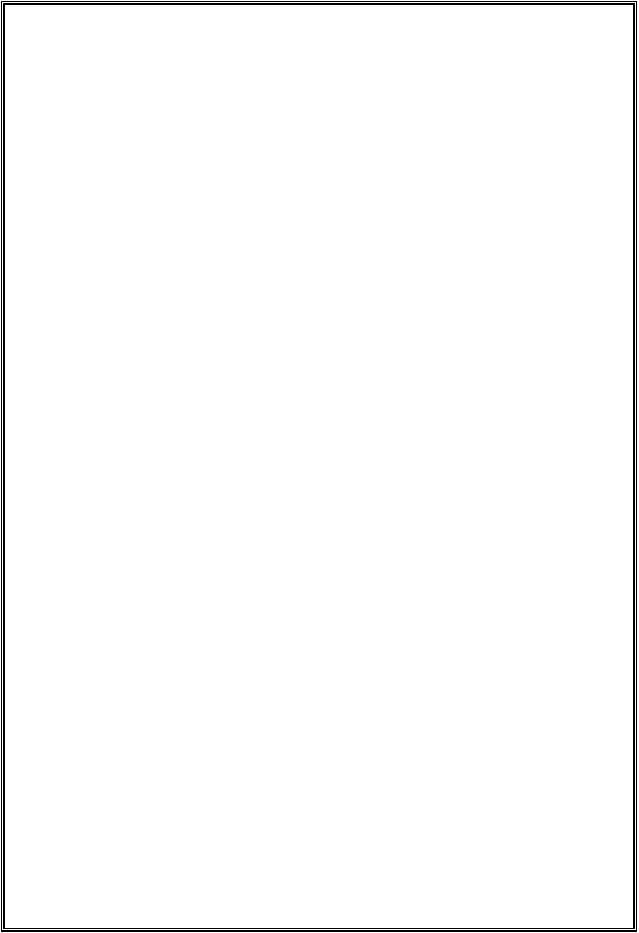
******

**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI**

**BÁO CÁO TỔNG KẾT**

**ĐỀ TÀI NGHIÊN CỨU KHOA HỌC CỦA SINH VIÊN**

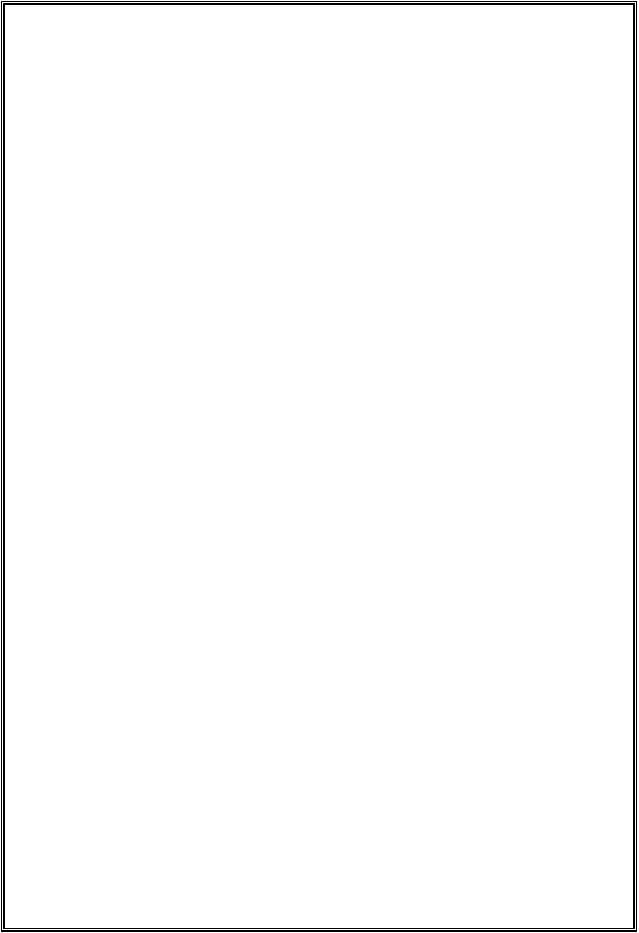
**NĂM 2019**

**<TÊN ĐỀ TÀI>**

**Sinh viên thực hiện**

|  |  |  |
| --- | --- | --- |
| **Nguyễn Trung Nghĩa** | Lớp: CNTT 3 | Khoa: Công nghệ thông tin |
| **Đỗ Văn Cường** | Lớp: CNTT 3 | Khoa: Công nghệ thông tin |
| Phạm Thị Thùy Linh | Lớp: CNTT 3 | Khoa: Công nghệ thông tin |

