

Phân tích Giải thuật String/Vector trên Codeforces

Elo 800-1000: Nền tảng cho Newbie/Pupil

Báo cáo Nghiên cứu Chuyên sâu

Ngày 14 tháng 1 năm 2026

Tổng quan Điều hành

Mục tiêu

Trang bị kiến thức vững chắc về `std::string` và `std::vector` trong C++ thông qua 20 bài tập chọn lọc.

Trọng tâm kỹ thuật

- Không yêu cầu thuật toán phức tạp (DP, Graph).
- Tập trung vào chuyển đổi ý tưởng thành mã nguồn sạch.
- Kỹ năng: Truy cập chỉ số, quản lý bộ nhớ, tư duy tham lam (Greedy) và mô phỏng (Simulation).

1. Way Too Long Words (71A)

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

Tóm tắt đề bài

Các từ có độ dài **thực sự lớn hơn 10** ký tự cần được viết tắt. Quy tắc: Ký tự đầu + số lượng ký tự ở giữa + ký tự cuối (ví dụ: "localization" → "l10n"). Từ ngắn giữ nguyên.

1. Way Too Long Words (71A)

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

Tóm tắt đề bài

Các từ có độ dài **thực sự lớn hơn 10** ký tự cần được viết tắt. Quy tắc: Ký tự đầu + số lượng ký tự ở giữa + ký tự cuối (ví dụ: "localization" → "l10n"). Từ ngắn giữ nguyên.

Phân tích Chiến lược

- **Chủ đề:** Truy cập ngẫu nhiên chuỗi, điều kiện rẽ nhánh.
- **Logic:** Ký tự đầu là $s[0]$, ký tự cuối là $s[n-1]$. Số ở giữa là $\text{length}() - 2$.
- **Lưu ý quan trọng:** Điều kiện đề bài là “strictly more than 10”. Dùng toán tử $>$ thay vì \geq .
- Độ phức tạp: $O(1)$ cho mỗi từ.

2. Word (59A)

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

Tóm tắt đề bài

Chuẩn hóa từ về toàn bộ chữ hoa hoặc toàn bộ chữ thường dựa trên số lượng hiện có.

- Số chữ hoa > Số chữ thường → Viết hoa toàn bộ.
- Ngược lại → Viết thường toàn bộ.

2. Word (59A)

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

Tóm tắt đề bài

Chuẩn hóa từ về toàn bộ chữ hoa hoặc toàn bộ chữ thường dựa trên số lượng hiện có.

- Số chữ hoa > Số chữ thường → Viết hoa toàn bộ.
- Ngược lại → Viết thường toàn bộ.

Phân tích Chiến lược

- **Kỹ thuật:** Duyệt mảng (Iteration) và thao tác ASCII.
- **Thư viện:** Sử dụng `<cctype>` với `isupper()`, `tolower()`, `toupper()`.
- **Quy trình:** 1. Đếm số lượng hoa/thường (Pass 1). 2. Quyết định và biến đổi chuỗi (Pass 2).

3. Word Capitalization (281A)

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

Tóm tắt đề bài

Viết hoa ký tự đầu tiên của một từ. Nếu đã viết hoa thì giữ nguyên.

3. Word Capitalization (281A)

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

Tóm tắt đề bài

Viết hoa ký tự đầu tiên của một từ. Nếu đã viết hoa thì giữ nguyên.

Phân tích Chiến lược

- **Tính chất:** Chuỗi trong C++ có tính khả biến (mutable). Ta có thể sửa trực tiếp `s[0]`.
- **Tối ưu:** Hàm `toupper(c)` có tính lũy đẳng (idempotent).
- **Hành động:** Gọi `s[0] = toupper(s[0])` vô điều kiện, không cần kiểm tra `if`, giúp code gọn và tránh branch prediction sai.

4. Translation (41A)

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

Tóm tắt đề bài

Kiểm tra xem chuỗi t có phải là dạng viết ngược của chuỗi s hay không.

