

CHALLENGE 1: DANH SÁCH ĐA LIÊN KẾT

1 Nội dung

1.1 Tìm hiểu

1. Tìm hiểu các cấu trúc dữ liệu(CTDL): **Sorted Singly Linkedlist, Skiplist, Indexable Skiplist**. Đối với mỗi loại cấu trúc dữ liệu nêu trên, thực hiện các yêu cầu sau:

- Định nghĩa và dùng hình ảnh minh họa rõ.
- Mô tả các thao tác:
 - Tìm kiếm một giá trị cho trước
 - Thêm một giá trị cho trước
 - Xóa một giá trị cho trước(sử dụng mã giả hoặc liệt kê từng bước).
- Đối với cấu trúc dữ liệu Skiplist, hãy cho biết ứng dụng thực tế của CTDL này.

2. Tìm hiểu và trả lời các câu hỏi dưới đây:

- (a) Hãy cho biết khái niệm MAX-LEVEL của một Skiplist. Tại sao một Skiplist lại cần MAX-LEVEL?
 - (b) Level của một node chuẩn bị được thêm vào một skiplist cho trước được chọn dựa trên yếu tố nào: giá trị của node đó hay vị trí nó chuẩn bị được chèn vào? Giải thích.
 - (c) Đề xuất một thuật toán đơn giản (không ngẫu nhiên và khác với thuật toán được đề xuất ở phần **1.2**) xác định level của một node (biết trước giá trị) khi thêm node này vào Skiplist. Giải thích lý do bạn chọn đề xuất thuật toán này.
3. (a) Hãy cho biết độ phức tạp phù hợp (thông qua bigO) của các thao tác trên các dạng danh sách đa liên kết bằng việc điền vào **Bảng 1** dưới đây.

Thao tác	Sorted Singly Linkedlist	Skiplist	Indexable Skiplist
Tìm kiếm vị trí của một giá trị cho trước	(1)	(4)	(7)
Thêm 1 giá trị vào danh sách	(2)	(5)	(8)
Xóa 1 giá trị khỏi danh sách	(3)	(6)	(9)

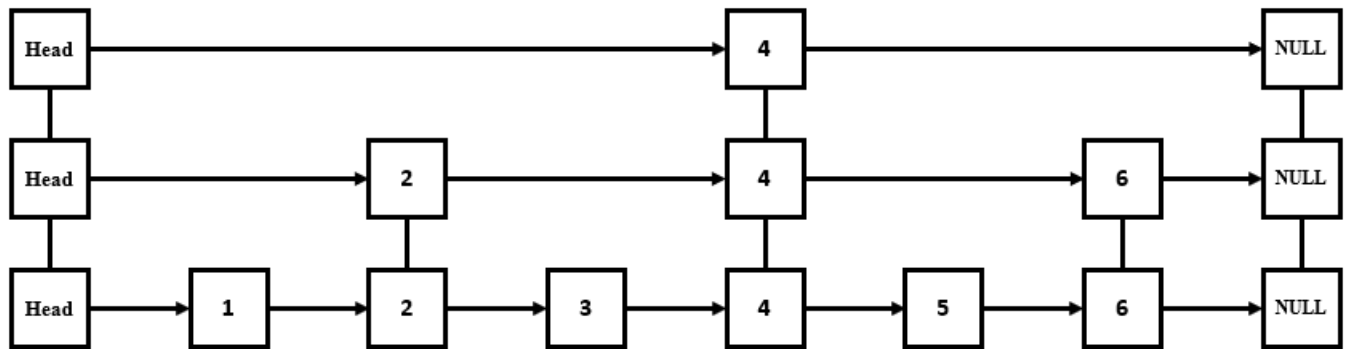
Bảng 1

- (b) Hãy lý giải độ phức tạp ở các thao tác (2), (3), (4), (5), (6), (7), (8), (9) đã được điền ở **Bảng 1**.
- (c) Dựa vào độ phức tạp đã được điền ở **Bảng 1**, hãy so sánh và giải thích sự khác biệt (nếu có) về độ phức tạp giữa các bộ thao tác:
- (1), (4) và (7)
 - (2), (5) và (8)

1.2 Cài đặt

Định nghĩa cấu trúc của một Skiplist chứa các giá trị nguyên dương và cài đặt các thao tác sau trên CTDL vừa định nghĩa:

- Tìm kiếm và trả về thông tin của một Node chứa giá trị cho trước.
- Thêm một giá trị(val) cho trước vào Skiplist có sẵn:
 - $\text{MAX-LEVEL} = 0.2 * \text{kích thước Skiplist hiện tại}$ (tối thiểu là 1).
 - $\text{Level} = (1 + \text{số lần dịch phải } (\gg) \text{ của val để đạt được 1 số lẻ}) \bmod \text{MAX-LEVEL}$. (tối thiểu là 1).
 VD: $\text{Level}(3) = 1, \text{Level}(4) = 3, \text{Level}(6) = 2, \dots$
- Xóa một giá trị(val) cho trước khỏi Skiplist có sẵn.