**Người hướng dẫn:** TS. Trần Văn Dũng

******

**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI**

**BÁO CÁO TỔNG KẾT**

**ĐỀ TÀI NGHIÊN CỨU KHOA HỌC CỦA SINH VIÊN**

**NĂM 2018**

**<TÊN ĐỀ TÀI>**

**Sinh viên thực hiện**

|  |  |  |
| --- | --- | --- |
| **Nguyễn Trung Nghĩa** | Nam, Nữ: Nam | Dân tộc: Kinh |
| Đỗ Văn Cường | Nam, Nữ: Nam | Dân tộc: Kinh |
| Phạm Thị Thùy Linh | Nam, Nữ: Nữ | Dân tộc: Kinh |
| Lớp: CNTT 3 | Khoa: Công nghệ thông tin | Năm thứ: 3 |

Ngành học: Công nghệ thông tin

**Người hướng dẫn:** TS. Trần Văn Dũng

Mục lục

[**1 Mở đầu:** 5](#_Toc533596911)

[1.1 Giới thiệu: 5](#_Toc533596912)

[**2 Kết quả nghiên cứu:** 6](#_Toc533596913)

[2.1 Phương pháp kí tự xấu (Bad character heuristics) : 6](#_Toc533596914)

[2.1.1 Nội dung (Content): 6](#_Toc533596915)

[2.1.2 Xử lý (Processing): 6](#_Toc533596916)

[2.2 Phương pháp hậu tố tốt (Good suffix heuristics) : 7](#_Toc533596917)

[2.2.1 Nội dung (Content): 7](#_Toc533596918)

[2.2.2 Xử lý (Processing): 7](#_Toc533596919)

[2.3 Quá trình xử lý : 8](#_Toc533596920)

[**3 Kết luận và hướng phát triển** 10](#_Toc533596921)

[3.1 Kết luận: 10](#_Toc533596922)

[3.2 Hướng phát triển 10](#_Toc533596923)

[**Danh sách tài liệu tham khảo:** 10](#_Toc533596924)

**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI**

**THÔNG TIN KẾT QUẢ NGHIÊN CỨU CỦA ĐỀ TÀI**

**1. Thông tin chung:**

**-** Tên đề tài:

- Sinh viên thực hiện: Nguyễn Trung Nghĩa, Đỗ Văn Cường, Phạm Thị Thùy Linh

- Lớp: CNTT 3 Khoa: Công nghệ thông tin Năm thứ: 3 Số năm đào tạo: 4

- Người hướng dẫn: TS. Trần Văn Dũng

**2. Mục tiêu đề tài:**

- Nghiên cứu phương pháp tìm kiếm chuỗi kí tự trong một đoạn văn.

- Tìm hiểu phương pháp được áp dụng phổ biến để xử lý bài toán trên là thuật toán Boyer Moore.

**3. Tính mới và sáng tạo:**

- Đề tài không mới song phức tạp trong thực nghiệm và vẫn còn tiềm năng phát triển.

**4. Kết quả nghiên cứu:**

**-** Tìm hiểu, nắm bắt thuật toán.

**-** Tái tạo được ứng dụng với dữ liệu mô phỏng.

**5. Đóng góp về mặt thực tiễn và khả năng áp dụng của đề tài:**

**-** Đề tài có nhiều ứng dụng đối với thực tiễn. Ví dụ, các thuật toán so khớp chuỗi dung để tìm kiếm các khuôn mẫu cụ thể trong các dãy DNA.

**6.** **Công bố khoa học của sinh viên từ kết quả nghiên cứu của đề tài** *(ghi rõ họ tên tác giả, nhan đề và các yếu tố về xuất bản nếu có)* hoặc nhận xét, đánh giá của cơ sở đã áp dụng các kết quả nghiên cứu *(nếu có)*:

Ngày tháng năm

|  |  |
| --- | --- |
|  | **Sinh viên chịu trách nhiệm chính**  **thực hiện đề tài**  *(ký, họ và tên)* |

**Nhận xét của người hướng dẫn về những đóng góp khoa học của sinh viên thực hiện đề tài** *(phần này do người hướng dẫn ghi):*