4. Translation (41A)

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

Tóm tắt đề bài

Kiểm tra xem chuỗi t có phải là dạng viết ngược của chuỗi s hay không.

Phân tích Chiến lược

- **Cách 1 (Constructive):** Đảo ngược s bằng `std::reverse` rồi so sánh $s == t$.
- **Cách 2 (Two Pointers - Tối ưu):** So sánh $s[i]$ với $t[len - 1 - i]$.
- **Lưu ý:** Kiểm tra độ dài trước. Nếu khác độ dài \rightarrow NO ngay lập tức.

5. Petya and Strings (112A)

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

Tóm tắt đề bài

So sánh hai chuỗi theo thứ tự từ điển (lexicographical), không phân biệt hoa thường. In ra -1, 1 hoặc 0.

5. Petya and Strings (112A)

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

Tóm tắt đề bài

So sánh hai chuỗi theo thứ tự từ điển (lexicographical), không phân biệt hoa thường. In ra -1, 1 hoặc 0.

Phân tích Chiến lược

- **Vấn đề ASCII:** 'Z' (90) nhỏ hơn 'a' (97). Cần chuẩn hóa về cùng một kiểu (thường là lowercase) trước khi so sánh.
- **Giải pháp:** 1. Duyệt và chuyển cả 2 chuỗi về chữ thường. 2. Dùng toán tử so sánh chuỗi hoặc hàm `strcmp`.
- Đây là bài tập kinh điển về Data Normalization.

6. Amusing Joke (141A)

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

Tóm tắt đề bài

Cho 3 chuỗi: Khách, Chủ nhà, và một chuỗi hỗn loạn. Kiểm tra xem chuỗi hỗn loạn có phải là tổ hợp chính xác của tên Khách và Chủ nhà không.

6. Amusing Joke (141A)

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

Tóm tắt đề bài

Cho 3 chuỗi: Khách, Chủ nhà, và một chuỗi hỗn loạn. Kiểm tra xem chuỗi hỗn loạn có phải là tổ hợp chính xác của tên Khách và Chủ nhà không.

Phân tích Chiến lược

- **Dạng bài:** Anagram / Bảo toàn số lượng ký tự.
- **Cách 1 (Sorting):** Gộp tên Khách + Chủ → Sort. Sort chuỗi hỗn loạn. So sánh 2 chuỗi đã sort. $O(N \log N)$.
- **Cách 2 (Frequency Map):** Đếm tần suất ký tự (mảng đếm hoặc map). Đảm bảo số lượng từng ký tự 'A'-'Z' khớp nhau hoàn toàn. $O(N)$.

7. Pangram (520A)

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

Tóm tắt đề bài

Kiểm tra xem chuỗi có chứa đủ tất cả các chữ cái từ A đến Z (không phân biệt hoa thường) hay không.

7. Pangram (520A)

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

Tóm tắt đề bài

Kiểm tra xem chuỗi có chứa đủ tất cả các chữ cái từ A đến Z (không phân biệt hoa thường) hay không.

Phân tích Chiến lược

- **Chủ đề:** Bao phủ tập hợp (Set coverage).
- **Cách dùng Set:** Chèn các ký tự (đã tolower) vào std::set. Kiểm tra size() == 26.
- **Cách dùng Mảng đánh dấu:** bool visited[26]. Duyệt chuỗi và đánh dấu. Kiểm tra toàn bộ mảng phải là true.
- **Early Exit:** Nếu độ dài chuỗi < 26, chắc chắn in "NO".

8. Boy or Girl (236A)

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

Tóm tắt đề bài

Đếm số lượng ký tự **phân biệt** trong tên. Số lẻ → Nam (IGNORE HIM), số chẵn → Nữ (CHAT WITH HER).

8. Boy or Girl (236A)

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

Tóm tắt đề bài

Đếm số lượng ký tự **phân biệt** trong tên. Số lẻ → Nam (IGNORE HIM), số chẵn → Nữ (CHAT WITH HER).

