极客大学算法训练营
第七课
泛型递归、树的递归

覃超

Sophon Tech 创始人，前 Facebook 工程师

前序知识回顾：

树的面试题解法一般都是递归

1. 节点的定义

2. 重复性（自相似性）

示例代码

```
def preorder(self, root):

  if root:
    self.traverse_path.append(root.val)
    self.preorder(root.left)
    self.preorder(root.right)

def inorder(self, root):

  if root:
    self.inorder(root.left)
    self.traverse_path.append(root.val)
    self.inorder(root.right)

def postorder(self, root):

  if root:
    self.postorder(root.left)
    self.postorder(root.right)
    self.traverse_path.append(root.val)
```

递归
Recursion

递归 Recursion

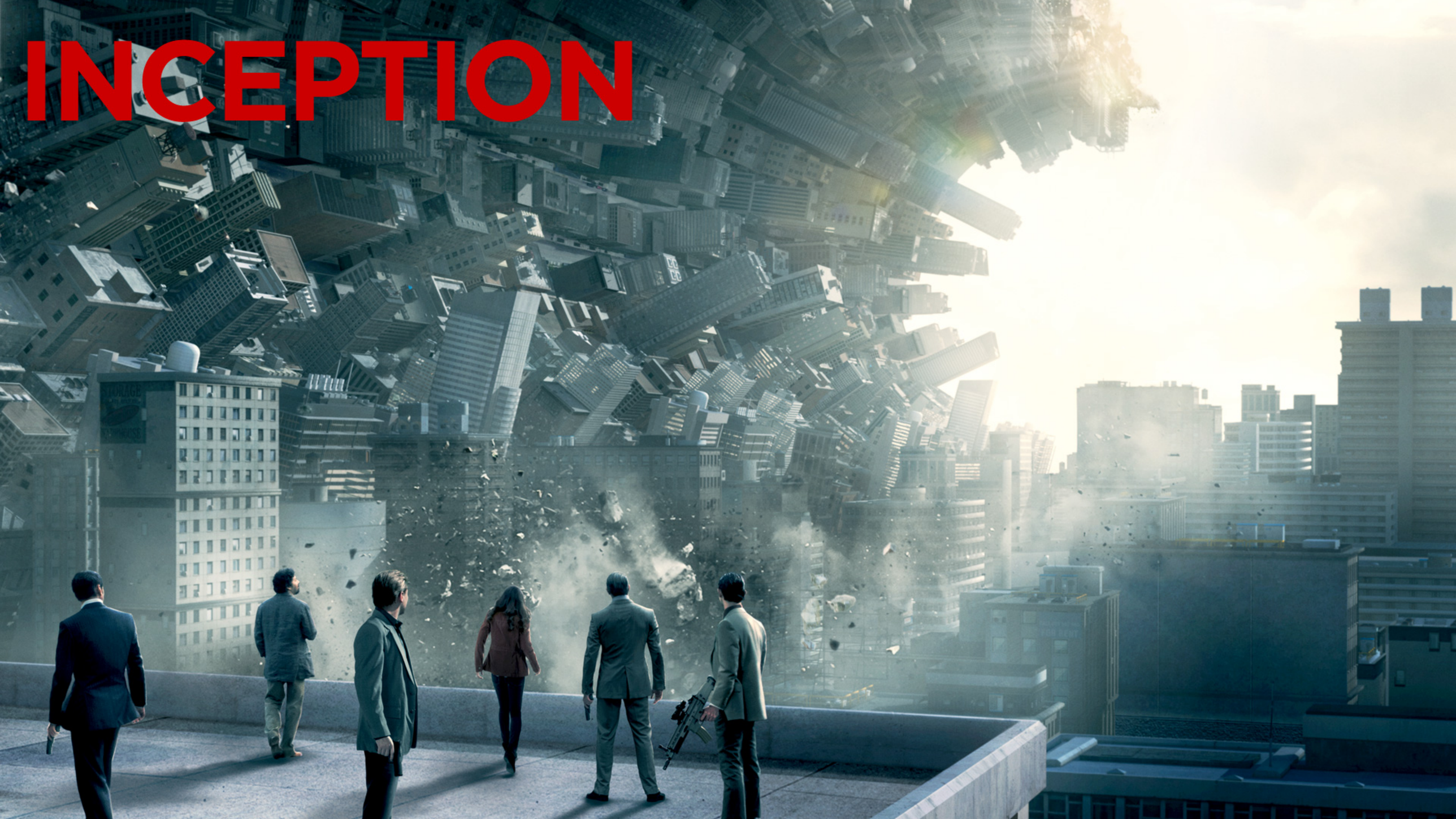
递归 - 循环

通过函数体来进行的循环

递归 Recursion

1. 从前有个山
2. 山里有个庙
3. 庙里有个和尚讲故事
4. 返回1

INCEPTION



盗梦空间

- 向下进入到不同梦境中；向上又回到原来一层
- 通过声音同步回到上一层
- 每一层的环境和周围的人都是一份拷贝、
  主角等几个人穿越不同层级的梦境（发生和携带变化）

递归 Recursion

计算 n!

n! = 1 * 2 * 3 * … * n

```
def Factorial(n):
    if n <= 1:
        return 1
    return n * Factorial(n - 1)
```

递归 Recursion

factorial(6)
6 * factorial(5)
6 * (5 * factorial(4))
6 * (5 * (4 * factorial(3)))
6 * (5 * (4 * (3 * factorial(2))))
6 * (5 * (4 * (3 * (2 * factorial(1)))))
6 * (5 * (4 * (3 * (2 * 1))))
6 * (5 * (4 * (3 * 2)))
6 * (5 * (4 * 6))
6 * (5 * 24)
6 * 120
720

Python 代码模版

```
def recursion(level, param1, param2, ...):
    # recursion terminator
    if level > MAX_LEVEL:
        process_result
        return

    # process logic in current level
    process(level, data...)

    # drill down
    self.recursion(level + 1, p1, ...)

    # reverse the current level status if needed
```

Java 代码模版

```java
public void recur(int level, int param) {

    // terminator
    if (level > MAX_LEVEL) {
      // process result
      return;
    }

    // process current logic
    process(level, param);

    // drill down
    recur( level: level + 1, newParam);

    // restore current status

}
```

思维要点

1. 不要人肉进行递归（最大误区）
2. 找到最近最简方法，将其拆解成可重复解决的问题（重复子问题）
3. 数学归纳法思维

实战题目

1. https://leetcode-cn.com/problems/climbing-stairs/

2. https://leetcode-cn.com/problems/generate-parentheses/

实战题目

1. https://leetcode-cn.com/problems/invert-binary-tree/description/
2. https://leetcode-cn.com/problems/validate-binary-search-tree
3. https://leetcode-cn.com/problems/maximum-depth-of-binary-tree
4. https://leetcode-cn.com/problems/minimum-depth-of-binary-tree
5. https://leetcode-cn.com/problems/serialize-and-deserialize-binary-tree/

Homework

1. https://leetcode-cn.com/problems/lowest-common-ancestor-of-a-binary-tree/

2. https://leetcode-cn.com/problems/construct-binary-tree-from-preorder-and-inorder-traversal

3. https://leetcode-cn.com/problems/combinations/

4. https://leetcode-cn.com/problems/permutations/
   https://leetcode-cn.com/problems/permutations-ii/