Ngày tháng năm

|  |  |
| --- | --- |
|  | **Người hướng dẫn**  *(ký, họ và tên)* |

# **1 Mở đầu:**

## 1.1 Giới thiệu:

Hiện nay, cùng với sự phát triển không ngừng của ngành khoa học máy tính chính là việc hệ thống thông tin được lưu trữ ngày càng đồ sộ. Đối với một kho thông tin lớn như vậy, việc người dùng muốn tra cứu, truy vấn dữ liệu cũng ngày càng khó khăn hơn. Bên cạnh đó, khi lượng thông tin phát triển quá nhiều, việc tổ chức, quản lí chúng để làm sao kiểm soát được việc bùng nổ thông tin cũng là một trong những vấn đề cần quan tâm của các nhà quản lí. Đã có rất nhiều công cụ truy vấn có thể hỗ trợ cho người dùng phần nào trong việc tìm kiếm. Tuy nhiên, các công cụ trên vẫn tồn tại những hạn chế nhất định.Trong khi đó, công việc tìm kiếm, truy vấn dữ liệu làm sao để nhanh chóng và hiệu quả vẫn đang là một vấn đề cấp thiết đang được rất nhiều người dùng quan tâm. Các thông tin được lưu trữ trên máy tính tuy lớn nhưng đa số đều được lưu dưới dạng văn bản, và mặc dù có rất nhiều công cụ tìm kiếm nhưng cơ chế chung của chúng vẫn là dựa trên phương pháp sử dụng chuỗi. So khớp chuỗi (pattern matching) là một bài toán quan trọng trong việc hỗ trợ tìm kiếm văn bản được áp dụng để tìm một xâu khớp với mẫu trong văn bản hoặc tìm các văn bản có chứa mẫu.

Trong đó thuật toán **Boyer Moore** là thuật toán tìm kiếm chuỗi rất có hiệu quả trong thực tiễn, các dạng khác nhau của thuật toán này thường được cài đặt trong các chương trình soạn thảo văn bản.

Các đặc điểm chính của nó:

- Thực hiện việc so sánh từ phải sang trái.

- Giai đoạn tìm kiếm có độ phức tạp O(m\*n).

- Độ phức tạp O(n/m) trong trường hợp tốt nhất.

Những vấn đề gặp phải trong thực tiễn: Trong việc phân tích sâu nội dung các gói tin, công đoạn đối sánh chuỗi đóng vai trò quyết định. Ngoài việc đảm bảo tính chính xác trong đối sánh, vấn đề tốc độ xử lý cũng rất quan trọng do số lượng các gói tin cần xử lý thường rất lớn.

Mục tiêu của nghiên cứu là nắm rõ thuật toán và xây dựng một chương trình ứng dụng thuật toán Boyer Moore trong việc tìm kiếm chuỗi kí tự trong một đoạn văn bản cho sẵn.

# **2 Kết quả nghiên cứu:**

Thuật toán Boyer Moore so sánh chuỗi kí tự với đoạn văn theo thứ tự từ phải sang trái. Nếu các đoạn kí tự được so sánh không trùng khớp hoàn toàn với các kí tự trong văn bản thì đoạn kí tự sẽ được dịch chuyển sang bên phải một đoạn đã được tính toán để tiếp tục so sánh.

Ví dụ:

0 1 2 3 4 5 6 7 8 9 ...

a b b a d a b a c b a

b a b a c

b a b a c

Đầu tiên việc so sánh kí tự d – c ở vị trí thứ 4 xảy ra lỗi. Kí tự d không xuất hiện trong chuỗi cần tìm. Do đó chuỗi cần tìm không xuất hiện ở bất kì vị trí nào trong khoảng từ 0 đến 4. Vậy nên chuỗi cần tìm sẽ được dịch đến vị trí thứ 5.

## 2.1 Phương pháp kí tự xấu (Bad character heuristics) :

### 2.1.1 Nội dung (Content):

Phương pháp này có thể được áp dụng khi xuất hiện một ký tự không trùng khớp với đoạn văn bản mà kí tự đấy lại xuất hiện ở một vị trí khác trong chuỗi cần tìm. Khi đó chuỗi cần tìm sẽ được dịch chuyển sao cho kí tự không trùng khớp đó được dóng thẳng hàng với kí tự giống nó trong đoạn văn bản.

Ví dụ:

0 1 2 3 4 5 6 7 8 9 ...

a b b a b a b a c b a

b a b a c

b a b a c

Việc so sánh b – c xảy ra lỗi. Kí tự b lại xuất hiện trong chuỗi cần tìm ở vị trí thứ 0 và thứ 2. Chuỗi cần tìm sẽ được dịch chuyển sang vị trí chứa kí tự b phải nhất .