Phân tích Chiến lược

- **Tương tự Pangram:** Cần đếm Cardinality của Set các ký tự.
- **Phương pháp:** 1. Dùng `std::set` hoặc `std::sort + std::unique`. 2. Lấy kích thước tập hợp chia lấy dư cho 2.
- Lưu ý: Đề bài yêu cầu đếm ký tự phân biệt, không phải độ dài chuỗi gốc.

9. Stones on the Table (266A)

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

Tóm tắt đề bài

Đếm số lượng tối thiểu các viên đá cần loại bỏ để không có hai viên đá nào cạnh nhau cùng màu (R, G, B).

9. Stones on the Table (266A)

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

Tóm tắt đề bài

Đếm số lượng tối thiểu các viên đá cần loại bỏ để không có hai viên đá nào cạnh nhau cùng màu (R, G, B).

Phân tích Chiến lược

- **Dạng bài:** Tham lam (Greedy) trên cấu trúc tuyến tính.
- **Mục tiêu:** Phá vỡ mọi cặp $s[i] == s[i+1]$.
- **Thực hiện:** Duyệt từ đầu đến sát cuối. Nếu $s[i] == s[i+1]$ thì tăng biến đếm.
- Độ phức tạp: $O(N)$ thời gian, $O(1)$ không gian.

10. Anton and Danik (734A)

<https://codeforces.com/contest/734/problem/A>

Tóm tắt đề bài

Đếm số trận thắng của Anton ('A') và Danik ('D'). So sánh xem ai thắng nhiều hơn hoặc hòa.

10. Anton and Danik (734A)

<https://codeforces.com/contest/734/problem/A>

Tóm tắt đề bài

Đếm số trận thắng của Anton ('A') và Danik ('D'). So sánh xem ai thắng nhiều hơn hoặc hòa.

Phân tích Chiến lược

- **Chủ đề:** Tổng hợp dữ liệu (Aggregation).
- **Giải pháp:** Dùng 2 biến đếm tích lũy. Duyệt chuỗi 1 lần.
- Sử dụng cấu trúc điều khiển if - else if - else để xử lý 3 trường hợp kết quả (Anton, Danik, Friendship).

11. Queue at the School (266B)

<https://codeforces.com/problemset/problem/266/B>

Tóm tắt đề bài

Mô phỏng hàng đợi: Nếu bé trai ('B') đứng trước bé gái ('G'), họ đổi chỗ. Việc này xảy ra đồng thời mỗi giây. Tìm trạng thái sau t giây.

11. Queue at the School (266B)

<https://codeforces.com/problemset/problem/266/B>

Tóm tắt đề bài

Mô phỏng hàng đợi: Nếu bé trai ('B') đứng trước bé gái ('G'), họ đổi chỗ. Việc này xảy ra đồng thời mỗi giây. Tìm trạng thái sau t giây.

Phân tích Chiến lược

- **Chủ đề:** Mô phỏng (Simulation).
- **Cạm bẫy:** Hiệu ứng "dồn toa" nếu xử lý không khéo trong 1 vòng lặp.
- **Giải pháp:** 1. Vòng lặp ngoài đếm thời gian t . 2. Vòng lặp trong duyệt hàng đợi. Nếu gặp "BG" → đổi thành "GB" và **tăng chỉ số thêm 1** để nhảy qua vị trí vừa đổi (tránh đổi tiếp người đó trong cùng 1 giây).

12. Chat Room (58A)

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

Tóm tắt đề bài

Kiểm tra xem có thể xóa một số ký tự trong chuỗi đã cho để thu được từ "hello" hay không (Subsequence).

12. Chat Room (58A)

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

Tóm tắt đề bài

Kiểm tra xem có thể xóa một số ký tự trong chuỗi đã cho để thu được từ "hello" hay không (Subsequence).

