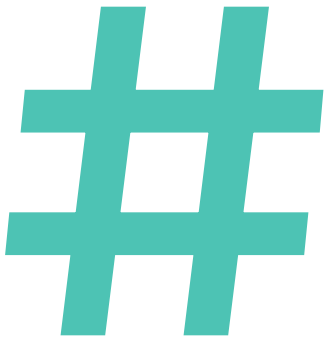


Backtracking Algorthim

Step 01 What is Backtracking Algorithm?

Backtracking is a general algorithmic technique to solve problems by recursively exploring all possible solutions.

01 How does Backtracking work?

- Choose: Choose the initial step towards a solution.
- Explore: Explore all possible options from the current choice.
- Evaluate: Determine whether the current choice leads to a solution.
- Backtrack: If the choice doesn't lead to a solution, undo (backtrack) the choice to the previous choice and try another option.

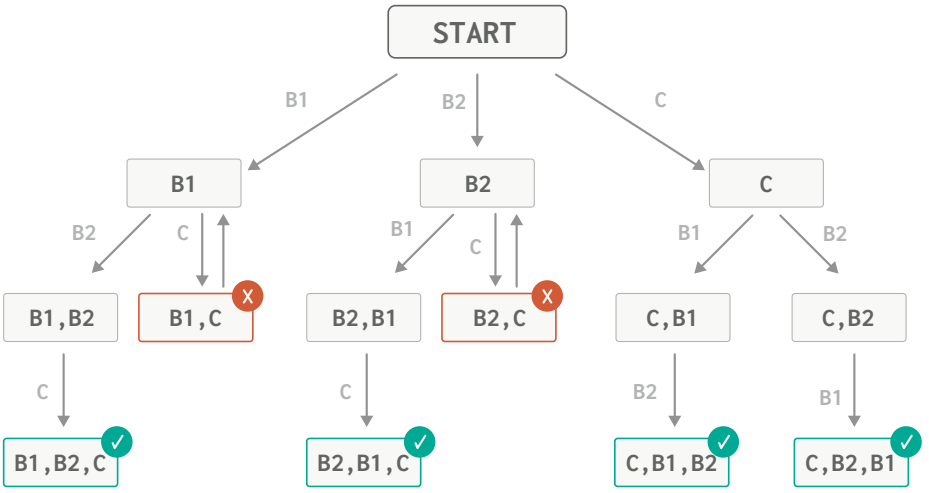
02 Types of problems solved by backtracking algorithms:

- Decision Problem: search for a feasible solution.
- Optimization Problem: search for the best possible solution.
- Enumeration Problem: try to find all feasible solutions.

Step 02 How to Implement Backtracking Algorithm

Suppose you have two bikes B1 & B2. And one car C. Find all possible ways to arrange them.

> **NOTE**
> Constraint: Car should not be between bikes.



```java
import java.util.ArrayList;
import java.util.Arrays;

public class Backtracking {
    public static void main(String[] args) {
        String[] vehicles = {"C", "B2", "B1"};
        arrangeVehicles(vehicles);
    }

    public static void arrangeVehicles(String[] vehicles) {
        ArrayList<String[]> solutions = new ArrayList<>();

        for (int i = 0; i < vehicles.length; i++) {
            for (int j = 0; j < vehicles.length; j++) {
                if (i == j) {
                    continue;
                }

                swap(vehicles, i, j);

                if (vehicles[1]== “C”) {
                    swap(vehicles, i, j); // Backtrack from the swap
                    continue;
                } else if (!isRowInList(vehicles, solutions)) {
                    solutions.add(Arrays.copyOf(vehicles, vehicles.length));
                }
            }
        }

        System.out.println("All the possible ways of arranging 2 bikes and 1 car:");
        for (String[] arrangement : solutions) {
            for (String vehicle : arrangement) {
                System.out.print(vehicle + “ “);
            }
            System.out.println();
        }
    }

    public static void swap(String[] array, int i, int j) {
        String temp = array[i];
        array[i] = array[j];
        array[j] = temp;
    }

    public static boolean isRowInList(String[] arr, ArrayList<String[]> list) {
        for (String[] row : list) {
            if (Arrays.equals(arr, row)) {
                return true;
            }
```

```
50          }
51          return false;
52      }
53  }
```

OUTPUT

All the possible ways of arranging 2 bikes and 1 car:

B1 B2 C

B2 B1 C

C B1 B2

C B2 B1