

Data Structure & Algorithm

Algorithm

Backtracking - Quay Lui

1. Định nghĩa

Backtracking = Brute Force + Recursion

- ❖ Là bài toán **duyệt trâu** thông thường (**Brute Force**)
- ❖ Thường được cài đặt bằng **Đệ quy** (**Recursion**)

It's nothing NEW!

WARNING

2. Đặc điểm

- ❖ Là bài toán **duyệt trâu** => **Độ phức tạp** không thay đổi
- ❖ Cài đặt bằng **đệ quy** => Chương trình **ngắn gọn, dễ hiểu**
- ❖ **Thích hợp** với các bài toán **liệt kê, tổ hợp, chỉnh hợp**.

3. Ý tưởng

❖ Bài toán đặc trưng / cơ sở:



Cho một bộ **5 hộp** màu nước.

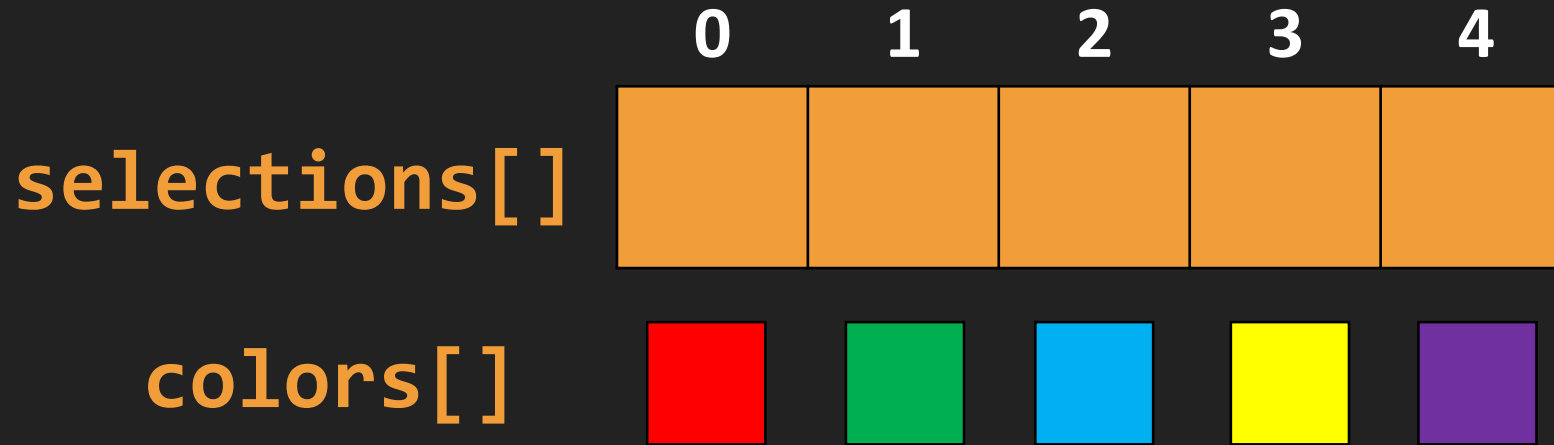
Hỏi **có thể tạo ra bao nhiêu màu** từ 5 hộp màu đó?

➤ Bài toán **tổ hợp**

➤ Bài toán **tổng quát**:

Cho tập **n phần tử**, hỏi có **bao nhiêu tập con**?

3. Ý tưởng



- `selections[i] = 1`: Hộp màu ***i*** được chọn
- `selections[i] = 0`: Hộp màu ***i*** không được chọn

