第 7 课
递归子集问题

薛浩

xuehao0618@outlook.com

阅读

- Programming Abstraction in C++ *Chapter 8.2*

今日话题

- 回顾
- 递归子集问题

回顧

递归的概念

将规模较大的问题，通过合理的简化手段，

转变为相同形式的规模较小的问题，

从而得到问题求解的一种技术。

递归的概念

将规模较大的问题，通过合理的简化手段，

转变为相同形式的规模较小的问题，

从而得到问题求解的一种技术。

解决递归问题步骤

解决递归问题步骤

- 最简单的情况要如何处理
  - Base Case

解决递归问题步骤

- 最简单的情况要如何处理
  - Base Case
- 把问题拆分成更小形式的子问题
  - Recursive Case

编程形式伪代码

```
if( 最简单的情况 ) {
    直接处理问题;
    返回处理结果。
} else {
    把问题分成一个或多个相同形式的子问题;
    一个个解决这些子问题; （采用相同的逻辑）
    整合这些子问题的结果;
    返回最终结果。
}
```

今日话题

- ~~回顾~~
- 递归子集问题

递归子集问题

问题

求给定集合{1， 3， 5}的所有子集

迭代法求子集

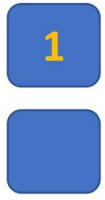
```
// 迭代法求所有子集
void listSubsetsOf(Vector<int> elems) {
    Vector<Vector<int>> subsets = {{}};
    for (int i = 0; i < elems.size(); i++) {
        int size = subsets.size();
        for (int j = 0; j < size; j++)
            subsets.add(subsets[j] + elems[i]);
    }
    for (const auto &subset : subsets)
        cout << subset << endl;
}
```


