# Các thuật toán đối sánh chuỗi

Nguyễn Tuấn Anh *

Ngày 23 tháng 3 năm 2022

## 1. Giới thiệu bài toán

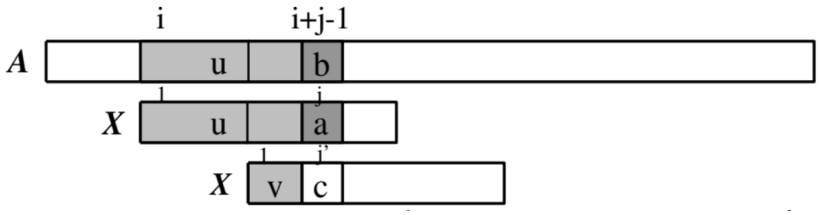
Trước khi đến với bài toán chính, ta cần biết một khái niệm có thể được phát biểu ngắn gọn. *Đối sánh chuỗi* là so sánh các chuỗi với nhau.

**Bài toán.** Cho xâu $A$ có độ dài $m$ và xâu $X$ có độ dài $n$ ($m \geq n$). Tìm xâu $X$ trong $A$. Nếu có, hãy cho biết vị trí xuất hiện đầu tiên của xâu $X$, còn không thì in ra $-1$.

Hai thuật toán đối sánh chuỗi dưới đây giải quyết bài toán trên đều cho kết quả chính xác.

## 2. Thuật toán Knuth-Morris-Pratt (KMP)

Có thể nói, đây là một bản nâng cấp tuyệt vời của BruceForce.
Thuật toán được minh họa chi tiết dưới đây.



