

# Báo Cáo Tuần 1 – Thực Hành DSA

## Bài 1:

- Sử dụng đệ quy với công thức là  $\text{fibonacci}(n) = \text{fibonacci}(n-1) + \text{fibonacci}(n-2)$
- Basement là  $n == 0$  thì trả về 0 và  $n = 1$  thì trả về 1

## Bài 2:

- Sử dụng đệ quy với công thức là  $\text{factorial}(n) = \text{factorial}(n-1) * n$
- Basement là  $n == 0$  thì trả về 1 vì  $0! = 1$

## Bài 3:

- Ý tưởng xây dựng là tạo lần lượt các strings gồm các char được bổ sung liên tục:
  - o Ta gọi hàm để + '0' hoặc '1' vào string bất kì
  - o Khi độ dài của string == n thì hoàn thành 1 xâu nhị phân
  - o Để xây dựng xâu nhị phân theo thứ tự, ta tạo str + '0' trước sau đó đến 1. Bởi vì 2 hàm generate đều được gọi lồng trong 1 hàm mỗi khi giảm n - 1 đi, hay nói cách khác là generate thêm 1 char nên thứ tự cũng được xen kẽ.
- Debug với n = 2 để hình dung:
  - o Ta thấy str = "" -> str = "0" -> str = "00" ( sẽ cout ra nhưng không ảnh hưởng tiến trình đệ quy) -> str = "01" ( tương tự )str = "1" -> str = "10" và "11" .Việc đệ quy được gọi chung với nhau 2 lần liên tiếp làm việc mô hình hóa trở nên phức tạp với việc qua mỗi lần đệ quy -> ở dưới sẽ xuất hiện gấp đôi.
- Ý tưởng gốc vẫn là chọn 0 hoặc 1 để cộng vào mảng nên khá đơn giản. Giả sử là mã thập lục phân thì vẫn như vậy với 16 hàm lồng vào mỗi lần đệ quy.

## Bài 4:

- Bài toán sử dụng tư tưởng đệ quy thuần với việc phải dịch n-1 tầng trước khi xử lý tầng thứ n. Vì vậy ta đệ quy và gọi mỗi lần dịch chuyển 1 tầng. Việc có 2 hàm được gọi trong là vì phải dịch qua 1 cột trung gian trước khi dịch về lại cột gốc. Vì vậy về bản chất ta vẫn chỉ gọi 2 phần độc lập, chứ không lồng nhau.
- Nguồn tham khảo chính: <https://www.geeksforgeeks.org/c-program-for-tower-of-hanoi/> .
- Cùng với kiến thức được học trong môn toán rời rạc ở HK1

## Bài 5:

```
- bool isSorted(int arr[], int n) {  
-     // Base: khi n về 0 hoặc 1 thì có nghĩa là hàm đã được  
-     sắp xếp  
-     if (n == 1 || n == 0) return true;  
-     // Chỉ so sánh phần tử đầu của mảng  
-     if (arr[0] > arr[1]) return false;  
-  
-     // Đệ quy với phần tử giảm về số lượng, và check các phần tử  
-     tiếp theo  
-     return isSorted(arr + 1, n - 1);  
- }  
- Ý tưởng phần code này của thầy trực quan và đúng tinh thần  
- đệ quy nên em xin phép sử dụng luôn ạ. Em cam đoan mình đã  
- đọc kĩ và suy nghĩ kĩ chứ không phải copy vào cho có ạ.  
- Code ngắn và rất dễ hình dung ạ
```

## Bài 6:

- Đầu tiên là hàm kiểm tra IsSafe, tạo ra để kiểm tra xem yêu cầu đề bài, với tính chất an toàn là không nằm trên vùng đã xuất hiện của 1 quân hậu khác. Ta kiểm tra điều này với 3 lệnh if và thao tác trên mảng 2 chiều ( vì chạy theo đường chéo nên biến tọa độ j phụ thuộc theo i).

- Hàm quan trọng sử dụng BackTracking :

Ý tưởng xây dựng ban đầu vẫn là chạy theo từng hàng cho nên ta tiếp cận hàm đệ quy theo hướng đó luôn với biến row là biến chặn và thoát hàm khi row == n ( basement )

Trong quá trình thực hiện ta lặp lại việc tìm kiếm các vị trí có thể đặt khác cùng với vòng for song song với đó là kiểm tra quay lui. Bởi lẽ khi nhánh đệ quy ở hàm tiếp theo hoàn tất hàm sẽ quay lại hàng hiện tại và xóa con hậu vừa gán đi và gán lại bằng 0 ( để thực hiện cho lần kiểm tra tiếp theo ) bởi lẽ vị trí đó vẫn là địa điểm an toàn ( theo hàm isSafe đã kiểm tra ).

Hay đơn giản là:

Tiến lên phía trước: Thử đặt quân hậu ở một cột của hàng hiện tại, nếu an toàn thì tiến tới hàng tiếp theo.

**Quay lui: Nếu không thể đặt quân hậu ở bất kỳ cột nào của hàng tiếp theo mà không gây xung đột, quay lại hàng trước đó và thử cột khác.**

**Quá trình này lặp lại cho đến khi thử hết tất cả các khả năng, đảm bảo tìm được mọi giải pháp hợp lệ.**

[https://github.com/BaoNTfit/TH\\_DSA/tree/main/24120023\\_Week01](https://github.com/BaoNTfit/TH_DSA/tree/main/24120023_Week01)

BaoNTfit / TH\_DSA

<> Code

Issues

Pull requests

Actions

Projects

Wiki

Security

Insights

Settings

Files

main

Go to file

24120023\_Week01

.vscode

ex1.cpp

ex2.cpp

ex3.cpp

ex4.cpp

ex5.cpp

ex6.cpp

24120023\_Week02

TH\_DSA / 24120023\_Week01

Add file

BaoNTfit Hoan Thanh Bai

Off4452 · yesterday History

Name	Last commit message	Last commit date
..		
.vscode	Hoan Thanh Bai	yesterday
ex1.cpp	Hoan Thanh Bai	yesterday
ex2.cpp	Hoan Thanh Bai	yesterday
ex3.cpp	Hoan Thanh Bai	yesterday
ex4.cpp	Hoan Thanh Bai	yesterday
ex5.cpp	Hoan Thanh Bai	yesterday
ex6.cpp	Hoan Thanh Bai	yesterday

-