1
3
5

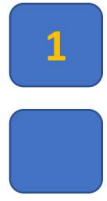

3
5


1


3
5

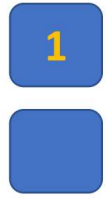

1


3
5


1

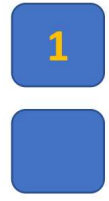

5


3


1

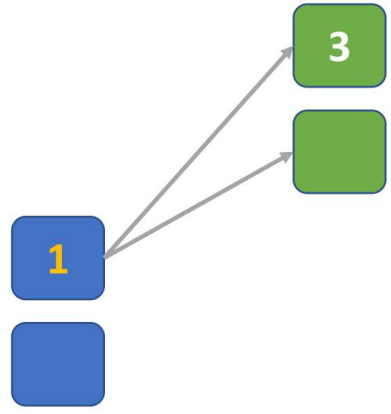

5


3

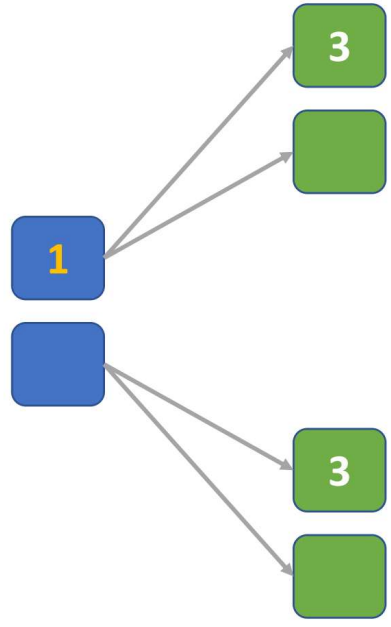

1

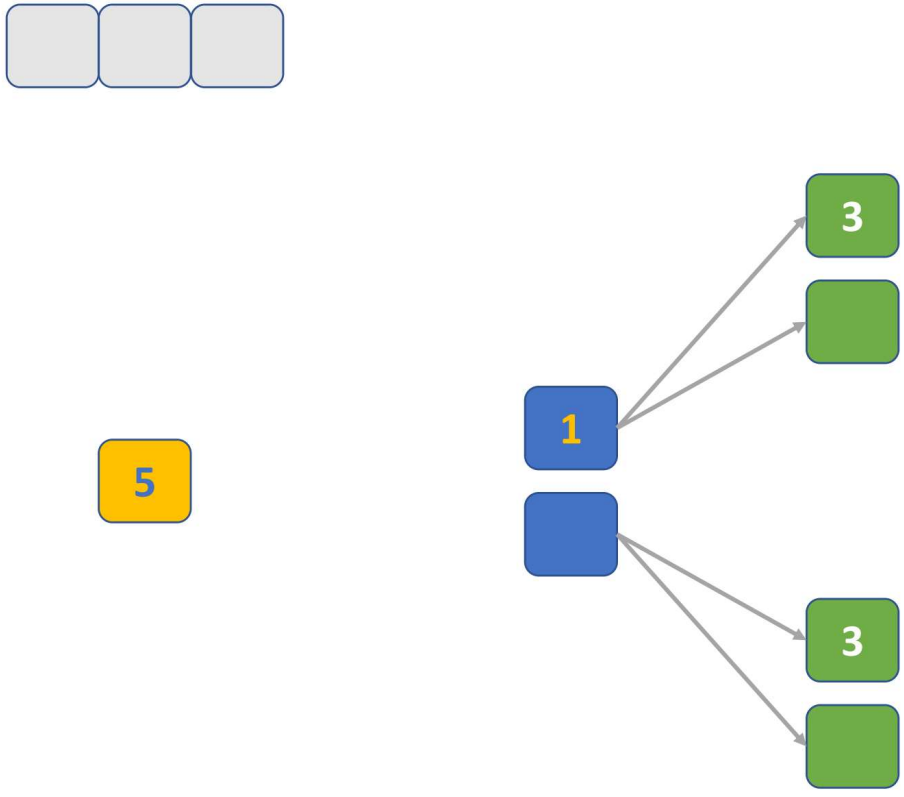

5


3

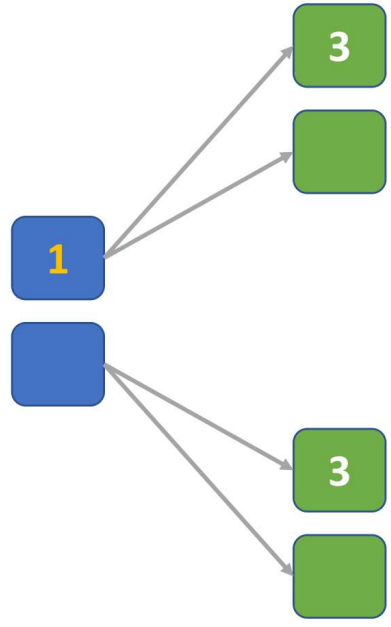