### 2.1.2 Xử lý (Processing):

**COMPUTE-LAST-OCCURRENCE-FUNCTION(P,m,∑)**

1 **for** mỗi ký tự a ϵ ∑

2 **do** λ[a] = -1

3 **for** j ← 1 **to** m

4 **do** λ [P[j]] ← j

5 **return** λ

## 2.2 Phương pháp hậu tố tốt (Good suffix heuristics) :

### 2.2.1 Nội dung (Content):

Đôi khi việc sử dụng phương pháp kí tự xấu (bad character heuristics) không thành công. Trong ví dụ sau đây, việc so sánh a – b xảy ra lỗi. Việc dóng thẳng hàng kí tự a trong đoạn văn bản với kí tự a phải nhất trong chuỗi cần tìm dẫn đến việc dịch chuyển ngược. Thay vào đó có thể dịch chuyển sang bên phải 1 kí tự. Tuy nhiên trong trường hợp này, tốt hơn là lấy khoảng cách dịch chuyển tối đa có thể từ cấu trúc của chuỗi cần tìm.

Ví dụ:

0 1 2 3 4 5 6 7 8 9 ...

a b a a b a b a c b a

c a b a b

c a b a b

Đuôi ab đã phù hợp. Chuỗi cần tìm có thể di chuyển cho tới khi dóng thẳng hàng với kí tự ab ở trong đoạn văn bản (dịch đến vị trí thứ 2).

Trường hợp sau đây, đuôi ab đã phù hợp. Nhưng không có kí tự ab nào khác ở trong đoạn văn bản. Do đó chuỗi cần tìm sẽ được dịch chuyển ra đằng sau kí tự ab (dịch đến vị trí thứ 5).

Ví dụ:

0 1 2 3 4 5 6 7 8 9 ...

a b c a b a b a c b a

c b a a b

c b a a b

Trong trường hợp sau đây, đuôi bab đã phù hợp. Không có kí tự bab nào khác trong đoạn văn bản. Nhưng trong trường hợp này chuỗi cần tìm không cần phải dịch chuyển đến vị trí thứ 5 mà chỉ dịch đén vị trí thứ 3 bởi vì tiền tố của chuỗi cần tìm (ab) phù hợp với đuôi của kí tự bab.

Ví dụ:

0 1 2 3 4 5 6 7 8 9 ...

a a b a b a b a c b a

a b b a b

a b b a b

### 2.2.2 Xử lý (Processing):

**COMPUTE-GOOD-SUFIX-FUNCTION(P,m)**

1 π ← COMPUTE-PREFIX-FUNCTION(P)

2 P’ ← reverse(P)

3 π’ ← COMPUTE-PREFIX-FUNCTION(P’)

4 **for** j ← 0 **to** m

5 **do** γ[j] ← m - π’[m]

6 **for** l ← 1 **to** m

7 **do** j ← m - π’[l]

8 **if** γ[j] > l - π’[l]

9 **then** γ[j] ← l - π’[l]

10 **return** γ

**COMPUTE-PREFIX-FUNCTION(P)**

1 m ← length[P]

2 π[l] ← 0

3 k ← 0

4 **for** q ← 2 **to** m

5 **do while** k > 0 và P[k+1] ≠ P[q]

6 **do** k ← π[k]

7 **if** P[k+1] = P[q]

8 **then** k ← k+1

9 π[q] ← k

10 **return** π

* **Chuỗi cần tìm sẽ được dịch chuyển một đoạn bằng khoảng cách dài nhất giữa 2 khoảng cách tìm được bởi 2 phương pháp kí tự xấu và hậu tố tốt.**

## 2.3 Quá trình xử lý :

BOYER MOORE MATCHER(T,P,∑)

1 n ← length[T]

2 m ← length[P]

3 λ ← COMPUTE-LAST-OCCURRENCE-FUNCTION(P,m,∑)

4 γ ← COMPUTE-GOOD-SUFIX-FUNCTION(P,m)

5 s ← 0

6 **while** s ≤ n – m

7 **do** j ← m

8 **while** j > 0 và P[j] = T[s+j]

9 **do** j ← j – 1

10 **if** j = 0

11 **then** in “Pattern occurs at shift” s

12 s ← s + γ[0]

13 **else** s ← s + max(γ[j], j – λ[T[s + j]])

# **3 Kết luận và hướng phát triển**

## 3.1 Kết luận:

## 3.2 Hướng phát triển

# **Danh sách tài liệu tham khảo:**

1. Giáo trình thuật toán lý thuyết và bài tập Sơ cấp – Trung cấp – Cao cấp
2. Algorithmen