# 01_NumberOfIsland_DFS

```
public class NumberOfIsland_dfs {

    public int numsIslands(char[][] grid) {
    }
}
```

# 02_NumberOfIsland_BFS

```
public class NumberOfIsland_dfs {

    public int numsIslands(char[][] grid) {
    }
}
```

# 03~05_Maximum Depthof Binary Tree

```
class TreeNode{
        int val;
        TreeNode left, right;
        TreeNode(int x){
            this.val = x;
        }
}


public class MaxDepth {

    public int solve(TreeNode root) {
    }
}
```

# 06_MaxOfIsland

```
public class MaxOfIsland {

    public int maxAreaOfIsland(int[][] grid) {
    }
}
```

# 07_WordLadder

```
public class WordLadder {

   public int ladder(String beginWord, String endWord, List<String>wordList){
   }
}
```

# 08_WordSearch

```
public class WordSearch {

   public boolean solve(char[][] grid, String word) {
   }
}
```

# 09_RemoveInvalidParentheses

```
public class RemoveInvalidParentheses{

   public List<String> solve(String s){
   }
}
```

# 10_Maze1_bfs

```
public class Maze1_bfs {

   public boolean hasPath(int[][] maze, int[] start, int[] destination) {
   }
}
```

# 11_Maze1_dfs

```
public class Maze1_dfs {

   public boolean hasPath(int[][] maze, int[] start, int[] destination) {
   }
}
```