

Cấu trúc dữ liệu (Data Structure) là gì ?

Cấu trúc dữ liệu là cách lưu trữ, tổ chức dữ liệu có thứ tự, có hệ thống để dữ liệu có thể được sử dụng một cách hiệu quả.

Dưới đây là hai khái niệm nền tảng hình thành nên một cấu trúc dữ liệu:

- **Interface:** Mỗi cấu trúc dữ liệu có một Interface. Interface biểu diễn một tập hợp các phép tính mà một cấu trúc dữ liệu hỗ trợ. Một Interface chỉ cung cấp danh sách các phép tính được hỗ trợ, các loại tham số mà chúng có thể chấp nhận và kiểu trả về của các phép tính này.
- **Implementation** (có thể hiểu là sự triển khai): Cung cấp sự biểu diễn nội bộ của một cấu trúc dữ liệu. Implementation cũng cung cấp phần định nghĩa của giải thuật được sử dụng trong các phép tính của cấu trúc dữ liệu.
- Đặc điểm của một Cấu trúc dữ liệu
- **Chính xác:** Sự triển khai của Cấu trúc dữ liệu nên triển khai Interface của nó một cách chính xác.
- **Độ phức tạp về thời gian (Time Complexity):** Thời gian chạy hoặc thời gian thực thi của các phép tính của cấu trúc dữ liệu phải là nhỏ nhất có thể.
- **Độ phức tạp về bộ nhớ (Space Complexity):** Sự sử dụng bộ nhớ của mỗi phép tính của cấu trúc dữ liệu nên là nhỏ nhất có thể.

Tại sao Cấu trúc dữ liệu là cần thiết ?

Ngày nay, các ứng dụng ngày càng phức tạp và lượng dữ liệu ngày càng lớn với nhiều kiểu đa dạng. Việc này làm xuất hiện 3 vấn đề lớn mà mỗi lập trình viên phải đối mặt:

- **Tìm kiếm dữ liệu:** Giả sử có 1 triệu hàng hóa được lưu giữ vào trong kho hàng hóa. Và giả sử có một ứng dụng cần để tìm kiếm một hàng hóa. Thì mỗi khi thực hiện tìm kiếm, ứng dụng này sẽ phải tìm kiếm 1 hàng hóa trong 1 triệu hàng hóa. Khi dữ liệu tăng lên thì việc tìm kiếm sẽ càng trở lên chậm và tốn kém hơn.
- **Tốc độ bộ vi xử lý:** Mặc dù bộ vi xử lý có tốc độ rất cao, tuy nhiên nó cũng có giới hạn và khi lượng dữ liệu lên tới hàng tỉ bản ghi thì tốc độ xử lý cũng sẽ không còn được nhanh nữa.
- **Đa yêu cầu:** Khi hàng nghìn người dùng cùng thực hiện một phép tính tìm kiếm trên một Web Server thì cho dù Web Server đó có nhanh đến mấy thì việc phải xử lý hàng nghìn phép tính cùng một lúc là thực sự rất khó.

Để xử lý các vấn đề trên, các cấu trúc dữ liệu là một giải pháp tuyệt vời. Dữ liệu có thể được tổ chức trong cấu trúc dữ liệu theo một cách để khi thực hiện tìm kiếm một phần tử nào đó thì dữ liệu yêu cầu sẽ được tìm thấy ngay lập tức.

Độ phức tạp thời gian thực thi trong cấu trúc dữ liệu và giải thuật

Có 3 trường hợp thường được sử dụng để so sánh thời gian thực thi của các cấu trúc dữ liệu khác nhau:

- Trường hợp xấu nhất (Worst Case): là tình huống mà một phép tính của cấu trúc dữ liệu nào đó tốn thời gian tối đa (thời gian dài nhất). Ví dụ với ba số 1, 2, 3 thì nếu sắp xếp theo thứ tự giảm dần thì thời gian thực thi sẽ là dài nhất (và đây là trường hợp xấu nhất); còn nếu sắp xếp theo thứ tự tăng dần thì thời gian thực thi sẽ là ngắn nhất (và đây là trường hợp tốt nhất).
- Trường hợp trung bình (Average Case): miêu tả thời gian thực thi trung bình một phép tính của một cấu trúc dữ liệu.
- Trường hợp tốt nhất (Best Case): là tình huống mà thời gian thực thi một phép tính của một cấu trúc dữ liệu là ít nhất. Ví dụ như trên.

Thuật ngữ cơ bản trong Cấu trúc dữ liệu

- Dữ liệu: Dữ liệu là các giá trị hoặc là tập hợp các giá trị.
- Phần tử dữ liệu: Phần tử dữ liệu là một đơn vị đơn lẻ của giá trị.
- Các phần tử nhóm: Phần tử dữ liệu mà được chia thành các phần tử con thì được gọi là các phần tử nhóm.
- Các phần tử cơ bản: Phần tử dữ liệu mà không thể bị chia nhỏ thành các phần tử con thì gọi là các phần tử cơ bản.
- Thuộc tính và Thực thể: Một thực thể là cái mà chứa một vài thuộc tính nào đó, và các thuộc tính này có thể được gán các giá trị.
- Tập hợp thực thể: Các thực thể mà có các thuộc tính tương tự nhau thì cấu thành một tập hợp thực thể.
- Trường: Trường là một đơn vị thông tin cơ bản biểu diễn một thuộc tính của một thực thể.
- Bản ghi: Bản ghi là một tập hợp các giá trị trường của một thực thể đã cho.
- File: Là một tập hợp các bản ghi của các thực thể trong một tập hợp thực thể đã cho.