

Chiến lược Giải bài toán Cấu tạo số Codeforces Rating 800-1000

Biên soạn từ Tài liệu Nghiên cứu

Ngày 20 tháng 1 năm 2026

Tổng quan về Phân khúc Rating 800-1000

Tầm quan trọng

- Rating 800-900: Yêu cầu kỹ năng cài đặt (implementation) thuần túy.
- Rating 1000: Bắt đầu yêu cầu quan sát toán học và tư duy tối ưu.
- Mục tiêu: Rèn luyện phản xạ cho các kỳ thi Div. 2 hoặc Div. 3.

Kỹ thuật cốt lõi

- 1 **Vòng lặp while:** Trích xuất chữ số ($n \% 10$, $n / 10$).
- 2 **Xử lý chuỗi (String):** Dùng cho số lớn vượt quá long long hoặc cần thao tác vị trí.
- 3 **Câu lệnh điều kiện:** Kiểm tra tính chia hết, số may mắn.

1. Sum of Round Numbers (1352A) - 800

Link: <https://codeforces.com/problemset/problem/1352/A>

Đề bài

Cho một số nguyên dương n . Hãy phân tích n thành tổng của ít nhất các "số tròn" (số chỉ có 1 chữ số khác 0). In ra số lượng và các số hạng đó.

Ví dụ: $5009 \rightarrow 5000 + 9$.

1. Sum of Round Numbers (1352A) - 800

Link: <https://codeforces.com/problemset/problem/1352/A>

Đề bài

Cho một số nguyên dương n . Hãy phân tích n thành tổng của ít nhất các "số tròn" (số chỉ có 1 chữ số khác 0). In ra số lượng và các số hạng đó.

Ví dụ: $5009 \rightarrow 5000 + 9$.

Hướng dẫn giải

- Duyệt n từ hàng đơn vị lên (dùng $n \% 10$).
- Duy trì biến đếm lũy thừa của 10 (1, 10, 100...).
- Nếu chữ số hiện tại khác 0: nhân chữ số đó với lũy thừa 10 hiện tại và lưu vào kết quả.
- Cập nhật $n = n / 10$.

2. Nearly Lucky Number (110A) - 800

Link: <https://codeforces.com/problemset/problem/110/A>

Đề bài

Số may mắn là số chỉ chứa các chữ số 4 và 7. Cho số n , hãy kiểm tra xem số lượng các chữ số may mắn trong n có phải là một số may mắn hay không.

2. Nearly Lucky Number (110A) - 800

Link: <https://codeforces.com/problemset/problem/110/A>

Đề bài

Số may mắn là số chỉ chứa các chữ số 4 và 7. Cho số n , hãy kiểm tra xem số lượng các chữ số may mắn trong n có phải là một số may mắn hay không.

Hướng dẫn giải

- Đọc n dưới dạng chuỗi hoặc dùng `while` để tách số.
- Đếm số lượng chữ số 4 và 7 (gọi là `cnt`).
- Kiểm tra: Nếu `cnt == 4` hoặc `cnt == 7` thì in "YES", ngược lại "NO".
- Lưu ý: Không kiểm tra n , mà kiểm tra `cnt`.

3. Wrong Subtraction (977A) - 800

Link: <https://codeforces.com/problemset/problem/977/A>

Đề bài

Thực hiện phép giảm số n đi k lần theo quy tắc:

- Nếu chữ số cuối của n khác 0, giảm n đi 1.
- Nếu chữ số cuối của n là 0, chia n cho 10.

3. Wrong Subtraction (977A) - 800

Link: <https://codeforces.com/problemset/problem/977/A>

Đề bài

Thực hiện phép giảm số n đi k lần theo quy tắc:

- Nếu chữ số cuối của n khác 0, giảm n đi 1.
- Nếu chữ số cuối của n là 0, chia n cho 10.