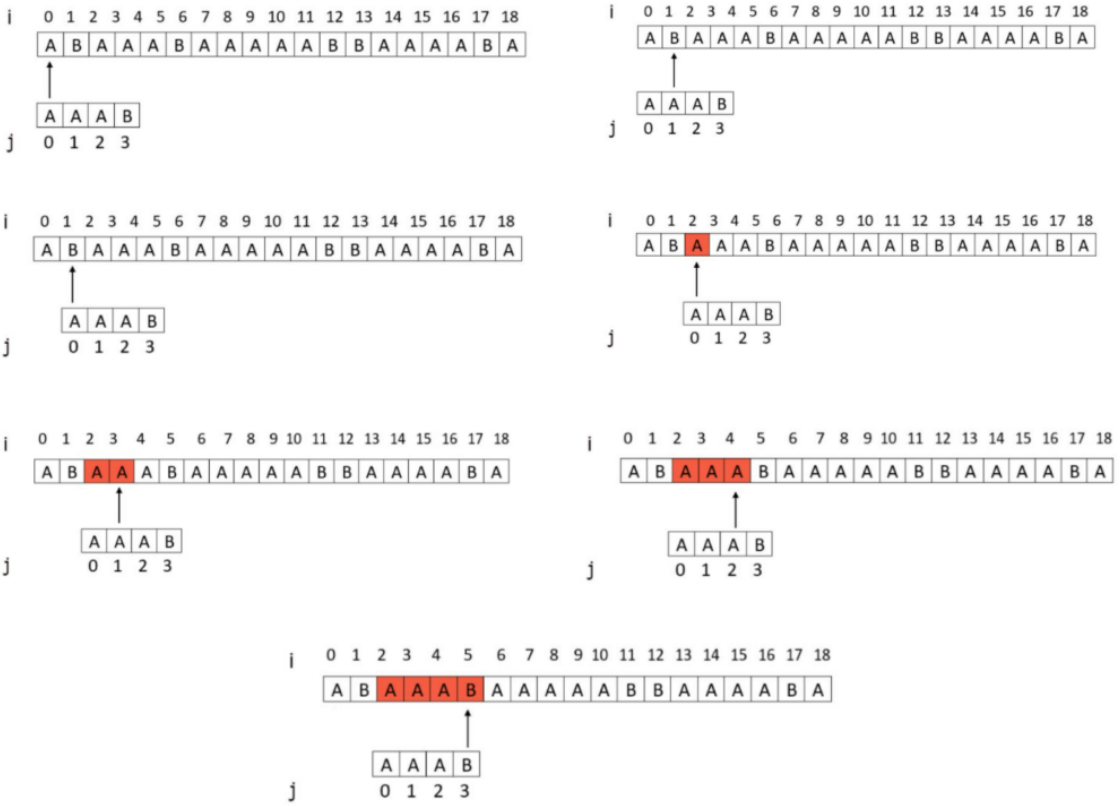

Đầu tiên, cố định chuỗi $A$, di chuyển chuỗi $X$ trượt theo chuỗi $A$.
Giả sử chuỗi $X$ đang ở vị trí $i$ (tức là $X[1] = A[i]$), xét lần lượt $X_1, X_2, ..., X_n$ với các kí tự tương ứng $A_i, A_{i+1}, ..., A_{i+n-1}$.
Sự khác biệt đầu tiên xuất hiện tại vị trí $j$ (nếu không, chúng ta đã có đáp án), có nghĩa $X_j \neq A_{i+j-1}$. Định nghĩa cách viết $A_{i...(i+j-2)}$ là chuỗi có $j-1$ phần tử. Ta sẽ dịch chuyển chuỗi $X$ một đoạn *ngắn nhất* mà:
- Phần tử đầu tiên của chuỗi $X$ trùng với phần tử cuối của chuỗi $A_{i...(i+j-2)}$.
- $X_j \neq X_{j'}$ (sau khi dịch chuyển, vị trí của $A_{i+j-1}$ là $X_{j'}$, hơn nữa $X_{1...(j-1)} = A_{i...(i+j-2)}$).
Nếu xét đến $j$ mà lại có $X_j \neq A_{i+j-1}$ thì chúng ta cần quan tâm đến $j'$ thỏa mãn $X_{1...(j'-1)}$ dài nhất đáp ứng được 2 điều kiện ở trên.
Chúng ta có thể kết hợp kỹ thuật Quy hoạch động khi tìm chuỗi dài nhất.

*Sinh viên K16 lớp KHTN2021, Trường Đại học Công nghệ Thông tin - ĐHQG TP. HCM

Nhận xét – Độ phức tạp $O(m+n)$.

*Chứng minh.* Mỗi lần dịch chuyển chúng ta không cần xét lại toàn bộ dãy. Các phép so sánh được thực hiện tuyến tính. □

Thuật toán nào cũng có ưu điểm và nhược điểm. Đối với KMP thì
**Ưu điểm:** Dễ thấy nhất, thuật toán thực thi khá nhanh (độ phức tạp tuyến tính).
**Nhược điểm:**
- Phụ thuộc kích thước bảng chữ cái.
- Bộ nhớ khá lớn.
Ví dụ cụ thể cho thuật toán:



Hình 1: Minh họa cụ thể thuật toán.

Ở trên, để đếm số lần xuất hiện, chỉ cần thêm 1 biến mà không làm thay đổi độ phức tạp.

## 3. Thuật toán Rabin-Karp

Ý tưởng chủ đạo của phương pháp này là biểu diễn xâu bởi số nguyên và băm. Mỗi một xâu sẽ được gán với một giá trị của hàm băm, hai xâu được gọi là bằng nhau nếu giá trị băm của nó bằng nhau. Như vậy, thay vì phải so sánh các xâu con của $X$ với mẫu $A$, ta chỉ cần so sánh giá trị hàm băm của chúng và đưa ra kết luận.
Chúng ta có thể biểu diễn xâu dựa trên bảng mã ASCII cùng các phép toán cộng, nhân,...

Hàm băm được dùng ở đây là hàm băm rollHash, giá trị băm của chuỗi con tiếp theo được tính dựa theo giá trị băm của chuỗi con trước đó. Hàm băm đơn giản đáp ứng được điều này, dễ nhận thấy nhất, chính là hàm băm dạng đa thức

$$f(X[0,1,...,m-1]) = c_1 b^{m-1} + c_2 b^{m-2} + ... + c_{m-1} b + c_m$$

Với $c_i (i = \overline{1,m})$ là giá trị thứ $i$ sau khi biểu diễn xâu bằng số nguyên, $b$ là cơ số do người lập trình chọn.
Bằng rollHash, hàm băm thứ $k$ có thể chọn theo

