

Phân tích Chiến lược Giải thuật Xử lý Chuỗi

Phân khúc Rating 800 - 1000 trên Codeforces

Slide Learning C++

Ngày 15 tháng 1 năm 2026

Tầm quan trọng của Xử lý Chuỗi

- Phân khúc 800-1000 là giai đoạn chuyển giao từ cú pháp sang tư duy giải thuật.
- Chuỗi ký tự là hình thức biểu diễn thông tin tự nhiên và phổ biến nhất.
- Yêu cầu sự kết hợp giữa logic, quản lý bộ nhớ và cấu trúc dữ liệu cơ bản.

Tầm quan trọng của Xử lý Chuỗi

- Phân khúc 800-1000 là giai đoạn chuyển giao từ cú pháp sang tư duy giải thuật.
- Chuỗi ký tự là hình thức biểu diễn thông tin tự nhiên và phổ biến nhất.
- Yêu cầu sự kết hợp giữa logic, quản lý bộ nhớ và cấu trúc dữ liệu cơ bản.

Mục tiêu báo cáo

Cung cấp bộ công cụ tư duy (mental models) để giải quyết các bài toán String thông qua 20 ví dụ điển hình.

Cơ sở Lý luận và Thách thức

- **Biểu diễn bộ nhớ:** `std::string` (C++) linh hoạt nhưng cần tối ưu; String trong Python là bất biến (immutable).
- **Bảng mã ASCII:** Tận dụng tính liên tục của 'a'-'z' để tính toán chỉ số: `c - 'a'`.
- **Độ phức tạp:** Với $N \approx 10^2 - 10^5$, các giải thuật $O(N)$ hoặc $O(N \log N)$ là lý tưởng.

Cạm bẫy

Lạm dụng các hàm thư viện như `find` hay `substr` bên trong vòng lặp có thể dẫn đến $O(N^2)$ và gây lỗi TLE.

1. Way Too Long Words (71A)

Link: <https://codeforces.com/problemset/problem/71/A>

Rating: 800

Đề bài

Nếu từ dài **hơn 10 ký tự**, thay thế bằng: [ký tự đầu] + [số ký tự ở giữa] + [ký tự cuối]. Ví dụ: "localization" → "l10n".

1. Way Too Long Words (71A)

Link: <https://codeforces.com/problemset/problem/71/A>

Rating: 800

Đề bài

Nếu từ dài **hơn 10 ký tự**, thay thế bằng: [ký tự đầu] + [số ký tự ở giữa] + [ký tự cuối]. Ví dụ: "localization" → "l10n".

Phân tích chiến lược

- Kiểm tra điều kiện $L > 10$.
- Số ký tự ở giữa: $L - 2$.
- Lưu ý: Xử lý bộ đệm (buffer) khi chuyển từ `cin` » `n` sang `getline` hoặc `cin` » `s`.

2. Word (59A)

Link: <https://codeforces.com/problemset/problem/59/A>

Rating: 800

Đề bài

Chuẩn hóa từ về toàn hoa hoặc toàn thường. Nếu số chữ hoa $>$ số chữ thường: chuyển toàn bộ thành hoa. Ngược lại: chuyển thành thường.

2. Word (59A)

Link: <https://codeforces.com/problemset/problem/59/A>

Rating: 800

Đề bài

Chuẩn hóa từ về toàn hoa hoặc toàn thường. Nếu số chữ hoa $>$ số chữ thường: chuyển toàn bộ thành hoa. Ngược lại: chuyển thành thường.

Phân tích chiến lược

- Duyệt 1: Đếm `upper_count` và `lower_count`.
- Duyệt 2: Biến đổi dựa trên so sánh.
- Kỹ thuật: Dùng `tolower()`/`toupper()` hoặc cộng trừ 32 trong mã ASCII.

3. Petya and Strings (112A)

Link: <https://codeforces.com/problemset/problem/112/A>

Rating: 800

Đề bài

So sánh thứ tự từ điển của hai chuỗi không phân biệt hoa thường. Trả về -1, 1, hoặc 0.

3. Petya and Strings (112A)

Link: <https://codeforces.com/problemset/problem/112/A>

Rating: 800

Đề bài

So sánh thứ tự từ điển của hai chuỗi không phân biệt hoa thường. Trả về -1, 1, hoặc 0.

Phân tích chiến lược

- **Chuẩn hóa:** Chuyển cả hai chuỗi về cùng một dạng (thường là lowercase).
- Sử dụng toán tử so sánh có sẵn (<, >, ==) trên `std::string`.

4. Word Capitalization (281A)

Link: <https://codeforces.com/problemset/problem/281/A>

Rating: 800

Đề bài

Viết hoa chữ cái đầu tiên của từ, các chữ cái khác giữ nguyên.