3
1

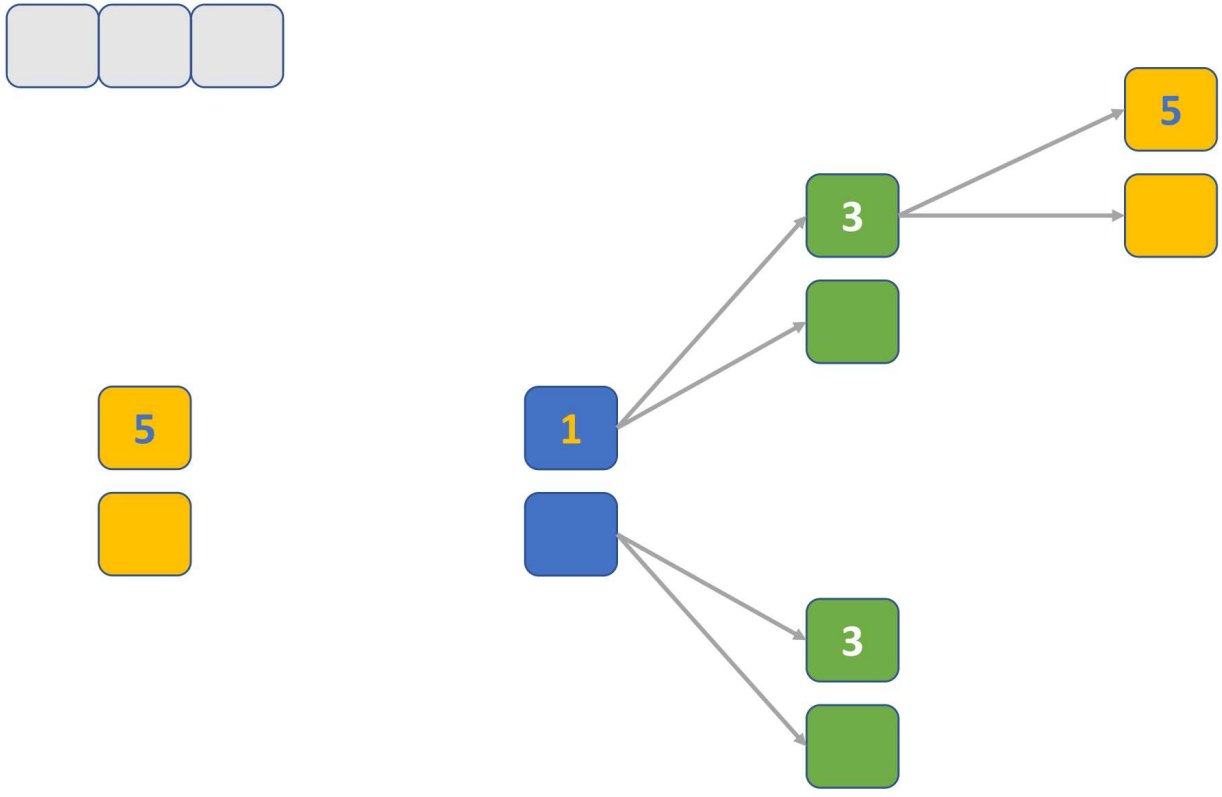

5

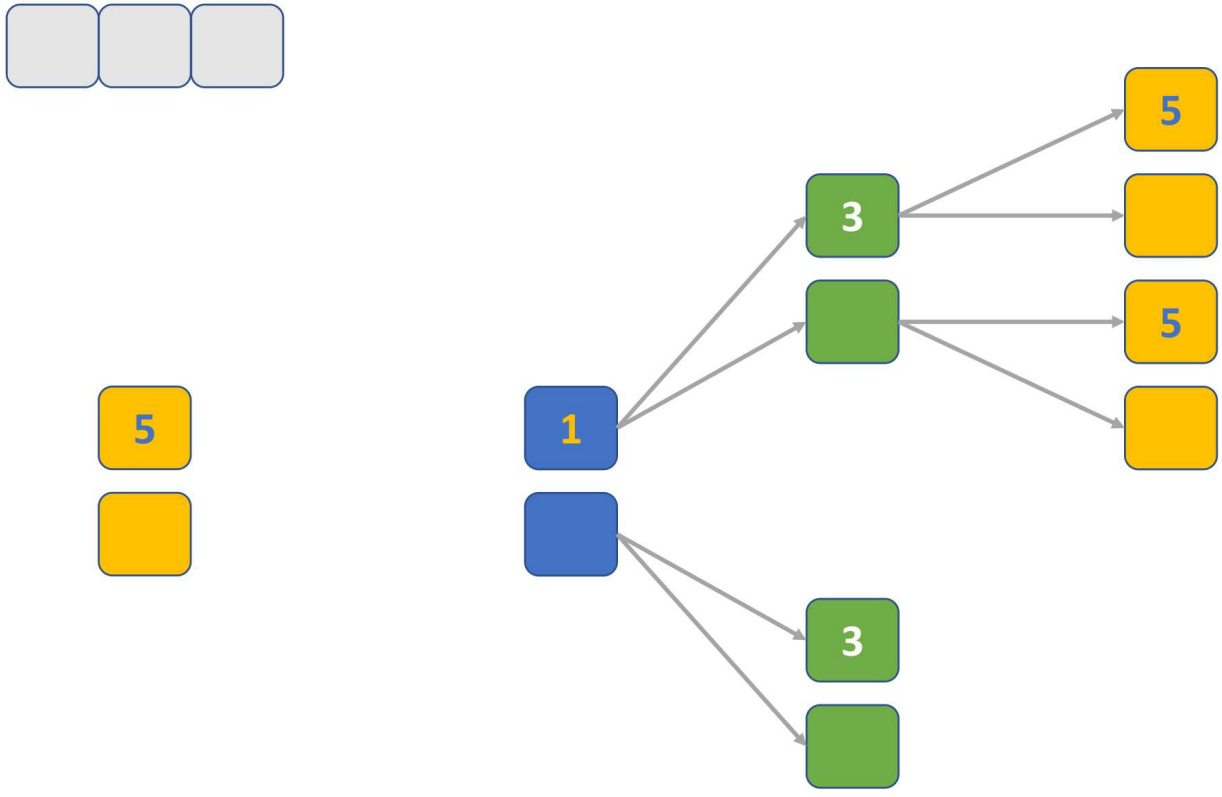

3
1
3


5
1
3
3

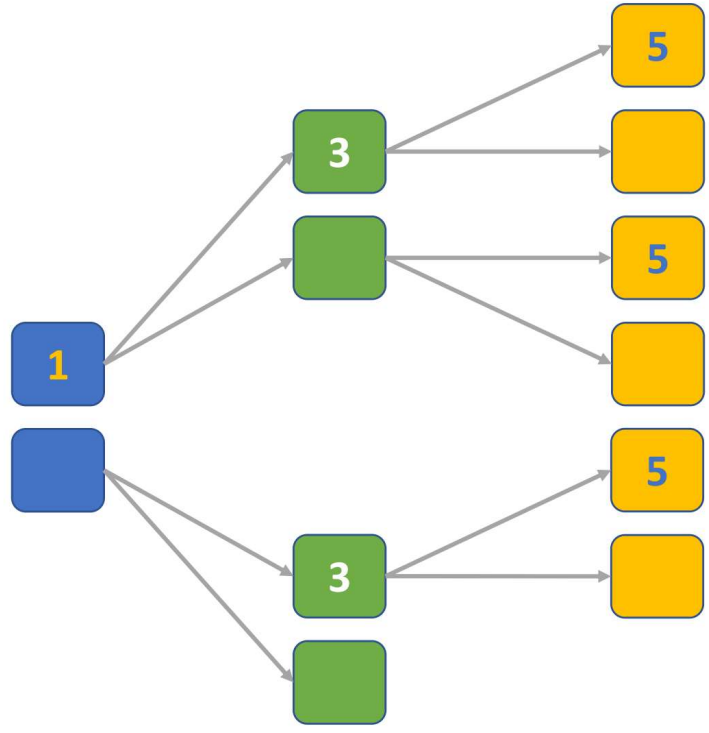

5

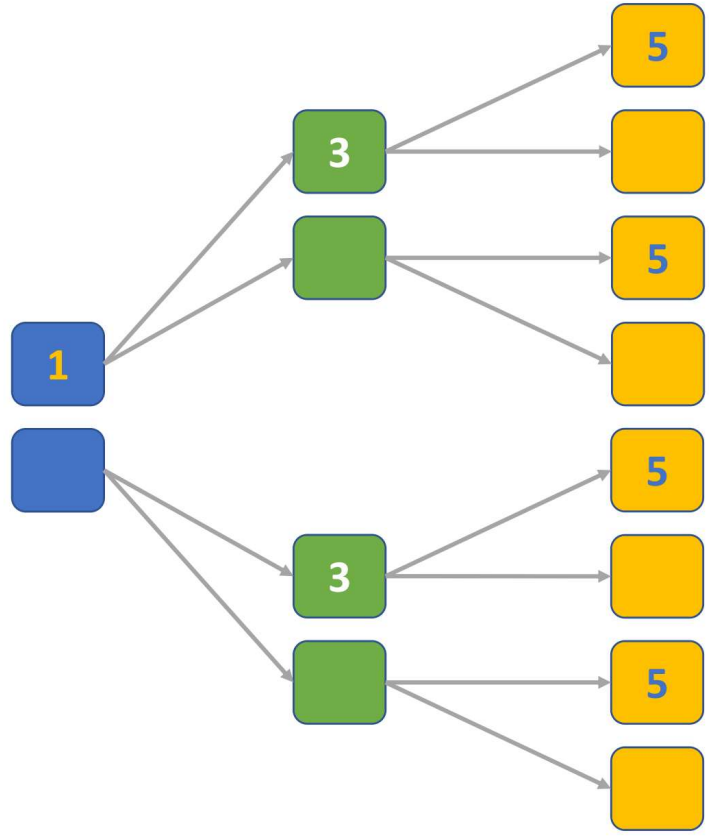

3
1
3

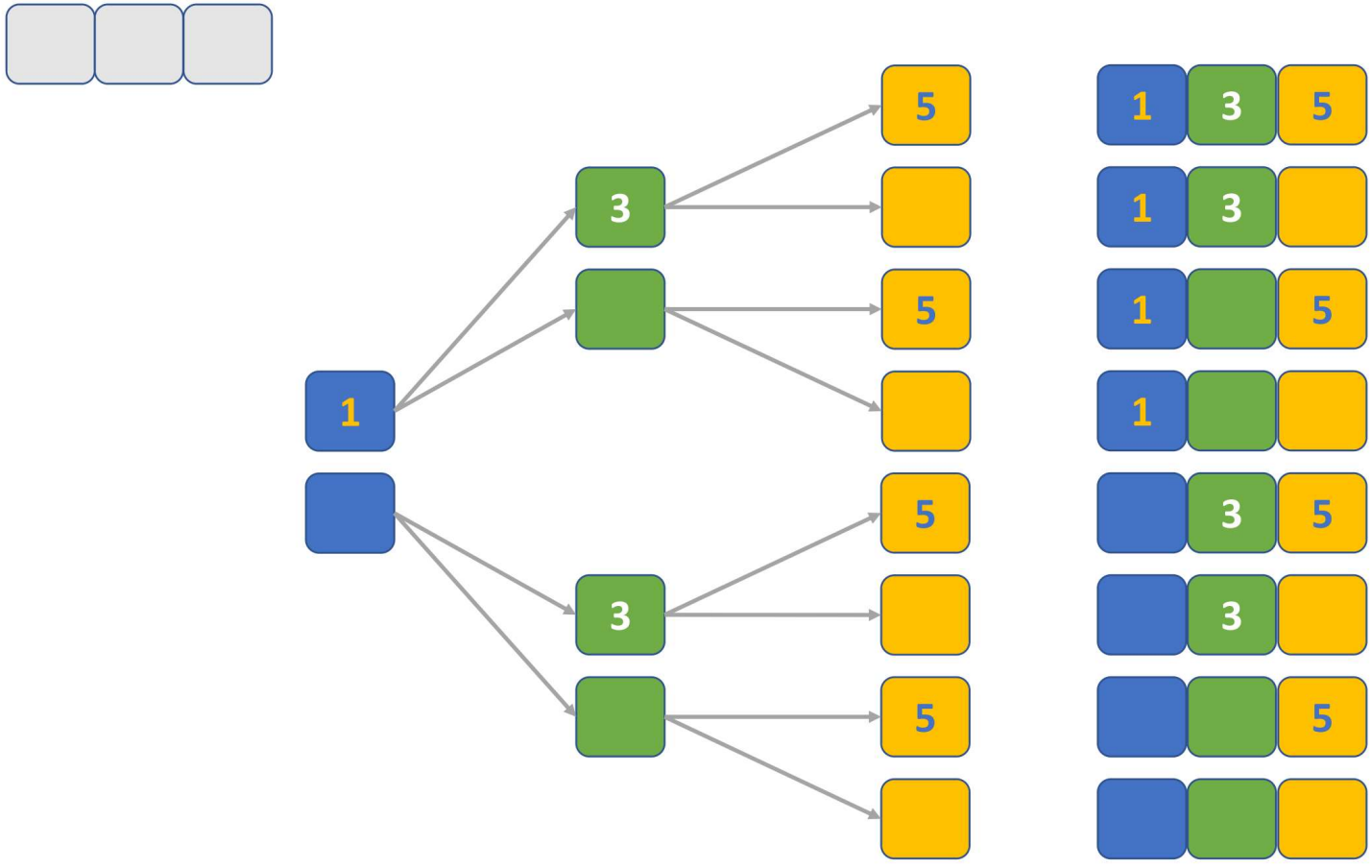

5
1
3
5
3

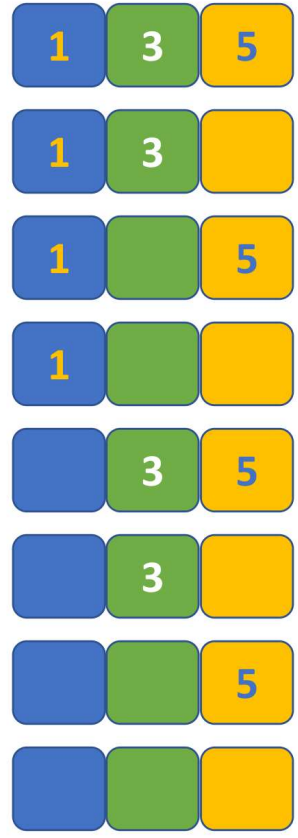

5
1
3
5
5
3


5


1
3
3
5
5
5


5
3
5
1
5
3
5


1
3
5
3
5
5
1
3
5
1
3
1
5
1


1 3 5
1 3
1 5
1
3 5
3
5

递归步骤

递归步骤

1. 最简单的情况

递归步骤

1. 最简单的情况
2. 合理降低问题规模

递归步骤

1. 最简单的情况
2. 合理降低问题规模
3. 进行递归调用

```
 1 void listSubsetsRec(Vector<int> elems, Vector<int> soFar) {
 2     // 1. Base Case
 3     if (elems.isEmpty()) {
 4         cout << soFar << endl;
 5     } else {
 6         // 2. Reducing
 7         int item = elems[0];
 8         Vector<int> remains = elems.subList(1);
 9         // 3.1 Include Recursive Case
10         listSubsetsRec(remains, soFar + item);
11         // 3.2 Exclude Recursive Case
12         listSubsetsRec(remains, soFar);
13     }
14 }
15
```

小试牛刀

将上述递归函数的参数修改为引用类型

今日话题

- ~~回顾~~
- ~~递归子集问题~~

THE END