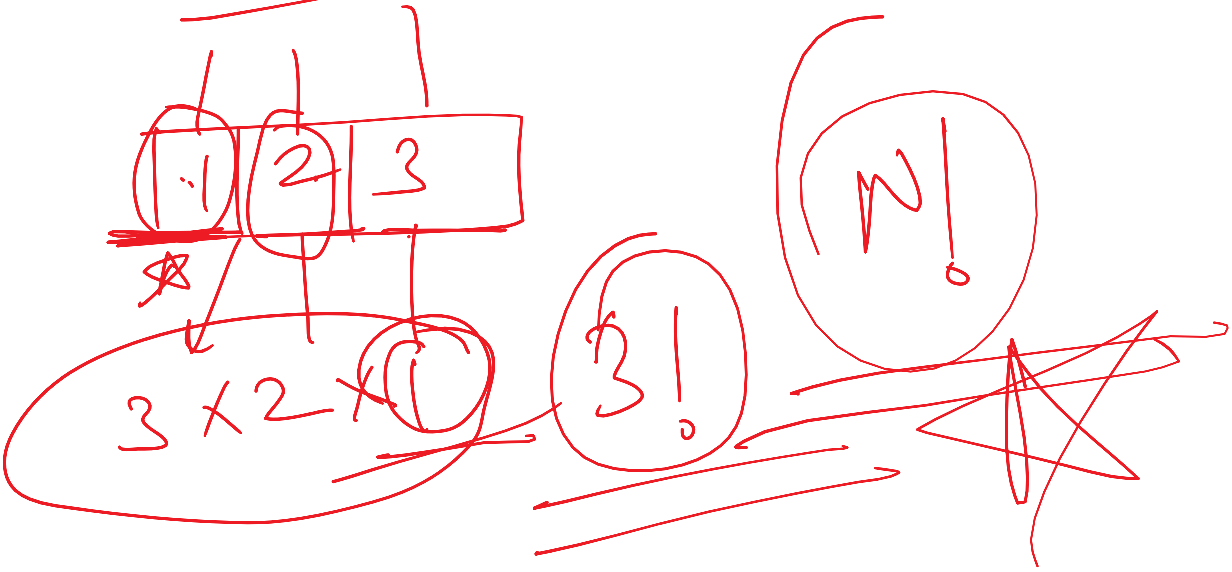

6957
6597
6579
l r
6 7 9 5
l i
5 9 7
l r
i
6597 / 6579
l r
1 2 3
i
l
2 1 3
1 3 2
1 2 3
l r
1 2 3
i
1 2 3
1 3 2
1 3 ⇒ 1 3 2
i l
3


1 2 3
i
l
1 3 2
i
3 1 2
i
l
3
1 2
l r
3 1 2
3 2 1
l r
3 2 1
u
1
l r
1 2 3
i
l r
2 3
i



```
void solve(vector<int> &nums,int l, int r, vector<vector<int>> &ans){
    // base case
    if(l == r){
        ans.push_back(nums);
        return;
    }
    for(int i = l;i<=r;i++){
        swap(nums[i],nums[l]);
        solve(nums,l+1,r,ans);
        // backtracking
        // important step
        swap(nums[i],nums[l]);
    }
}
vector<vector<int>> permute(vector<int>& nums) {
    int l = 0;
    int r = nums.size()-1;
    vector<vector<int>> ans;
    solve(nums,l,r,ans);
    return ans;
}
```

T.C - O(N!)