Hướng dẫn giải

- Dùng vòng lặp chạy k lần.
- Trong mỗi lần lặp: kiểm tra `if (n % 10 != 0) n--; else n /= 10;`.
- Đây là bài toán mô phỏng trực tiếp.

4. Lucky Division (122A) - 1000

Link: <https://codeforces.com/problemset/problem/122/A>

Đề bài

Kiểm tra xem số n ($1 \leq n \leq 1000$) có chia hết cho bất kỳ "số may mắn" nào không. Số may mắn là số chỉ chứa 4 và 7 (vd: 4, 7, 44, 47, 74, 444...).

4. Lucky Division (122A) - 1000

Link: <https://codeforces.com/problemset/problem/122/A>

Đề bài

Kiểm tra xem số n ($1 \leq n \leq 1000$) có chia hết cho bất kỳ "số may mắn" nào không. Số may mắn là số chỉ chứa 4 và 7 (vd: 4, 7, 44, 47, 74, 444...).

Hướng dẫn giải

- Cách 1 (Pre-calculation): Liệt kê các số may mắn nhỏ hơn 1000: {4, 7, 44, 47, 74, 77, 444, 447, 474, 477, 744, 747, 777}.
- Duyệt qua danh sách, nếu n chia hết cho số nào thì in "YES".
- Cách 2: Hàm đệ quy sinh số may mắn và kiểm tra.

5. Sum of Digits (102B) - 1000

Link: <https://codeforces.com/problemset/problem/102/B>

Đề bài

Cho số nguyên n (có thể rất lớn, tới 10^{100000}). Thay thế n bằng tổng các chữ số của nó. Lặp lại cho đến khi n chỉ còn 1 chữ số. Đếm số lần thực hiện.

5. Sum of Digits (102B) - 1000

Link: <https://codeforces.com/problemset/problem/102/B>

Đề bài

Cho số nguyên n (có thể rất lớn, tới 10^{100000}). Thay thế n bằng tổng các chữ số của nó. Lặp lại cho đến khi n chỉ còn 1 chữ số. Đếm số lần thực hiện.

Hướng dẫn giải

- Bước 1: Đọc n vào biến string (vì số quá lớn).
- Bước 2: Nếu độ dài chuỗi bằng 1, kết quả là 0.
- Bước 3: Tính tổng chữ số, gán lại vào một biến số nguyên sum. Tăng biến đếm.
- Bước 4: Trong khi $\text{sum} \geq 10$, tiếp tục tính tổng các chữ số của sum.

6. Lucky Sum of Digits (109A) - 1000

Link: <https://codeforces.com/problemset/problem/109/A>

Đề bài

Tìm số nhỏ nhất chỉ gồm các chữ số 4 và 7 sao cho tổng các chữ số của nó đúng bằng n . Nếu không có, in -1.

6. Lucky Sum of Digits (109A) - 1000

Link: <https://codeforces.com/problemset/problem/109/A>

Đề bài

Tìm số nhỏ nhất chỉ gồm các chữ số 4 và 7 sao cho tổng các chữ số của nó đúng bằng n . Nếu không có, in -1.

Hướng dẫn giải

- Để số là nhỏ nhất, số lượng chữ số phải ít nhất \rightarrow Ưu tiên dùng nhiều số 7 nhất có thể.
- Giải phương trình $4x + 7y = n$.
- Duyệt y (số lượng số 7) giảm dần từ $n/7$ về 0.
- Kiểm tra phần dư còn lại $(n - 7y)$ có chia hết cho 4 không. Nếu có, đó là cấu hình tối ưu.
- In ra y số 4 và x số 7 (Lưu ý: đề yêu cầu số nhỏ nhất, nên in ít chữ số nhất, nhưng theo thứ tự số học thì 4 đứng trước 7. Cần kiểm tra lại logic sắp xếp: Cần in 4 trước 7 để giá trị nhỏ nhất).

