Print all Subsets

arr = {1,2,3}

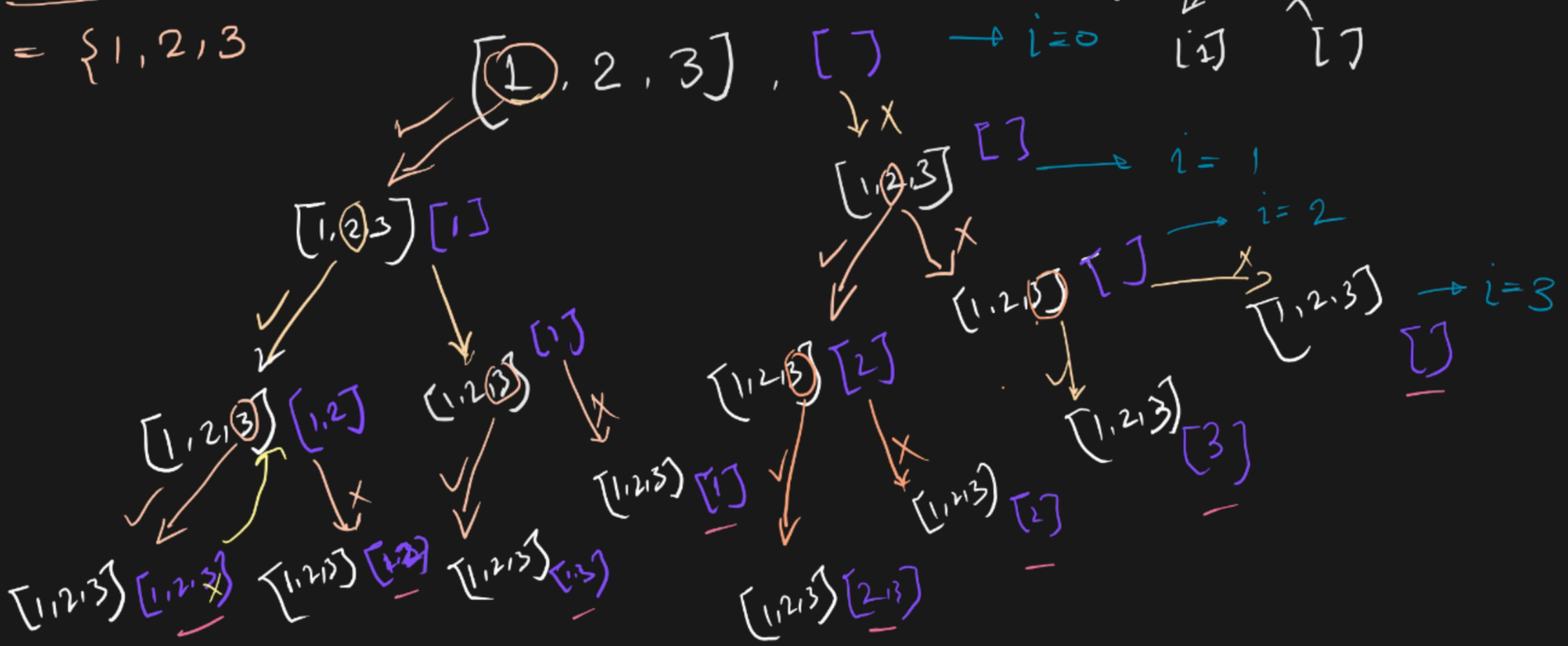
{1} {2} {3} , {1,2} , {1,3} , {2,3} , {1,2,3} , {}

n=3 $8 = 2^3$

n elements = 2^n subsets

Using Recursion :

arr = {1,2,3



we have all subsets : [1,2,3] , [1,2], [1,3] , [1] , [2,3] , [2] , [3] , [

→ arr [ ]

→ ans = subset [ ]

→ i → index = 0

arr = [1,2,3]

```
Void Printsubsets (arr[], ans[], i){
    if (i == arr.size()) return;
    ans.push_back(arr[i]),                    } include
    Printsubsets ( arr, ans, i+1);
    // Backtrack
    ans.pop_back();                           } exc
    Printsubsets (arr, ans, i+1);
```

// Full code

```
# include <iostream>
```

```
#  include <vector>
   using namespace std;
Void Printsubsets (vector<int> &arr, vector<int> &ans, int i){
      if (i == arr.size())
      {     for(int val : ans ){
                cout << val<< " ";
            }
            cout<< endl ;
            return;
      }
      // include
      ans.push_back (arr[i]);
      Printsubsets (arr, ans, i+1);

      ans.pop_back ();          // Backtrack
      Printsubsets (arr, ans, i+1);
}
 int main () {
      vector<int> arr = {1, 2, 3};
      vector<int> ans;
      Printsubsets (arr, ans, 0);
      return 0;
}
```

Complexities:- total calls * work Done by each calls

$2^n * n$