Combination

n=4 k=2

[1, n]

n=4 k=2

[1, 4]

1, 2, 3, 4

[1, 2] [1, 3] [1, 4] [2, 3] [2, 4] [3, 4]

~~2,1, 4~~

n=4 k=2

1	2	3	4

1	2

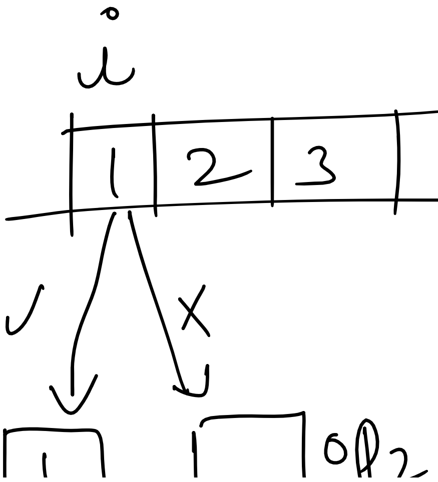
1	3

1	4

2	3

2	4

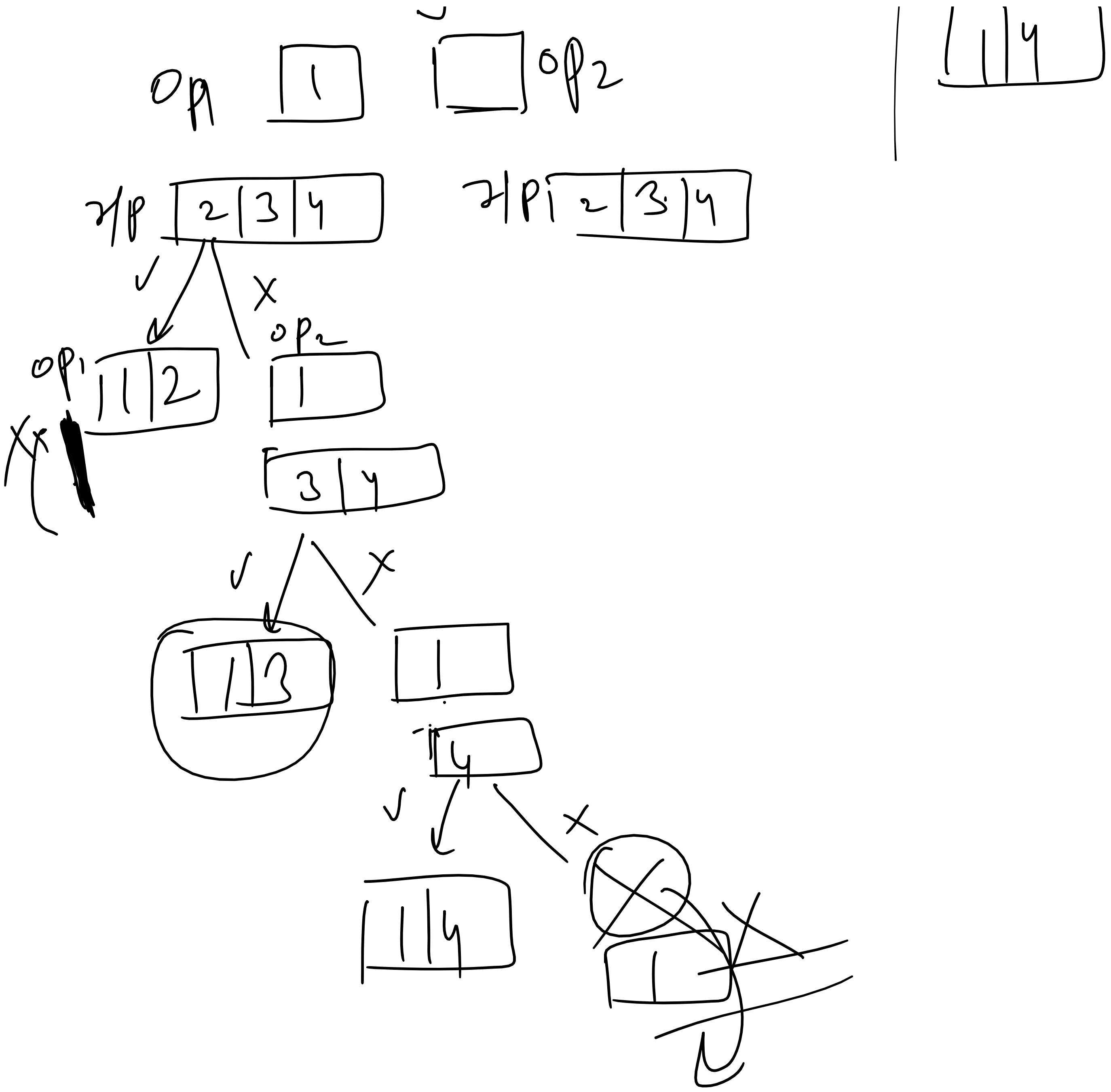
3	4

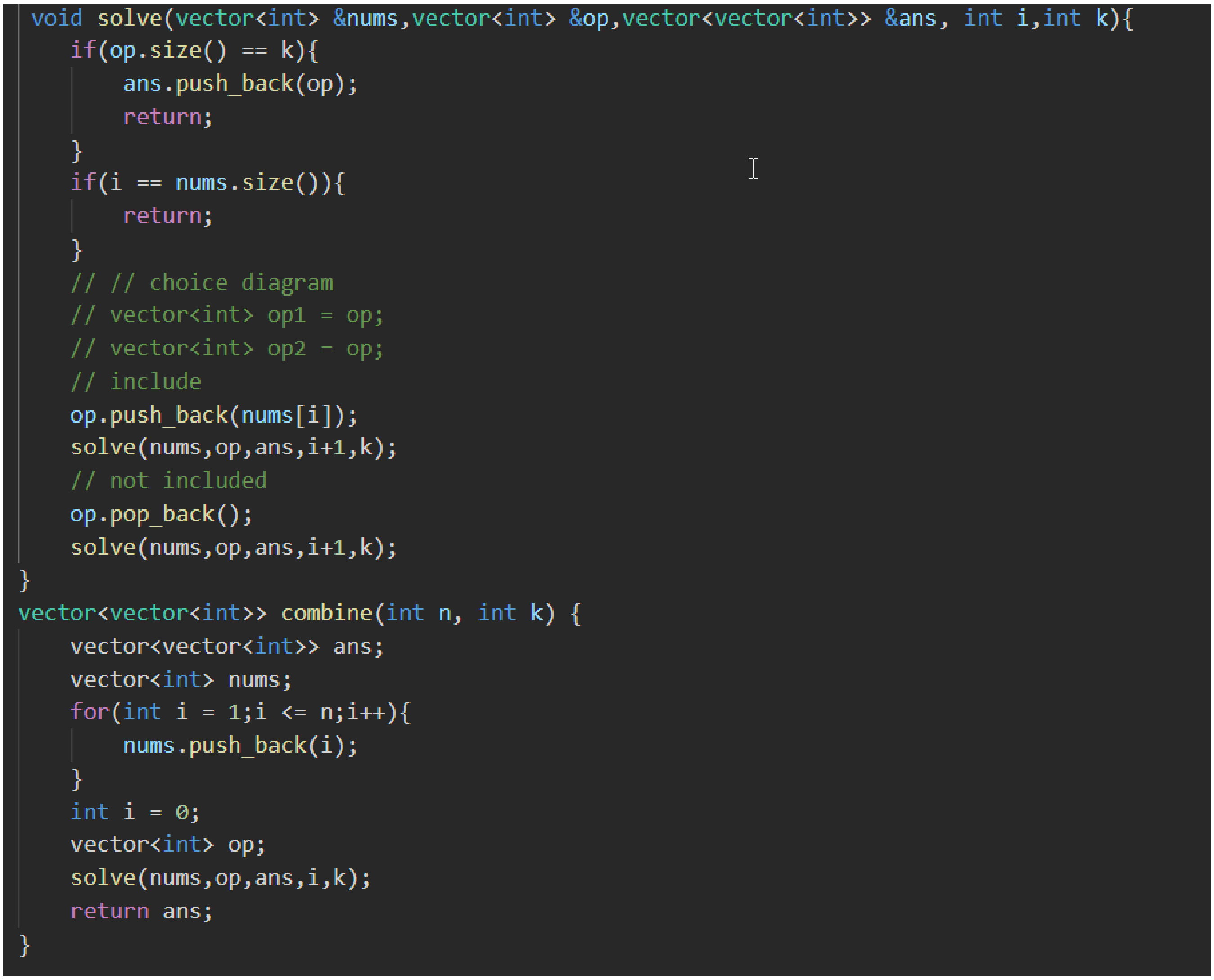


1	2

1	3

1	4


Op 1
1 op2
1 | 4
i/p 2 | 3 | 4
i/p 2 | 3 | 4
✓
X
op1 1 | 2
op2 1
3 | 4
✓
X
1 | 3
1
4
✓
X
1 | 4
1



```
void solve(vector<int> &nums,vector<int> &op,vector<vector<int>> &ans, int i,int k){
    if(op.size() == k){
        ans.push_back(op);
        return;
    }
    if(i == nums.size()){
        return;
    }
    // // choice diagram
    // vector<int> op1 = op;
    // vector<int> op2 = op;
    // include
    op.push_back(nums[i]);
    solve(nums,op,ans,i+1,k);
    // not included
    op.pop_back();
    solve(nums,op,ans,i+1,k);
}
vector<vector<int>> combine(int n, int k) {
    vector<vector<int>> ans;
    vector<int> nums;
    for(int i = 1;i <= n;i++){
        nums.push_back(i);
    }
    int i = 0;
    vector<int> op;
    solve(nums,op,ans,i,k);
    return ans;
}
```