7. Digital Root (1107B) - 1000

Link: <https://codeforces.com/problemset/problem/1107/B>

Đề bài

Căn nguyên số (digital root) của x là giá trị thu được khi liên tục tính tổng chữ số cho đến khi còn 1 chữ số (giống bài 102B). Tìm số nguyên dương thứ k có căn nguyên số bằng x .

7. Digital Root (1107B) - 1000

Link: <https://codeforces.com/problemset/problem/1107/B>

Đề bài

Căn nguyên số (digital root) của x là giá trị thu được khi liên tục tính tổng chữ số cho đến khi còn 1 chữ số (giống bài 102B). Tìm số nguyên dương thứ k có căn nguyên số bằng x .

Hướng dẫn giải

- Nhận xét toán học: Căn nguyên số của n chính là $n \% 9$ (nếu $n \% 9 == 0$ thì là 9).
- Các số có cùng căn nguyên số x cách nhau 9 đơn vị:
 $x, x + 9, x + 18, \dots$
- Công thức $O(1)$: Kết quả $= (k - 1) * 9 + x$.
- Dùng long long vì k có thể lớn.

8. Extremely Round (1766A) - 800

Link: <https://codeforces.com/problemset/problem/1766/A>

Đề bài

Số "cực tròn" là số chỉ có đúng một chữ số khác 0 (ví dụ: 5, 9, 10, 300).
Đếm xem có bao nhiêu số cực tròn $\leq n$.

8. Extremely Round (1766A) - 800

Link: <https://codeforces.com/problemset/problem/1766/A>

Đề bài

Số "cực tròn" là số chỉ có đúng một chữ số khác 0 (ví dụ: 5, 9, 10, 300).
Đếm xem có bao nhiêu số cực tròn $\leq n$.

Hướng dẫn giải

- Các số cực tròn có dạng: $1, \dots, 9$ (9 số), $10, \dots, 90$ (9 số), $100, \dots, 900 \dots$
- Cách 1: Pre-calculate danh sách tất cả số cực tròn (chỉ có khoảng 90 số trong giới hạn 10^9) rồi đếm.
- Cách 2: Dựa vào độ dài L của n và chữ số đầu tiên d . Kết quả $\approx (L - 1) * 9 + d$. Xử lý biên cẩn thận.

9. Insert Digit (1811A) - 800

Link: <https://codeforces.com/problemset/problem/1811/A>

Đề bài

Cho một số nguyên dương n dưới dạng chuỗi và một chữ số d . Hãy chèn d vào vị trí bất kỳ trong n để thu được số lớn nhất có thể.

9. Insert Digit (1811A) - 800

Link: <https://codeforces.com/problemset/problem/1811/A>

Đề bài

Cho một số nguyên dương n dưới dạng chuỗi và một chữ số d . Hãy chèn d vào vị trí bất kỳ trong n để thu được số lớn nhất có thể.

Hướng dẫn giải

- Nguyên tắc tham lam (Greedy): Để số lớn nhất, chữ số lớn phải đứng ở hàng cao nhất có thể.
- Duyệt chuỗi từ trái sang phải. Chèn d ngay trước ký tự đầu tiên nhỏ hơn d .
- Nếu duyệt hết chuỗi mà chưa chèn (tức là d nhỏ hơn hoặc bằng tất cả các số), chèn d vào cuối cùng.

10. Digit Minimization (1684A) - 800

Link: <https://codeforces.com/problemset/problem/1684/A>

Đề bài

Cho số n . Bạn thực hiện hoán đổi hai chữ số bất kỳ, sau đó xóa chữ số cuối cùng. Lặp lại cho đến khi còn 1 chữ số. Tìm chữ số nhỏ nhất có thể đạt được.

10. Digit Minimization (1684A) - 800

Link: <https://codeforces.com/problemset/problem/1684/A>

Đề bài

Cho số n . Bạn thực hiện hoán đổi hai chữ số bất kỳ, sau đó xóa chữ số cuối cùng. Lặp lại cho đến khi còn 1 chữ số. Tìm chữ số nhỏ nhất có thể đạt được.