3. Ý tưởng

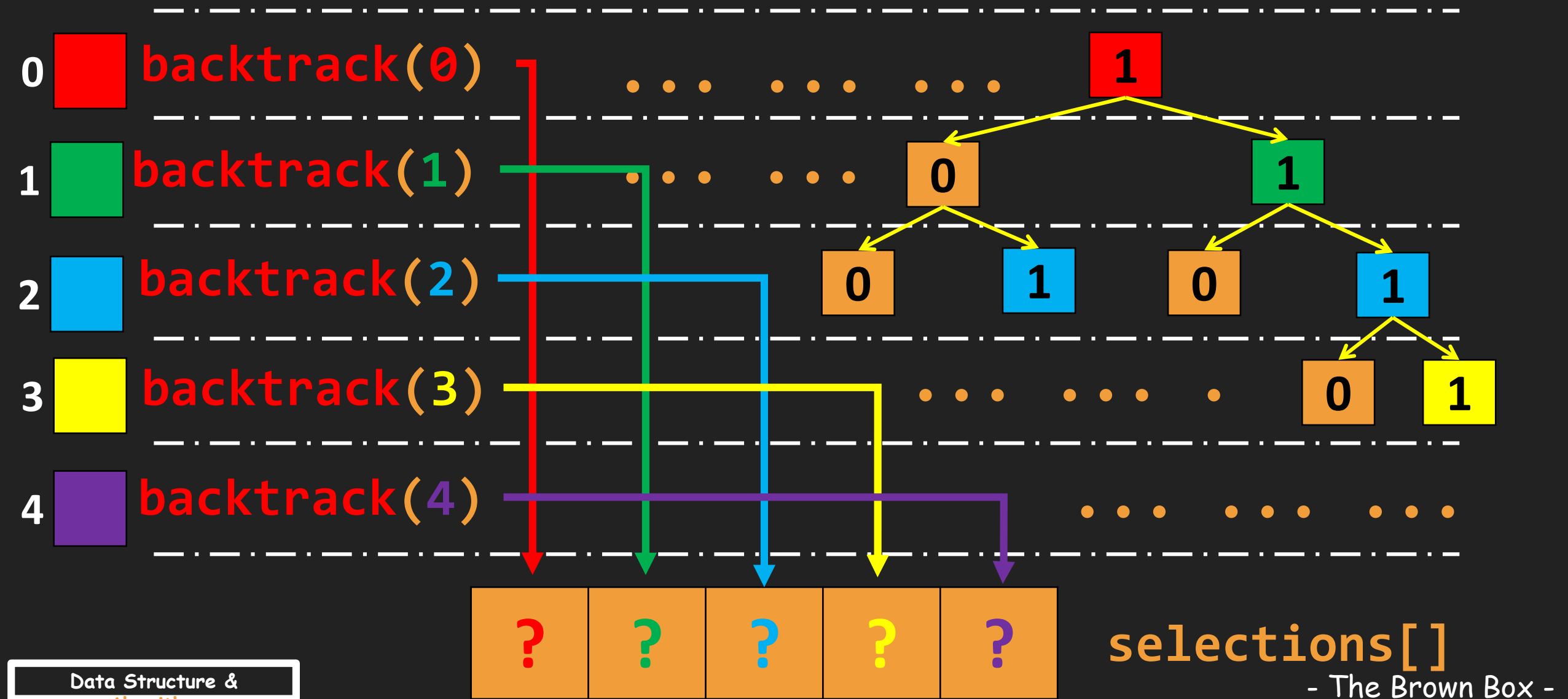
	0	1	2	3	4
<code>selections[]</code>	0	1	0	1	0
	X		X		X

	0	1	2	3	4
<code>selections[]</code>	1	1	0	0	1
			X	X	

```
selections[]
```



3. Ý tưởng



4. Giải thuật

```
// backtrack(i) tìm giá trị cho phần tử thứ i
trong cấu hình selections[]
backtrack(i)
{
    thử với selections[i] với {0, 1}
    với mỗi giá trị của selections[i] :

    - Nếu i là phần tử cuối cùng:
      => Xử lý selections thu được
    - Nếu i chưa là phần tử cuối cùng:
      => backtrack(i+1)
}
```

➤ Chạy bằng “cơ m”

5. Cài đặt

6. Tổng quát

```
// backtrack(i) tìm giá trị cho phần tử thứ i
backtrack(i)
{
    thử với cau_hinh[i] với {valid_values}
    với mỗi giá trị của cau_hinh[i] :

        - Nếu i là phần tử cuối cùng:
          => Xử lý cau_hinh thu được
        - Nếu i chưa là phần tử cuối cùng:
          => backtrack(i+1)
}
```

- ✓ **Cấu hình** có thể thay đổi (mảng 1 chiều, 2 chiều, ...)
- ✓ **Tập giá trị** có thể khác nhau với mỗi phần tử
 - ✓ Có thể không cần chạy đến **cuối cùng** để **cắt tỉa** những **nhánh** biết chắc rằng sẽ **không thoả mãn**.

6. Thực chiến



Daily **LeetCode** Challenge

67 Easy

1863. Sum of All Subset XOR Totals

Tags: **#backtracking** **#recursion** **#BitManipulation**

- The Brown Box - hoangvancong.com



Daily **LeetCode** Challenge

68 Easy

401. Binary Watch

Tags: **#Backtracking** **#Recursion**

- The Brown Box - hoangvancong.com

✓ Tìm bài tập với hashtag: **#backtracking**