Kth symbol in Grammer

n=1   k=1

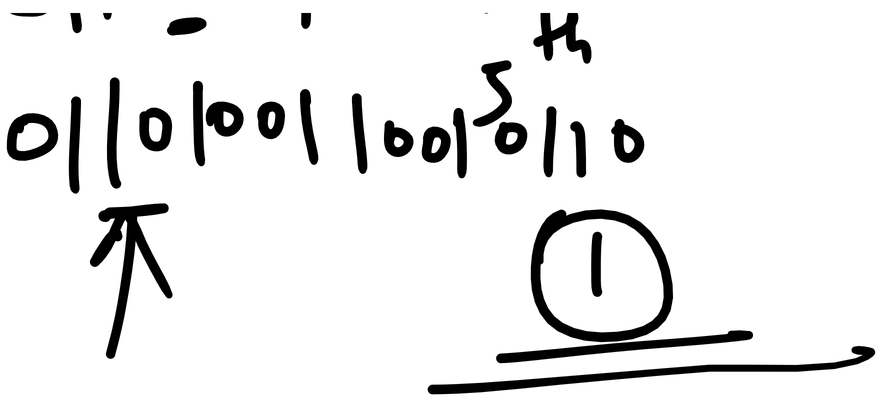
0 → 1st

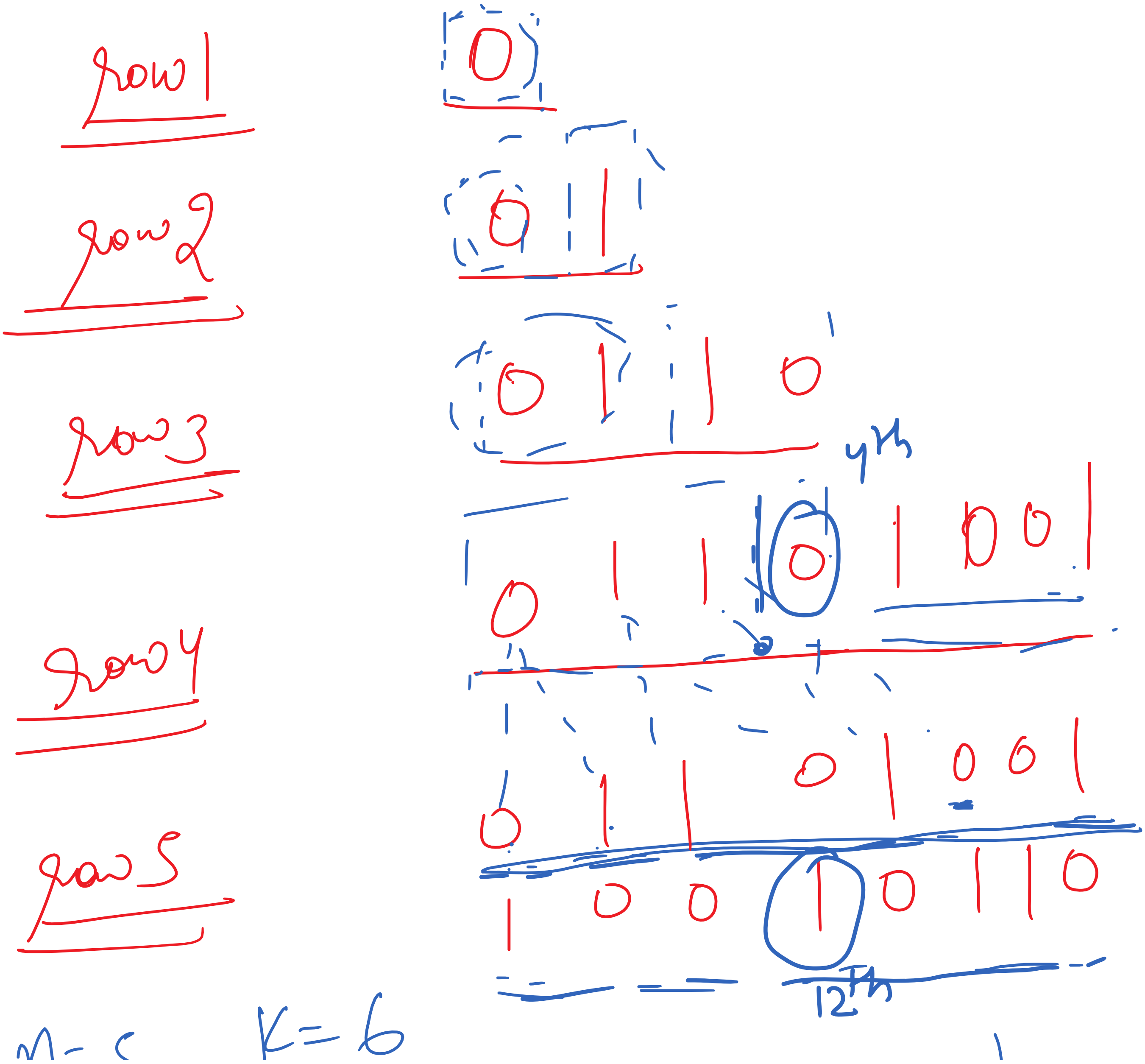
01 → 2nd

0110 → 3rd

01101001 → 4th

5th


5th
0110100110010110
1


row1
0
row2
0 1
row3
0 1 1 0
4th
row4
0 1 1 0 1 0 0 1
row5
0 1 1 0 1 0 0 1
1 0 0 1 0 1 1 0
12th
n- c
K=6

1st half (n-1th row)

equal

2nd half | (n-1 th row)

n=5 k=12

n=5   K=6

Total ⇒ 2^{n-1} ⇒ $\frac{16}{\cancel{2}}$ $\boxed{8}$

$\boxed{\frac{2^{n-1}}{2}}$

left | ! row(n-1)

row (n-1)

