BÁO CÁO

Bùi Huỳnh Tây
MSSV: 24521589

25/03/2025

—

Cấu trúc dữ liệu và giải thuật -
IT003.P21.CTTN

—

GV Lý Thuyết: Duy Lê

GV Lý Thuyết: Nguyễn Thanh Sơn

GV Lý Thuyết: Văn Thái Hùng

GV HDTH: Trần Đình Khang

Phân tích các ứng dụng thực tế của các cấu trúc dữ liệu: Singly Linked List, Doubly Linked List, Stack, Queue

1. Singly Linked List

a. Ứng dụng thực tế có giá trị cao:

- Quản lý danh sách người dùng (User list) trong các hệ thống mạng xã hội
- Hệ thống điều hướng tuyến tính như lịch sử học tập trong nền tảng học trực tuyến

b. Giải thích:

- Trong các hệ thống như Facebook hay Instagram, dữ liệu người dùng có thể được tổ chức và duyệt tuần tự (dạng dòng thời gian). Singly Linked List giúp thao tác thêm hoặc xóa người dùng khỏi danh sách một cách hiệu quả mà không cần dịch chuyển các phần tử khác như mảng.
- Các bài học hoặc bài kiểm tra được lưu theo trình tự thời gian. Singly Linked List phù hợp vì chỉ cần duyệt theo một chiều, và việc thêm bài học mới vào cuối danh sách rất thuận tiện.

c. Cách sử dụng:

```
struct Node {
    string username;
    Node* next;
};

void insertAtEnd(Node*& head, string name) {
    Node* newNode = new Node{name, nullptr};
    if (!head) {
        head = newNode;
        return;
    }
    Node* temp = head;
    while (temp->next) temp = temp->next;
    temp->next = newNode;
}
```

2. Doubly Linked List

a. Ứng dụng thực tế có giá trị cao:

- Trình duyệt web (Browser) – chức năng Back/Forward
- Ứng dụng nghe nhạc – duyệt danh sách nhạc có thể tới lui

b. Giải thích:

- Khi người dùng nhấn "Back" hoặc "Forward", hệ thống cần di chuyển lui hoặc tới giữa các trang đã duyệt. Doubly Linked List rất phù hợp vì có thể duyệt cả hai chiều.
- Người dùng có thể chuyển bài trước hoặc sau trong playlist – danh sách nhạc cần được quản lý hai chiều để hỗ trợ chuyển hướng linh hoạt.

c. Cách sử dụng:

```
struct DNode {
    string page;
    DNode* prev;
    DNode* next;
};

void goBack(DNode*& current) {
    if (current && current->prev)
        current = current->prev;
}

void goForward(DNode*& current) {
    if (current && current->next)
        current = current->next;
}
```

3. Stack

a. Ứng dụng thực tế có giá trị cao:

- Undo/Redo trong phần mềm soạn thảo văn bản
- Chuyển đổi biểu thức (infix → postfix) trong trình biên dịch

b. Giải thích:

- Mỗi thao tác của người dùng được lưu vào stack. Khi nhấn "Undo", chương trình chỉ cần pop thao tác gần nhất để hoàn tác.
- Khi biên dịch code, cần chuyển đổi biểu thức có dấu ngoặc. Stack giúp kiểm tra dấu ngoặc đúng hay sai, và hỗ trợ đánh giá thứ tự toán tử.

c. Cách sử dụng:

```
stack<string> history;

void undo() {
    if (!history.empty()) {
        string lastAction = history.top();
        history.pop();
        // Thực hiện hoàn tác dựa trên lastAction
    }
}
```

4. Queue

a. Ứng dụng thực tế có giá trị cao:

- Hệ thống hàng đợi khách hàng trong ngân hàng, bệnh viện
- Tác vụ in tài liệu trong máy in

b. Giải thích:

- Dữ liệu được xử lý theo thứ tự đến trước xử lý trước (FIFO). Queue là cấu trúc lý tưởng cho loại vấn đề này.
- Các tài liệu gửi đến máy in được xếp vào queue và in tuần tự.

c. Cách sử dụng:

```
queue<string> printJobs;

void addJob(string file) {
    printJobs.push(file);
}

void processJob() {
    if (!printJobs.empty()) {
        string currentJob = printJobs.front();
        printJobs.pop();
        // Gửi currentJob đến máy in
    }
}
```

Tài liệu tham khảo

1. Cormen, T.H., Leiserson, C.E., Rivest, R.L., & Stein, C. (2009). *Introduction to Algorithms*. MIT Press.
2. GeeksforGeeks: https://www.geeksforgeeks.org/data-structures/