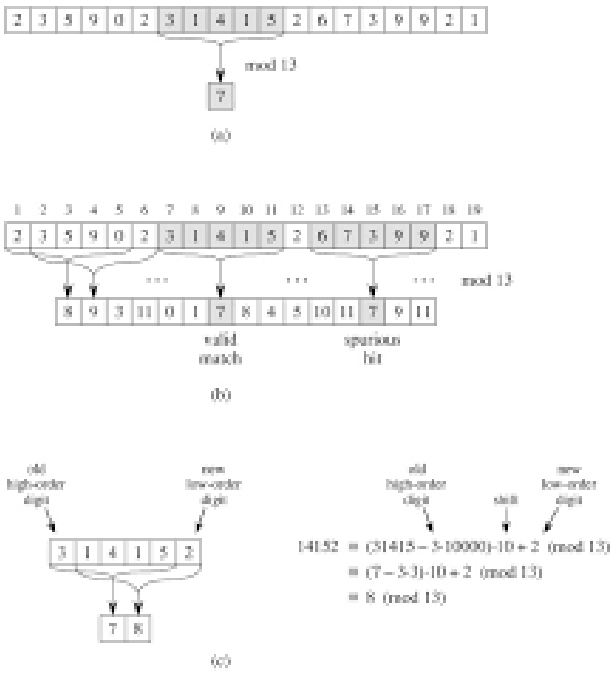
$$f = (f_{k-1} - c_{k-1} b^{m-1}) b + c_n$$

Để tránh tràn số, chúng ta nên chọn hàm $f' = f \bmod p$.

> Nhận xét – Độ phức tạp trung bình $O(m+n)$, trong trường hợp xấu nhất $O(mn)$.

*Chứng minh.* Được đề cập chi tiết tại https://www.giaithuatlaptrinh.com/?p=290 với công cụ Analytic Number Theory. □

Đối với Rabin-Karp
**Ưu điểm:** Với hàm băm đủ tốt, thuật toán hoạt động tốt và dễ thực hiện.
**Nhược điểm:** Khi giá trị băm của mẫu khớp với giá trị băm của một tập trong đoạn văn bản nhưng tập đó không phải là mẫu thực thì đó được gọi là hiện tượng trùng khớp nhầm. Hiện tượng này làm tăng độ phức tạp của thuật toán. Tuy nhiên, nhược điểm này có thể khắc phục bằng cách xét modulo.
Ví dụ minh họa

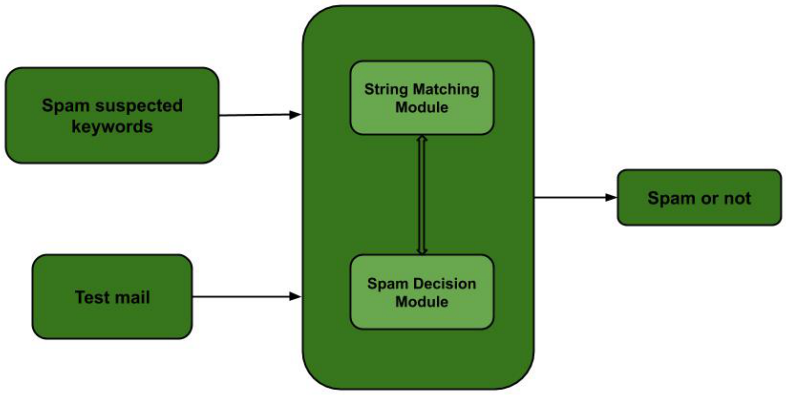


# 4. Ứng dụng

Hai thuật toán trên có nhiều ứng dụng, dưới đây là 3 ứng dụng quan trọng.

## 4.1. Lọc thư rác


Thư rác (spam) là thư điện tử, tin nhắn được gửi đến người nhận mà người nhận đó không mong muốn. Có không ít phương pháp lọc thư rác, trong đó *Lọc theo từ khóa* là một phương pháp truyền thống. Người ta dựa vào những từ hay cụm từ có trong đầu đề của thư và nội dung của thư để lọc. Thuật toán KMP sử dụng kỹ thuật lọc dựa trên phương pháp lọc theo từ khóa.