Phân tích Chiến lược

- **Khái niệm:** Subsequence Matching (khác với Substring).
- **Mô hình:** Máy trạng thái (State Machine) hoặc con trỏ tham lam.
- **Thực hiện:** Dùng biến trỏ vào "hello". Duyệt chuỗi đầu vào, nếu khớp ký tự đang cần thì trỏ sang ký tự tiếp theo của "hello".
- Nếu trỏ hết "hello" → YES.

13. Dubstep (208A)

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

Tóm tắt đề bài

Khôi phục bài hát gốc bằng cách loại bỏ các từ "WUB" chèn vào giữa/đầu/cuối. Các từ gốc cách nhau 1 dấu cách.

13. Dubstep (208A)

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

Tóm tắt đề bài

Khôi phục bài hát gốc bằng cách loại bỏ các từ "WUB" chèn vào giữa/đầu/cuối. Các từ gốc cách nhau 1 dấu cách.

Phân tích Chiến lược

- **Chủ đề:** Tokenization / Parsing.
- **Xử lý:** "WUB" là dấu phân cách (delimiter).
- **Logic:** 1. Nếu gặp "WUB" → bỏ qua, bật cờ needs_space. 2. Nếu gặp từ thường → in dấu cách (nếu cờ đang bật và không phải đầu câu), in từ, tắt cờ.

14. String Task (118A)

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

Tóm tắt đề bài

Xóa nguyên âm (A, O, Y, E, U, I), chèn dấu "." trước phụ âm, chuyển tất cả về chữ thường.

14. String Task (118A)

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

Tóm tắt đề bài

Xóa nguyên âm (A, O, Y, E, U, I), chèn dấu "." trước phụ âm, chuyển tất cả về chữ thường.

Phân tích Chiến lược

- **Kỹ thuật:** Lọc (Filtering) và Biến đổi (Transformation).
- **Lưu ý:** Chữ 'Y' được tính là nguyên âm trong bài này.
- **Tối ưu:** Nên tạo chuỗi kết quả mới (StringBuilder pattern) thay vì xóa/chèn trực tiếp trên chuỗi gốc (tránh $O(N^2)$).

15. Twins (160A)

<https://codeforces.com/problemset/problem/160/A>

Tóm tắt đề bài

Lấy số lượng đồng xu ít nhất sao cho tổng giá trị lớn hơn tổng giá trị các đồng xu còn lại.

15. Twins (160A)

<https://codeforces.com/problemset/problem/160/A>

Tóm tắt đề bài

Lấy số lượng đồng xu ít nhất sao cho tổng giá trị lớn hơn tổng giá trị các đồng xu còn lại.

Phân tích Chiến lược

- **Chủ đề:** Thuật toán Tham lam (Greedy) kết hợp Sắp xếp.
- **Chiến lược:** Luôn lấy đồng xu mệnh giá cao nhất.
- **Các bước:** 1. Tính tổng toàn bộ. 2. Sắp xếp vector giảm dần. 3. Lấy lần lượt đến khi `current_sum > total_sum / 2`.

16. Arrival of the General (144A)

<https://codeforces.com/problemset/problem/144/A>

Tóm tắt đề bài

Tính số lần đổi chỗ tối thiểu (chỉ đổi 2 người cạnh nhau) để người cao nhất về đầu hàng, người thấp nhất về cuối hàng.

16. Arrival of the General (144A)

<https://codeforces.com/problemset/problem/144/A>

Tóm tắt đề bài

Tính số lần đổi chỗ tối thiểu (chỉ đổi 2 người cạnh nhau) để người cao nhất về đầu hàng, người thấp nhất về cuối hàng.

Phân tích Chiến lược

- **Chủ đề:** Tìm cực trị (Min/Max) và khoảng cách chỉ số.
- **Chọn chỉ số:** Max (ưu tiên chỉ số nhỏ nhất), Min (ưu tiên chỉ số lớn nhất).
- **Công thức:** $Cost = Index_{Max} + (N - 1 - Index_{Min})$.
- **Điều chỉnh:** Nếu $Index_{Max} > Index_{Min}$, trừ đi 1 (do hoán đổi chéo nhau).