4. Word Capitalization (281A)

Link: <https://codeforces.com/problemset/problem/281/A>

Rating: 800

Đề bài

Viết hoa chữ cái đầu tiên của từ, các chữ cái khác giữ nguyên.

Phân tích chiến lược

- Chỉ thao tác tại chỉ số `s[0]`.
- Sử dụng `toupper(s[0])`. Không cần kiểm tra nếu đã là chữ hoa vì hàm thư viện đã xử lý giúp.

5. Boy or Girl (236A)

Link: <https://codeforces.com/problemset/problem/236/A>

Rating: 800

Đề bài

Đếm số ký tự riêng biệt (distinct characters). Nếu chẵn là nữ (CHAT WITH HER!), nếu lẻ là nam (IGNORE HIM!).

5. Boy or Girl (236A)

Link: <https://codeforces.com/problemset/problem/236/A>

Rating: 800

Đề bài

Đếm số ký tự riêng biệt (distinct characters). Nếu chẵn là nữ (CHAT WITH HER!), nếu lẻ là nam (IGNORE HIM!).

Phân tích chiến lược

- **Cách 1:** Dùng `std::set<char>` để lưu và lấy `size()`.
- **Cách 2:** Dùng mảng tần suất `bool seen[26]`.
- **Cách 3:** Sắp xếp chuỗi rồi đếm các cặp `s[i] != s[i-1]`.

6. Translation (41A)

Link: <https://codeforces.com/problemset/problem/41/A>

Rating: 800

Đề bài

Kiểm tra xem chuỗi t có phải là phiên bản đảo ngược của chuỗi s hay không.

6. Translation (41A)

Link: <https://codeforces.com/problemset/problem/41/A>

Rating: 800

Đề bài

Kiểm tra xem chuỗi t có phải là phiên bản đảo ngược của chuỗi s hay không.

Phân tích chiến lược

- **Cách 1:** Dùng `std::reverse(s.begin(), s.end())` rồi so sánh $s == t$.
- **Cách 2:** So sánh $s[i]$ với $t[n-1-i]$.
- **Lưu ý:** Kiểm tra độ dài trước khi so sánh chi tiết.

7. Anton and Danik (734A)

Link: <https://codeforces.com/problemset/problem/734/A>

Rating: 800

Đề bài

Đếm số lần 'A' (Anton) và 'D' (Danik) xuất hiện. Ai nhiều hơn thì thắng.

7. Anton and Danik (734A)

Link: <https://codeforces.com/problemset/problem/734/A>

Rating: 800

Đề bài

Đếm số lần 'A' (Anton) và 'D' (Danik) xuất hiện. Ai nhiều hơn thì thắng.

Phân tích chiến lược

- Duyệt tuyến tính (Linear Scan).
- Sử dụng hai biến đếm đơn giản.

8. Amusing Joke (141A)

Link: <https://codeforces.com/problemset/problem/141/A>

Rating: 800

Đề bài

Cho 3 chuỗi: A , B , C . Kiểm tra xem các ký tự của C có tạo thành đúng tập hợp ký tự của $(A + B)$ hay không.

8. Amusing Joke (141A)

Link: <https://codeforces.com/problemset/problem/141/A>

Rating: 800

Đề bài

Cho 3 chuỗi: A , B , C . Kiểm tra xem các ký tự của C có tạo thành đúng tập hợp ký tự của $(A + B)$ hay không.

Phân tích chiến lược

- Nối chuỗi: $D = A + B$.
- Sắp xếp cả C và D . So sánh $C == D$.
- Hoặc sử dụng mảng tần suất (Frequency Map) để so sánh.

9. Pangram (520A)

Link: <https://codeforces.com/problemset/problem/520/A>

Rating: 800

Đề bài

Kiểm tra xem chuỗi có chứa đủ 26 chữ cái Latin (không phân biệt hoa thường) hay không.

9. Pangram (520A)

Link: <https://codeforces.com/problemset/problem/520/A>

Rating: 800

Đề bài

Kiểm tra xem chuỗi có chứa đủ 26 chữ cái Latin (không phân biệt hoa thường) hay không.

Phân tích chiến lược

- Chuyển về lowercase.
- Đưa vào `std::set<char>`.
- Kết quả là "YES" nếu `set.size() == 26`.

10. Stones on the Table (266A)

Link: <https://codeforces.com/problemset/problem/266/A>

Rating: 800

Đề bài

Tìm số đá tối thiểu cần bỏ để không có hai viên đá nào cạnh nhau cùng màu.

10. Stones on the Table (266A)

Link: <https://codeforces.com/problemset/problem/266/A>

Rating: 800

Đề bài

Tìm số đá tối thiểu cần bỏ để không có hai viên đá nào cạnh nhau cùng màu.