Hướng dẫn giải

- Trường hợp đặc biệt: Nếu n chỉ có 2 chữ số (vd: 87), ta bắt buộc xóa số cuối sau khi hoán đổi (hoặc không). Chiến lược tối ưu là đưa số nhỏ ra sau để bị xóa? Không, lượt cuối còn 1 số.
- Logic: Với n có 2 chữ số ab , ta đổi thành ba rồi xóa a , còn b . Vậy kết quả là chữ số hàng đơn vị ban đầu (số thứ 2).
- Với $n > 2$ chữ số: Ta luôn có thể hoán đổi để giữ lại chữ số nhỏ nhất bất kỳ trong n đến cuối cùng.
- Kết quả: Min(các chữ số) nếu len > 2 ; Chữ số thứ 2 nếu len $== 2$.

11. Nearest Interesting Number (1183A) - 800

Link: <https://codeforces.com/problemset/problem/1183/A>

Đề bài

Tìm số nguyên $x \geq a$ nhỏ nhất sao cho tổng các chữ số của x chia hết cho 4.

11. Nearest Interesting Number (1183A) - 800

Link: <https://codeforces.com/problemset/problem/1183/A>

Đề bài

Tìm số nguyên $x \geq a$ nhỏ nhất sao cho tổng các chữ số của x chia hết cho 4.

Hướng dẫn giải

- Vì yêu cầu chia hết cho 4 (số nhỏ), khoảng cách giữa các số thỏa mãn rất ngắn.
- Dùng vòng lặp `while(true)` bắt đầu từ a .
- Viết hàm `sumDigits(x)`. Nếu `sumDigits(x) % 4 == 0` thì in x và dừng lại.
- Tăng x lên 1 sau mỗi lần kiểm tra.

12. Distinct Digits (1228A) - 800

Link: <https://codeforces.com/problemset/problem/1228/A>

Đề bài

Tìm một số x trong đoạn $[l, r]$ sao cho tất cả các chữ số của x đều khác nhau. Nếu không có, in -1.

12. Distinct Digits (1228A) - 800

Link: <https://codeforces.com/problemset/problem/1228/A>

Đề bài

Tìm một số x trong đoạn $[l, r]$ sao cho tất cả các chữ số của x đều khác nhau. Nếu không có, in -1.

Hướng dẫn giải

- Giới hạn bài toán thường nhỏ hoặc mật độ số thỏa mãn cao.
- Duyệt i từ l đến r .
- Với mỗi i , chuyển sang chuỗi hoặc dùng mảng đánh dấu để kiểm tra trùng lặp chữ số.
- Gặp số thỏa mãn đầu tiên thì in ra và kết thúc.

13. Sum of 2050 (1517A) - 800

Link: <https://codeforces.com/problemset/problem/1517/A>

Đề bài

Một số được gọi là "số 2050" nếu nó là 2050×10^k . Cho số n , hãy biểu diễn n thành tổng của ít nhất các số 2050. In ra số lượng số hạng, hoặc -1 nếu không thể.

13. Sum of 2050 (1517A) - 800

Link: <https://codeforces.com/problemset/problem/1517/A>

Đề bài

Một số được gọi là "số 2050" nếu nó là 2050×10^k . Cho số n , hãy biểu diễn n thành tổng của ít nhất các số 2050. In ra số lượng số hạng, hoặc -1 nếu không thể.

Hướng dẫn giải

- Điều kiện cần: n phải chia hết cho 2050. Nếu không $\rightarrow -1$.
- Nếu chia hết, đặt $q = n/2050$.
- Bài toán trở thành: Phân tích q thành tổng các lũy thừa của 10 (10^k).
- Số lượng ít nhất chính là tổng các chữ số của q .

14. Fair Numbers (1411B) - 1000

Link: <https://codeforces.com/problemset/problem/1411/B>

Đề bài

Số "công bằng" là số chia hết cho tất cả các chữ số khác 0 của nó. Tìm số công bằng nhỏ nhất $x \geq n$.