Hình 1

Chương trình cần được xây dựng để liên tục nhận, thao tác và xuất thông tin ra màn hình console những thông tin tương ứng với các câu lệnh sau:

Lệnh 1: `new S x_1 x_2 ... x_S`

- Xóa Skiplist hiện tại (nếu có).
- Xây dựng Skiplist mới với kích thước S bằng cách thêm lần lượt các giá trị x_1, x_2, \dots, x_S vào Skiplist rỗng ban đầu.
- **Console output:** `List built`

Lệnh 2: `Print`

- In ra Skiplist hiện tại theo từng level.
- **Console output:**
 - Nếu Skiplist rỗng: `List is empty`
 - Nếu Skiplist không rỗng:


```

Level ...: ...
          
```
 - Ví dụ (Hình 1):


```

Level 3:          4
Level 2:    2      4      6
Level 1:  1  2  3  4  5  6
          
```

Lệnh 3: `Size`

- In ra kích thước Skiplist hiện tại theo từng level.
- **Console output:**
 - Nếu Skiplist rỗng: `List is empty`

- Nếu Skiplist không rỗng:
 - Max Level: ...
 - size of Level: ...
- Ví dụ (Hình 1):
 - Max Level: 3
 - size of Level 1: 6
 - size of Level 2: 3
 - size of Level 3: 1

Lệnh 4: Search val

- Tìm node mang giá trị `val` trong Skiplist hiện tại.
- **Console output:**
 - Nếu giá trị `val` không xuất hiện trong Skiplist: `val not found`
 - Nếu giá trị `val` xuất hiện trong Skiplist:
 - Level of `val`: ...
 - Traverse path: ...
 - Ví dụ (Hình 1): (Giả sử thêm node mang giá trị 6 vào danh sách)
 - Level of 6: 2
 - Traverse path: 2 4 6

Lệnh 5: Insert val

- Thêm giá trị (`val`) vào Skiplist hiện tại (bao gồm cả Skiplist rỗng).
 - **Console output:**
 - Nếu giá trị `val` đã tồn tại: `val already inserted`.
 - Nếu giá trị `val` chưa tồn tại và đã được thêm vào:
 - `val inserted with level ...`
- với `level` là level đã được tính toán của node mang giá trị `val`.

Lệnh 6: Remove val

- Xóa giá trị (`val`) khỏi Skiplist hiện tại.
- **Console output:**
 - Nếu giá trị `val` không xuất hiện trong Skiplist: `val not found`
 - Nếu giá trị `val` xuất hiện trong Skiplist: `val removed`

2 Yêu cầu và Quy định nộp bài

2.1 Yêu cầu

- Phần Challenge là phần bài tập không bắt buộc. Điểm số của bài Challenge này sẽ được cộng thêm vào điểm tổng kết môn học.
- Sẽ có 3 bài Challenge xuyên suốt quá trình học. Sinh viên làm các bài Challenge này theo nhóm 3 người.
- Một bài Challenge hợp lệ (Được chấm và tính điểm cộng) cần thỏa mãn các điều kiện sau:
 - Nằm trong số 13 bài nộp đầu tiên (tính theo mốc thời gian của lần nộp bài cuối cùng). Link nộp bài sẽ tự động đóng khi đã đủ nhóm nộp hoặc hết thời hạn nộp bài.
 - Bài làm cần đảm bảo thể hiện đầy đủ các yêu cầu được giao. Không chấp nhận các bài làm thiếu sót một hoặc nhiều phần dưới bất kì hình thức nào.
 - Trình bày và thực hiện đầy đủ các quy định ở phần **Quy định nộp bài** bên dưới.
 - Đối với mỗi Challenge, sinh viên cần lập một nhóm 3 người mới, nghĩa là, 2 sinh viên bất kì chỉ được chung nhóm tối đa 1 Challenge.
- Giảng viên lý thuyết có quyền yêu cầu một trong các nhóm có bài nộp Challenge hợp lệ trình bày trước lớp trong buổi học lý thuyết bất kì.

2.2 Quy định nộp bài

- Mã nhóm là chuỗi kết hợp của 3 số cuối của mã số sinh viên của các thành viên trong nhóm, ngăn cách bởi dấu "_". Thành viên khóa lớn hơn đứng trước, nếu cùng khóa thì MSSV lớn hơn đứng trước. Ví dụ:
 - 1812999, 20120995, 20120994 \rightarrow 999_995_994
 - 20120999, 20120995, 20120994 \rightarrow 994_995_999

- Nhóm sinh viên nộp bài trên Moodle dưới dạng [**Mã_nhóm.rar**] hoặc [**Mã_nhóm.zip**], là file nén của thư mục [**Mã_nhóm**]. Thư mục này bao gồm:
 - File báo cáo được trình bày dưới dạng văn bản [**Report.pdf**] hoặc dạng slide [**Report.pptx**]. File này chứa nội dung tìm hiểu ở phần **1.1** và trình bày hướng giải quyết vấn đề lập trình ở phần **1.2**.
 - * Nếu bài nộp file báo cáo là dạng slide, cần giải thích cặn kẽ từng slide ở phần *Note* của slide đó.
 - * Trang đầu (hoặc slide đầu) của file báo cáo cần ghi rõ thông tin (Họ và tên, MSSV) của từng thành viên trong nhóm.
 - * File báo cáo cần trình bày có cấu trúc, logic, rõ ràng, mạch lạc, trả lời thẳng vào trọng tâm của câu hỏi. Độ dài của bài nộp không vượt quá 15 trang đối với file văn bản, và 30 trang đối với slide.
 - Thư mục lập trình **Source**. Thư mục này chỉ chứa các file *".cpp"* và *".h"*, nếu sinh viên có nộp kèm file có định dạng khác, cần giải thích rõ ràng công dụng của file này trong báo cáo. Yêu cầu mã nguồn cần trình bày logic, có chú thích rõ ràng.