DSA Tutorial - 1

Topics: 1. Recursion
2. Merge Sort and Quick Sort

Anmol Agarwal and Dhvanil Sanghi

Recursion

Recursion: Objectives

- What is Recursion
- How Recursion Works
  - Call Stack
  - Recursion Tree
- Framework for solving Recursion Problems
  - Subproblems
  - Decisions

What is Recursion

- In simple terms: A function calling itself is recursion.

```
f (….)
{
        <code>
        f(….)
        <code>
}
```

```
f(int x)
{   if(x>=3) return;
    cout << x << endl;
    x++;
    f(x);
    cout << x << endl;
}
main(){
    f(1);
    return 0;
}
```

Framework for Solving Recursion Problems

- Formulate a decision
- Define Subproblems
- Come up with a recurrence relation
- Define base cases (Very Important!)
- Code

Framework for Solving Recursion Problems

- Given a set of distinct integers, return all the subsets of the set.

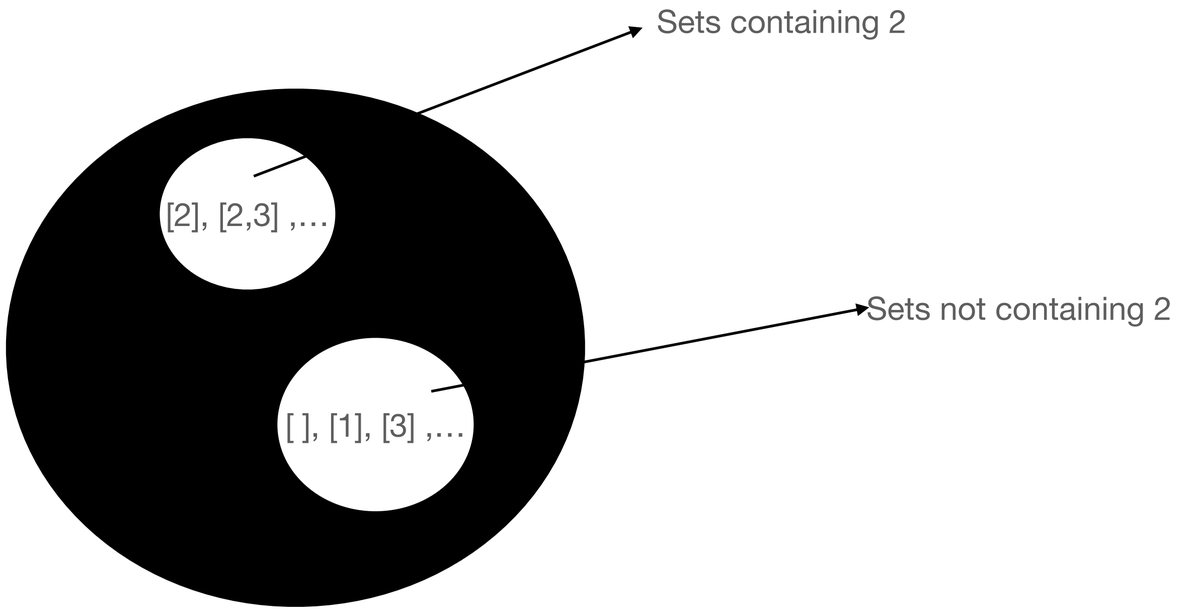
  Eg: [1,2,3]

  Answer : [ ] , [1] , [2] , [3] , [1,2] , [1,3] , [2,3] , [1,2,3]

Decision

Decision on the basis of 1st element of array: Include/Exclude

- Take the set [2,3,4,5,6] for example



Define the Subproblem

2	3	4	5	6

$$S_1 = [3,4,5,6]$$ ← Suffix Subarray

Subproblem: Smaller instance of the same problem, there suffix subarray

Recurrence Relation: Relationship between the problem and subproblem

$$S = S \cup \{2 \cup S_1\}$$

Base Case

Solution to smallest subproblem

- Smallest subproblem : Empty set [ ]

Code

https://leetcode.com/problems/subsets/