Phân tích chiến lược

- **Tham lam (Greedy):** So sánh các phần tử liền kề.
- Nếu $s[i] == s[i+1]$, tăng biến đếm (loại bỏ 1 viên).

11. Football (96A)

Link: <https://codeforces.com/problemset/problem/96/A>

Rating: 900

Đề bài

Kiểm tra xem có ít nhất 7 cầu thủ cùng đội (ký tự giống nhau) đứng liên tiếp hay không.

11. Football (96A)

Link: <https://codeforces.com/problemset/problem/96/A>

Rating: 900

Đề bài

Kiểm tra xem có ít nhất 7 cầu thủ cùng đội (ký tự giống nhau) đứng liên tiếp hay không.

Phân tích chiến lược

- Duyệt và đếm số ký tự trùng lặp liên tiếp. Reset về 1 nếu gặp ký tự khác.
- Hoặc dùng hàm `s.find("0000000")` và `s.find("1111111")`.

12. Dubstep (208A)

Link: <https://codeforces.com/problemset/problem/208/A>

Rating: 900

Đề bài

Khôi phục bài hát gốc bằng cách loại bỏ "WUB". Các từ gốc cách nhau bằng 1 dấu cách.

12. Dubstep (208A)

Link: <https://codeforces.com/problemset/problem/208/A>

Rating: 900

Đề bài

Khôi phục bài hát gốc bằng cách loại bỏ "WUB". Các từ gốc cách nhau bằng 1 dấu cách.

Phân tích chiến lược

- Duyệt chuỗi, nếu gặp "WUB" thì bỏ qua 3 ký tự.
- Dùng biến cờ hiệu (`flag`) để xử lý khoảng trắng giữa các từ, tránh khoảng trắng thừa ở đầu/cuối.

13. HQ9+ (133A)

Link: <https://codeforces.com/problemset/problem/133/A>

Rating: 900

Đề bài

Kiểm tra xem chương trình HQ9+ có in ra gì không (lệnh H, Q, 9 có in, lệnh + không in).

13. HQ9+ (133A)

Link: <https://codeforces.com/problemset/problem/133/A>

Rating: 900

Đề bài

Kiểm tra xem chương trình HQ9+ có in ra gì không (lệnh H, Q, 9 có in, lệnh + không in).

Cạm bẫy

Đừng cố mô phỏng lệnh '+'. Nó chỉ thay đổi biến nội bộ, không tạo ra output.

Phân tích chiến lược

Tìm sự tồn tại của ký tự 'H', 'Q', hoặc '9' trong chuỗi đầu vào.

14. String Task (118A)

Link: <https://codeforces.com/problemset/problem/118/A>

Rating: 1000

Đề bài

Xóa nguyên âm (bao gồm cả 'y'), viết thường các phụ âm và thêm dấu '.' trước mỗi phụ âm.

14. String Task (118A)

Link: <https://codeforces.com/problemset/problem/118/A>

Rating: 1000

Đề bài

Xóa nguyên âm (bao gồm cả 'y'), viết thường các phụ âm và thêm dấu '.' trước mỗi phụ âm.

Phân tích chiến lược

- Chuyển về lowercase trước.
- Duyệt từng ký tự, kiểm tra nếu không phải nguyên âm thì in ra "." + c.

15. Chat Room (58A)

Link: <https://codeforces.com/problemset/problem/58/A>

Rating: 1000

Đề bài

Kiểm tra xem "hello" có phải là một dãy con (subsequence) của chuỗi đầu vào hay không.

15. Chat Room (58A)

Link: <https://codeforces.com/problemset/problem/58/A>

Rating: 1000

Đề bài

Kiểm tra xem "hello" có phải là một dãy con (subsequence) của chuỗi đầu vào hay không.

Phân tích chiến lược

- Dùng một biến chỉ mục $idx = 0$ cho chuỗi mục tiêu "hello".
- Duyệt chuỗi input, nếu $s[i] == target[idx]$ thì tăng idx .
- Nếu $idx == 5$ thì kết luận YES.

16. cAPS IOCK (131A)

Link: <https://codeforces.com/problemset/problem/131/A>

Rating: 1000

Đề bài

Sửa lỗi Caps Lock: Nếu từ toàn in hoa HOẶC chỉ chữ cái đầu viết thường còn lại in hoa → Đảo ngược toàn bộ trạng thái hoa/thường.

16. cAPS LOCK (131A)

Link: <https://codeforces.com/problemset/problem/131/A>

Rating: 1000

Đề bài

Sửa lỗi Caps Lock: Nếu từ toàn in hoa HOẶC chỉ chữ cái đầu viết thường còn lại in hoa → Đảo ngược toàn bộ trạng thái hoa/thường.

