**ĐÂY LÀ BUỔI 1/48 BUỔI HỌC CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT**

1. **NỘI DUNG & NGUỒN TÀI LIỆU:**

**Phần 1: Giới thiệu về Cấu trúc Dữ liệu và Giải thuật**

**1.1 Cấu trúc dữ liệu là gì?**

* Định nghĩa cấu trúc dữ liệu và tầm quan trọng trong lập trình.
* Ví dụ về một số cấu trúc dữ liệu phổ biến:
  + Mảng (Array)
  + Danh sách liên kết (Linked List)
  + Ngăn xếp (Stack) & Hàng đợi (Queue)
  + Cây (Tree) & Đồ thị (Graph)
  + Bảng băm (Hash Table)

**1.2 Giải thuật là gì?**

* Giải thuật là gì?
* Mối quan hệ giữa giải thuật và cấu trúc dữ liệu.
* Các loại thuật toán phổ biến:
  + Thuật toán tìm kiếm (Search)
  + Thuật toán sắp xếp (Sort)
  + Thuật toán đệ quy (Recursion)
  + Thuật toán tham lam (Greedy)
  + Thuật toán chia để trị (Divide and Conquer)
  + Thuật toán quy hoạch động (Dynamic Programming)

Phần 2: Giới thiệu về Big O Notation

**2.1 Tại sao cần đánh giá độ phức tạp thuật toán?**

* Giải thích lý do cần phân tích độ phức tạp thuật toán (tối ưu hiệu suất, giảm thời gian chạy).
* So sánh hai thuật toán giải cùng một bài toán nhưng có hiệu suất khác nhau.

**2.2 Big O Notation là gì?**

* Định nghĩa **Big O Notation**.
* Cách xác định Big O của một thuật toán.
* Các loại độ phức tạp phổ biến:
  + **O(1) - Constant Time**
  + **O(log n) - Logarithmic Time**
  + **O(n) - Linear Time**
  + **O(n log n) - Linearithmic Time**
  + **O(n²) - Quadratic Time**
  + **O(2ⁿ) - Exponential Time**
  + **O(n!) - Factorial Time**

**Phần 3: Phân tích Big O của một số thuật toán cơ bản (50 phút)**

**3.1 Phân tích thuật toán tìm kiếm**

* **Tìm kiếm tuyến tính (Linear Search) – O(n)**
* **Tìm kiếm nhị phân (Binary Search) – O(log n)**

**3.2 Phân tích thuật toán sắp xếp**

* **Bubble Sort – O(n²)**
* **Merge Sort – O(n log n)**
* **Quick Sort – O(n log n) (trung bình)**