17. Helpful Maths (339A)

<https://codeforces.com/problemset/problem/339/A>

Tóm tắt đề bài

Sắp xếp lại các số hạng trong phép cộng (ví dụ " $3+2+1 \rightarrow 1+2+3$ ").

17. Helpful Maths (339A)

<https://codeforces.com/problemset/problem/339/A>

Tóm tắt đề bài

Sắp xếp lại các số hạng trong phép cộng (ví dụ " $3+2+1 \rightarrow 1+2+3$ ").

Phân tích Chiến lược

- **Quy trình:** String → Vector → String.
- **Parsing:** Tách các số (bỏ qua dấu '+'). Đẩy vào vector<int> hoặc vector<char>.
- **Sort:** Dùng std::sort.
- **Output:** In lại kèm dấu '+' ở giữa.

18. Magnets (344A)

<https://codeforces.com/problemset/problem/344/A>

Tóm tắt đề bài

Đếm số nhóm nam châm liên tiếp. Nhóm mới hình thành khi hai cực gặp nhau đẩy nhau (0-0 hoặc 1-1).

18. Magnets (344A)

<https://codeforces.com/problemset/problem/344/A>

Tóm tắt đề bài

Đếm số nhóm nam châm liên tiếp. Nhóm mới hình thành khi hai cực gặp nhau đẩy nhau (0-0 hoặc 1-1).

Phân tích Chiến lược

- **Chủ đề:** Phát hiện chuyển đổi (Transition detection).
- **Logic:** Một nhóm mới được tính khi mã nam châm hiện tại **khác** nam châm trước đó.
- **Code:** Duyệt i từ 1 đến n , nếu `magnet[i] != magnet[i-1]` thì tăng đếm.

19. Trạm (116A)

<https://codeforces.com/problemset/problem/116/A>

Tóm tắt đề bài

Tính sức chứa tối thiểu của tàu điện dựa trên số người lên/xuống tại các trạm (tìm số khách cực đại tại một thời điểm).

19. Trạm (116A)

<https://codeforces.com/problemset/problem/116/A>

Tóm tắt đề bài

Tính sức chứa tối thiểu của tàu điện dựa trên số người lên/xuống tại các trạm (tìm số khách cực đại tại một thời điểm).

Phân tích Chiến lược

- **Chủ đề:** Tổng tiền tố (Prefix Sum) / Biến tích lũy.
- **Thực hiện:** `current -= exit; current += enter; max_cap = max(max_cap, current);`
- Kết quả là giá trị lớn nhất ghi nhận được trong suốt hành trình.

20. Ultra-Fast Mathematician (61A)

<https://codeforces.com/problemset/problem/61/A>

Tóm tắt đề bài

Thực hiện XOR hai số nhị phân có độ dài lớn (lên tới 100 chữ số).

20. Ultra-Fast Mathematician (61A)

<https://codeforces.com/problemset/problem/61/A>

Tóm tắt đề bài

Thực hiện XOR hai số nhị phân có độ dài lớn (lên tới 100 chữ số).

Phân tích Chiến lược

- **Vấn đề:** Số quá lớn (Big Integer), không thể lưu bằng long long.
- **Giải pháp:** Xử lý dưới dạng chuỗi ký tự.
- **Logic:** Duyệt từng vị trí i : Nếu $s1[i] == s2[i] \rightarrow$ in '0'. Nếu $s1[i] != s2[i] \rightarrow$ in '1'.

Tổng kết Lộ trình phát triển

Kỹ năng đạt được

- Thao tác String: ASCII, Mutation, Normalization.
- Tư duy Vector: Sắp xếp, biến đổi dữ liệu.
- Tối ưu hóa: Phân biệt $O(N)$ vs $O(N^2)$.

Bước tiếp theo

- ① Lý thuyết Số cơ bản (Nguyên tố, GCD).
- ② Tìm kiếm Nhị phân (Binary Search).
- ③ Cấu trúc dữ liệu nâng cao: Map, Stack, Queue.

Chúc các bạn học tốt!