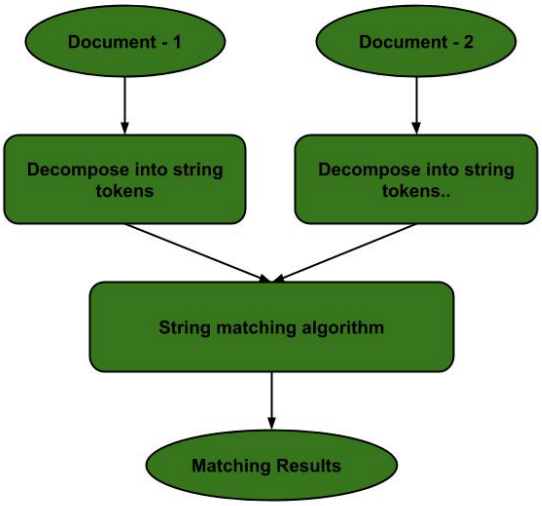

Hình 2: Xử lí spam.

Chẳng hạn, hệ thống có trước một số keyword được định nghĩa spam. Khi tài khoản người dùng nhận được email mới, hệ thống sẽ thực hiện so sánh chuỗi. Nếu chuỗi xuất hiện thì đoạn tin nhắn sẽ bị nghi ngờ hoặc thậm chí bị đánh dấu spam. Vì vậy, người dùng trao đổi thư từ cần chú ý văn phong.

## 4.2. Kiểm tra đạo văn

Kiểm tra đạo văn là xác định tỉ lệ đạo văn trong một tài liệu nào đó. Điều này thật sự cần thiết vì lí do bản quyền hoặc tài sản trí tuệ.

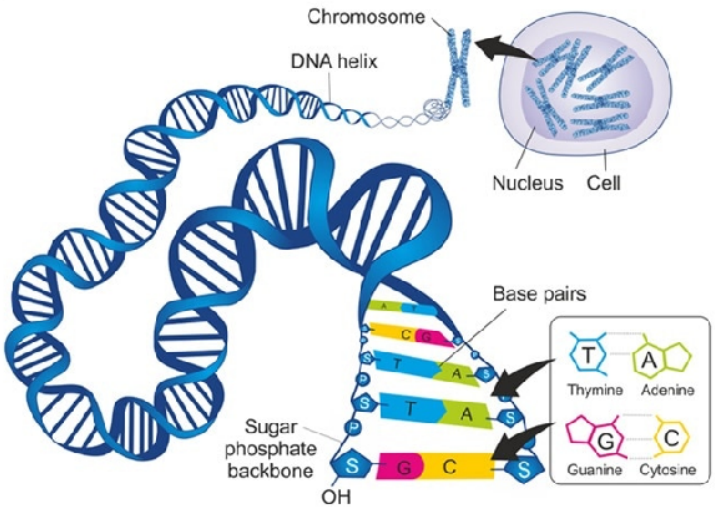
Trong một văn bản, người đạo văn có thể biến tấu rất nhiều, kết hợp nhiều tài liệu gốc, tính rời rạc hóa lúc này rất điển hình. Lí do này khiến Rabin-Karp trở nên khá hiệu quả.




Hình 3: Phát hiện đạo văn.

Rabin-Karp thực hiện đối sánh chuỗi trong hai tài liệu, sau đó tính toán tỉ lệ đạo văn theo một công thức riêng. Tỉ lệ này nếu đạt tới mức nào đó, tài liệu được xem là đạo văn (thường sẽ có quy định về tỉ lệ).
Hiện nay, có một số phần mềm kiểm tra đạo văn như: Small SEO Tool, Plagium, Turnitin,...

### 4.3. Giải trình tự DNA




Chiều dài của tất cả các đoạn DNA chưa được cuộn bên trong cơ thể người lên đến 67 tỷ dặm, được biểu diễn bởi 4 gốc nitơ $(A, T, G, X)$. Trình tự DNA chi phối con người trong hành động, suy nghĩ,..., việc tìm hiểu cũng như giải trình tự DNA được sự quan tâm đặc biệt của các nhà khoa học.
Thuật toán Rabin-Karp cùng kỹ thuật băm so khớp các trình tự nucleotide chính xác với bộ gen người trong bộ hàng tỷ nucleotide. Bằng cách sử dụng băm, chúng ta có thể giảm các trình tự nucleotide lên đến hàng triệu thành chuỗi dài chỉ từ một đến hai chữ số.