Phân tích chiến lược

- Kiểm tra điều kiện: Mọi ký tự từ vị trí thứ 2 trở đi ($s[1 \dots n-1]$) phải là in hoa.
- Nếu thỏa mãn, duyệt lại và dùng `islower/isupper` để đảo ngược.

17. Case of the Zeros and Ones (556A)

Link: <https://codeforces.com/problemset/problem/556/A>

Rating: 1000

Đề bài

Xóa cặp "01" hoặc "10" liên tiếp. Tìm độ dài nhỏ nhất còn lại.

17. Case of the Zeros and Ones (556A)

Link: <https://codeforces.com/problemset/problem/556/A>

Rating: 1000

Đề bài

Xóa cặp "01" hoặc "10" liên tiếp. Tìm độ dài nhỏ nhất còn lại.

Quan sát Toán học

Mỗi lần xóa luôn mất một số 0 và một số 1. Thứ tự không quan trọng.

- Gọi n_0 là số lượng '0', n_1 là số lượng '1'.
- Số lượng cặp bị xóa: $\min(n_0, n_1)$.
- Độ dài còn lại: $|n_0 - n_1|$.

18. Keyboard (474A)

Link: <https://codeforces.com/problemset/problem/474/A>

Rating: 900

Đề bài

Khôi phục ký tự gốc khi tay bị lệch sang trái ('L') hoặc phải ('R') trên bàn phím.

18. Keyboard (474A)

Link: <https://codeforces.com/problemset/problem/474/A>

Rating: 900

Đề bài

Khôi phục ký tự gốc khi tay bị lệch sang trái ('L') hoặc phải ('R') trên bàn phím.

Phân tích chiến lược

- Lưu bàn phím vào 1 chuỗi: "qwertyuiopasdfghjkl;zxcvbnm,./"
- Tìm vị trí ký tự đã gõ, rồi lấy ký tự ở vị trí $\text{index}-1$ (nếu lệch 'R') hoặc $\text{index}+1$ (nếu lệch 'L').

19. Double-ended Strings (1506C)

Link: <https://codeforces.com/problemset/problem/1506C>

Rating: 1000

Đề bài

Tìm số lần xóa ký tự ở hai đầu ít nhất để hai chuỗi bằng nhau.

19. Double-ended Strings (1506C)

Link: <https://codeforces.com/problemset/problem/1506C>

Rating: 1000

Đề bài

Tìm số lần xóa ký tự ở hai đầu ít nhất để hai chuỗi bằng nhau.

Phân tích chiến lược

- Thực chất là tìm **Chuỗi con chung dài nhất** (Longest Common Substring - LCS).
- Với $N \leq 20$, dùng Brute Force duyệt mọi chuỗi con của A và kiểm tra trong B bằng `find()`.
- Đáp án: $\text{len}(A) + \text{len}(B) - 2 \times \text{len}(LCS)$.

20. Comparison String (1837B)

Link: <https://codeforces.com/problemset/problem/1837/B>

Rating: 900

Đề bài

Xây dựng dãy số thỏa mãn chuỗi dấu so sánh sao cho số lượng giá trị phân biệt là ít nhất.

20. Comparison String (1837B)

Link: <https://codeforces.com/problemset/problem/1837/B>

Rating: 900

Đề bài

Xây dựng dãy số thỏa mãn chuỗi dấu so sánh sao cho số lượng giá trị phân biệt là ít nhất.

Phân tích chiến lược

- Quan sát: Ta chỉ cần thêm số mới khi dấu so sánh kéo dài liên tục theo một hướng ($<$ hoặc $>$).
- Đáp án = (Độ dài đoạn ký tự giống nhau liên tiếp dài nhất) + 1.

Tổng hợp Kỹ thuật

Bài toán	Rating	Kỹ thuật	Cạm bẫy
71A	800	Parsing	Định dạng số
112A	800	Chuẩn hóa	So sánh trực tiếp
236A	800	Set/Freq Map	Đếm sai ký tự trùng
96A	900	Scan/Find	Reset biến đếm sai
58A	1000	Greedy Subsequence	Nhầm với Substring
556A	1000	Math Observation	Mô phỏng gây TLE

- **Thành thạo STL:** `std::set`, `sort`, `reverse`, `find`.
- **Chuẩn hóa sớm:** Đưa về lowercase ngay sau khi đọc input.
- **Phân biệt rõ:** *Substring* (liên tiếp) và *Subsequence* (không nhất thiết liên tiếp).
- **Tư duy tối ưu:** Luôn tìm quy luật toán học trước khi thực hiện mô phỏng tốn kém.

Chúc các bạn thành công trên con đường CP!