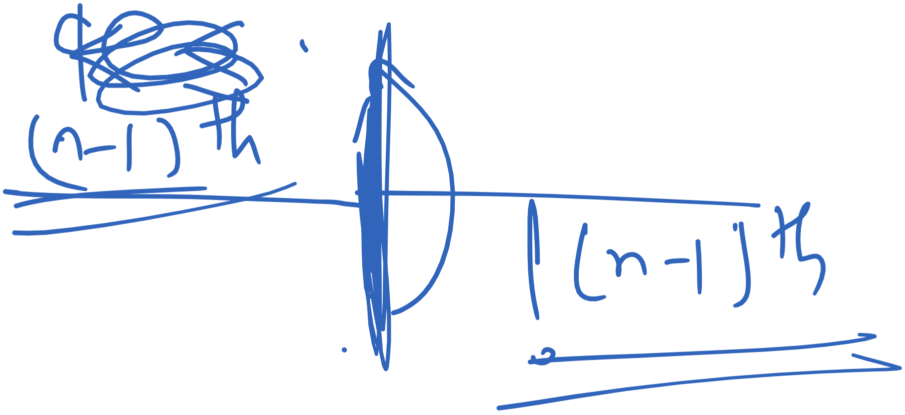
2^{n-1} n=5

$2^{5-1} = 2^4 \Rightarrow 16$

$2^{5-1} = 2^1 \Rightarrow \underline{16}$

☆ $(n-1)^{th}$ row half n^{th} row

second half n^{th} row $!(n-1)^{th}$ row



n=5 k=6

n=12

$$\text{element} = \left(\frac{2^{n-1}}{2}\right) \Rightarrow \underline{2^{n-2}}$$

$\boxed{K \le 2^{n-2}}$ else

$(n-1, K)$ $!(n-1, K-2^{n-2})$

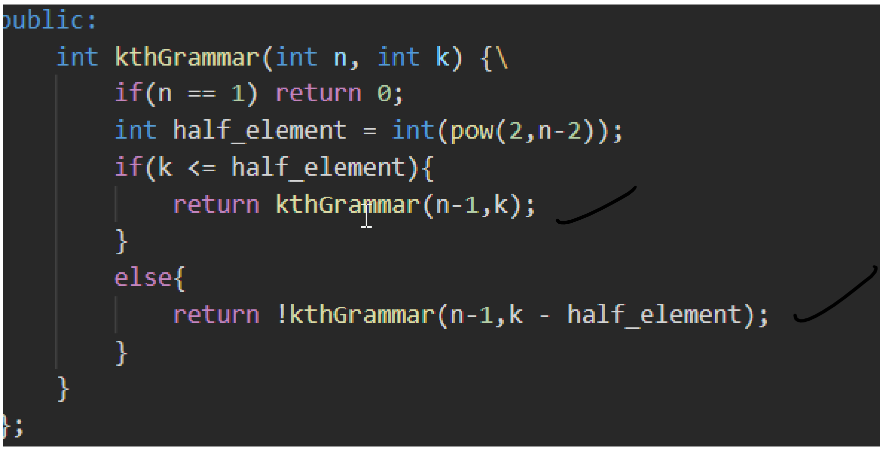
n=5
k=12

```
int kthGrammar(int n, int k){
// [illegible] int(pow(2,n-1));
```

I | int element = int(pow(2,n-1));
  | int half elem = element/2

II { int halfelement= int(pow(2,n-2));

if(K<=half element) {
return kthGramme(n-1, K);
else {
return !kthGramma(n-1,K-half element);
}



```
ublic:
    int kthGrammar(int n, int k) {\
        if(n == 1) return 0;
        int half_element = int(pow(2,n-2));
        if(k <= half_element){
            return kthGrammar(n-1,k);
        }
        else{
            return !kthGrammar(n-1,k - half_element);
        }
    }
};
```

T.C — O(N) ★★★★

n7 S.C — O(N) → [Recursive stack]

n+1 S.C – O(N) →

1

|

/

9th