**BÁO CÁO TUẦN 1**

Nguyễn Văn Bắc

# Kiến thức đã học

## Kiến thức cơ bản về lập trình Java

## Những hiểu biết cơ bản về *Thiết kế và đánh giá thuật toán*

* Khái niệm ***Độ phức tạp về thời gian*** *(Time Complexity)* và ***Độ phức tạp về không gian*** *(Space Complexity).*

+ Độ phức tạp không đổi O(1)

+ Độ phức tạp Logarit O(logn)

+ Độ phức tạp tuyến tính O(n)

+ Độ phức tạp bậc 2: O(n2)

+ Độ phức tạp khối: O(n3)

## Một số thuật toán đã tìm hiểu

* Phương pháp ***Chia để trị*** *(Divide and conquer).*

Dựa trên ý tưởng chia nhỏ bài toán để xử lý và tổng hợp lại để nhận được đầu ra mong muốn.

Gồm 3 bước : Chia (Divide) , Trị (Conquer) và Kết hợp (Combine)

Một số thuật toán sử dụng phương pháp *Chia để trị* : Merge sort, Quick sort, Binary search, …

* **Đệ quy**.

Là phương pháp giải quyết bài toán thành phiên bản nhỏ hơn của chính nó, thường sử dụng kết quả của những bài toán nhỏ hơn này để xử lý chính bài toán lớn đó.

Yếu cầu :

Trường hợp cơ sở (Base case) : có thể giải quyết trực tiếp và trả về kết quả.

Gọi đệ quy (Recursive case) : gọi lại chính nó với một biến thể nhỏ hơn bài toán ban đầu.

* Phương pháp xây dựng và các thao tác trên ***Cây đỏ đen*** *(Red – Black Tree).*

Dựa trên ý tưởng giải quyết những hạn chế mà cây nhị phân (BST) chưa giải quyết được.

Các ràng buộc đối với cây đỏ đen (Red – Black Tree) :

* + 1. Một node là đỏ thì không có cha hoặc con là node đỏ
    2. Node root luôn đen
    3. Mọi đường đi từ một node đến bất kì lá nào của nó đều đi qua số lượng node đen bằng nhau.

Các phép quay trong cây đỏ đen:

-Quay trái (Left rotation)

-Quay phải (Right rotation)

Chèn một node mới vào cây (Insert):

Xóa một node trong cây:

## Hoạt động luyện tập

* Triển khai một số thuật toán sắp xếp như ***Bubble sort, Insertion sort, Merge sort*** bằng Java.
* Triển khai bài toán ***Binary Seach***  bằng Java.
* Triển khai bài toán ***Tháp Hà Nội*** *(Hanoi Tower)* bằng Java.