14. Fair Numbers (1411B) - 1000

Link: <https://codeforces.com/problemset/problem/1411/B>

Đề bài

Số "công bằng" là số chia hết cho tất cả các chữ số khác 0 của nó. Tìm số công bằng nhỏ nhất $x \geq n$.

Hướng dẫn giải

- Duyệt x bắt đầu từ $n, n + 1, \dots$
- Viết hàm kiểm tra: Tách từng chữ số d của x . Nếu $d \neq 0$ và $x \% d \neq 0$, thì x không thỏa mãn.
- Tại sao không quá thời gian? Bội chung nhỏ nhất của các chữ số từ 1-9 là 2520. Ta chắc chắn sẽ tìm thấy một số thỏa mãn trong khoảng rất ngắn.

15. Digits Sum (1553A) - 800

Link: <https://codeforces.com/problemset/problem/1553/A>

Đề bài

Đếm xem có bao nhiêu số x ($1 \leq x \leq n$) thỏa mãn: Tổng chữ số của x lớn hơn tổng chữ số của $x + 1$.

$(S(x) > S(x + 1))$.

15. Digits Sum (1553A) - 800

Link: <https://codeforces.com/problemset/problem/1553/A>

Đề bài

Đếm xem có bao nhiêu số x ($1 \leq x \leq n$) thỏa mãn: Tổng chữ số của x lớn hơn tổng chữ số của $x + 1$.

$(S(x) > S(x + 1))$.

Hướng dẫn giải

- Quan sát: $S(x + 1)$ chỉ nhỏ hơn $S(x)$ khi phép cộng 1 gây ra nhớ (carry), tức là chữ số tận cùng của x là 9.
- Ví dụ: $19 \rightarrow 20$ (Tổng: $10 \rightarrow 2$, giảm). $18 \rightarrow 19$ (Tổng tăng).
- Bài toán quy về: Đếm số lượng số kết thúc bằng 9 trong đoạn $[1, n]$.
- Công thức: $(n + 1)/10$.

16. Simple Design (1884A) - 800

Link: <https://codeforces.com/problemset/problem/1884/A>

Đề bài

Cho số nguyên x và k . Tìm số nguyên nhỏ nhất $y \geq x$ sao cho tổng các chữ số của y chia hết cho k .

16. Simple Design (1884A) - 800

Link: <https://codeforces.com/problemset/problem/1884/A>

Đề bài

Cho số nguyên x và k . Tìm số nguyên nhỏ nhất $y \geq x$ sao cho tổng các chữ số của y chia hết cho k .

Hướng dẫn giải

- Tương tự bài 1183A (Nearest Interesting Number).
- Duyệt từ x trở đi. Tính tổng chữ số.
- Kiểm tra tính chia hết cho k .
- Do k thường nhỏ (trong các bài 800), vòng lặp sẽ kết thúc rất nhanh.

17. Ordinary Numbers (1520B) - 800

Link: <https://codeforces.com/problemset/problem/1520/B>

Đề bài

Số "bình thường" là số có tất cả các chữ số giống nhau (vd: 1, 2, 99, 333).
Đếm số lượng số bình thường từ 1 đến n .

17. Ordinary Numbers (1520B) - 800

Link: <https://codeforces.com/problemset/problem/1520/B>

Đề bài

Số "bình thường" là số có tất cả các chữ số giống nhau (vd: 1, 2, 99, 333).
Đếm số lượng số bình thường từ 1 đến n .

Hướng dẫn giải

- Thay vì duyệt $1 \rightarrow n$, hãy sinh các số bình thường.
- Các số có dạng: $d \times \underbrace{11 \dots 1}_{k \text{ lần}}$.
- Duyệt độ dài k từ 1 đến 9 (hoặc 18). Duyệt chữ số d từ 1 đến 9.
- Tạo số val . Nếu $val \leq n$, tăng biến đếm.

