

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN



CHALLENGE 2: TRIES

Môn: Cấu trúc dữ liệu và giải thuật

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN



CHALLENGE 2: TRIES

GVHD: Thầy Văn Chí Nam – Thầy Bùi Huy Thông

Mã nhóm: 471_473_480

Danh sách sinh viên thực hiện:

1. Trương Văn Hào _ 20120471
2. Dương Minh Hiếu _ 20120473
3. Võ Minh Hiếu_ 20120480

MỤC LỤC

MỤC LỤC	
DANH MỤC HÌNH.....	
DANH MỤC BẢNG	
Phần 1: Tìm Hiểu	1
1. Độ phức tạp của một số thao tác	1
2. Một số lợi thế của Tries so với các cấu trúc dữ liệu khác	1
2.1. Lợi thế của Tries với Binary Search Tree	1
2.2. Lợi thế của Tries với Hash Table	1
Phần 2: Trình bày hướng giải quyết	2
1. Cài đặt Tries	2
1.1. Định nghĩa 1 Node của Trie	2
1.2. Thêm một từ vào Trie	2
2. Xây dựng chương trình tạo từ tiếng Anh hợp lệ	2
TÀI LIỆU THAM KHẢO	4

DANH MỤC HÌNH

Không tìm thấy mục nhập nào của bảng hình minh họa.

DANH MỤC BẢNG

Table 1: Bảng BigO của các thao tác trên Tries 1

Phần 1: Tìm Hiểu

1. Độ phức tạp của một số thao tác

Thao tác	Độ phức tạp
Thêm 1 từ	$O(n)$ với n là chiều dài từ cần thêm
Xóa 1 từ	$O(n)$ với n là chiều dài từ cần xóa
Tìm 1 từ	$O(n)$ với n là chiều dài từ cần tìm
Tìm tất cả các từ cùng tiền tố có độ dài i	$O(26^n)$ và n là độ dài tiền tố

Table 1: Bảng BigO của các thao tác trên Tries

2. Một số lợi thế của Tries so với các cấu trúc dữ liệu khác

2.1. Lợi thế của Tries với Binary Search Tree

- Thời gian tìm kiếm ngắn. Chỉ mất $O(n)$ với n là độ dài chuỗi cần tìm.
- Sử dụng ít bộ nhớ hơn bởi các tiền tố chung chỉ cần được lưu trữ 1 lần
- Cho phép tìm kiếm tiền tố trùng hợp dài nhất
- Số lượng node từ gốc tới lá đúng bằng chiều dài của khóa

2.2. Lợi thế của Tries với Hash Table

- Thời gian tra cứu $O(k)$ có thể đoán trước trong đó k là kích thước của khóa
- Cho phép liệt kê các khóa theo thứ tự từ điển một cách dễ dàng
- Cho phép tìm kiếm tiền tố trùng hợp dài nhất
- Thực hiện các thao tác nhanh hơn với trường hợp số nguyên hay con trỏ
- Thao tác xóa đơn giản

Phần 2: Trình bày hướng giải quyết

1. Cài đặt Tries

1.1. Định nghĩa 1 Node của Trie

Gồm 2 phần:

- Mảng next gồm 26 **Node** con trỏ, lần lượt trỏ đến 26 **Node** ứng với các chữ cái từ **a** đến **z**.
- Biến **isWord** để kiểm tra xem có phải là kết thúc của 1 từ hay không.

```
struct trieNode
{
    trieNode* next[soKiTu];
    bool isWord;
};
```

1.2. Thêm một từ vào Trie

- Cho **Node p** bằng **Node Root** của **Trie**.
 - Duyệt từ chữ cái đầu tiên đến cuối cùng của từ muốn thêm vào:
 - Nếu con trỏ **next** ứng với chữ cái bằng **NULL**, ta tạo con trỏ mới.
 - Di chuyển **p** đến con trỏ **next** kế tiếp ứng với chữ cái.
- Đánh dấu biến **isBool** của **p** bằng 1 để biết đây là kết thúc của một từ.

2. Xây dựng chương trình tạo từ tiếng Anh hợp lệ

- **Bước 1:** Đọc file, tạo Trie từ các từ trong file.
- **Bước 2:** Nhập danh sách ký tự, tạo 1 mảng gồm 26 phần tử chứa số lượng xuất hiện của các ký tự từ a đến z (VD: a[0] chứa số lần xuất hiện của ký tự 'a').
- **Bước 3:** Tìm từ:
 - Bắt đầu tại **Node p** là **root** của **Tries**.

- Kiểm tra nếu tại **p** là kết thúc của từ và có hơn 3 kí tự. Thêm từ vào hàng đợi các chữ cái tìm được.
- Lặp **i** chạy từ **1** đến **26** (ứng với chữ cái từ **a** đến **z**):
 - Nếu **a[i] > 0** và **p ->next[i] != NULL** (chữ cái còn số lần xuất hiện và có thể thêm vào để tạo từ mới) thì :
 - Giảm **a[i]** xuống **1**.
 - Đệ quy với **p = p->next[i]** (tìm kí tự tiếp theo từ các chữ cái còn lại).
 - Tăng lại **a[i]** lên **1**.
- **Bước 4:** Xuất số lượng từ và danh sách các từ đã tạo được trong hàng đợi.

TÀI LIỆU THAM KHẢO

- [1] bharadwajrohan, “Coding,” 4 2015. [Trực tuyến]. Available: <https://thenextcode.wordpress.com/2015/04/12/trie-vs-bst-vs-hashtable/>. [Đã truy cập 22 11 2021].
- [2] N. R. T. Trung, “vnoi,” [Trực tuyến]. Available: <https://vnoi.info/wiki/algo/data-structures/trie.md>. [Đã truy cập 22 11 2021].
- [3] “Geeksforgeeks,” 11 8 2021. [Trực tuyến]. Available: <https://www.geeksforgeeks.org/trie-insert-and-search/>. [Đã truy cập 22 11 2021].