18. Dislike of Threes (1560A) - 800

Link: <https://codeforces.com/problemset/problem/1560/A>

Đề bài

Polycarp không thích số chia hết cho 3 hoặc số có tận cùng là 3. Hãy tìm số thứ k trong dãy số nguyên dương đã loại bỏ các số Polycarp ghét.

18. Dislike of Threes (1560A) - 800

Link: <https://codeforces.com/problemset/problem/1560/A>

Đề bài

Polycarp không thích số chia hết cho 3 hoặc số có tận cùng là 3. Hãy tìm số thứ k trong dãy số nguyên dương đã loại bỏ các số Polycarp ghét.

Hướng dẫn giải

- Dùng vòng lặp để sinh dãy số thỏa mãn.
- Biến chạy i (số tự nhiên), biến đếm count (số thứ tự trong dãy mới).
- Nếu $i \% 3 == 0$ hoặc $i \% 10 == 3$, bỏ qua.
- Ngược lại, tăng count. Nếu $count == k$, in i .

19. Div. 7 (1633A) - 800

Link: <https://codeforces.com/problemset/problem/1633/A>

Đề bài

Cho số nguyên n . Hãy thay đổi ít chữ số của n nhất có thể để n chia hết cho 7.

19. Div. 7 (1633A) - 800

Link: <https://codeforces.com/problemset/problem/1633/A>

Đề bài

Cho số nguyên n . Hãy thay đổi ít chữ số của n nhất có thể để n chia hết cho 7.

Hướng dẫn giải

- Nếu $n \% 7 == 0$, in n .
- Nếu không, ta luôn có thể thay đổi chỉ **chữ số hàng đơn vị** để số đó chia hết cho 7.
- Duyệt các số từ $(n/10) * 10$ đến $(n/10) * 10 + 9$. Chắc chắn có ít nhất một số chia hết cho 7 trong mỗi khoảng 10 số liên tiếp.
- Chọn số trong khoảng đó chia hết cho 7.

20. Primary Task (2000A) - 1000

Link: <https://codeforces.com/problemset/problem/2000/A>

Đề bài

Kiểm tra xem số n có dạng $10^k + x$ với $k \geq 2$ và $x \geq 2$ (được viết dưới dạng ghép chuỗi "10" và phần mũ x ?)

Lưu ý: Đề bài thực tế của 2000A (Codeforces Round 962) yêu cầu kiểm tra định dạng "10" theo sau là một số mũ hợp lệ (số ≥ 2 và không có số 0 vô nghĩa ở đầu).

20. Primary Task (2000A) - 1000

Link: <https://codeforces.com/problemset/problem/2000/A>

Đề bài

Kiểm tra xem số n có dạng $10^k + x$ với $k \geq 2$ và $x \geq 2$ (được viết dưới dạng ghép chuỗi "10" và phần mũ x ?)

Lưu ý: Đề bài thực tế của 2000A (Codeforces Round 962) yêu cầu kiểm tra định dạng "10" theo sau là một số mũ hợp lệ ($s \geq 2$ và không có số 0 vô nghĩa ở đầu).

Hướng dẫn giải

- Chuyển n thành chuỗi s .
- Điều kiện 1: Độ dài $s \geq 3$.
- Điều kiện 2: Hai ký tự đầu phải là "10".
- Điều kiện 3: Ký tự thứ 3 không được là '0' (trừ khi số đó là 0, nhưng mũ phải ≥ 2).
- Điều kiện 4: Phần còn lại của chuỗi (từ index 2) chuyển thành số

Tổng kết và Lưu ý

Các lỗi thường gặp

- **Tràn số:** Luôn kiểm tra xem biến có vượt quá 2×10^9 không. Dùng long long khi cần thiết.
- **Biên (Corner cases):** Số 0, số 1, chuỗi rỗng, hoặc số có tận cùng là 0 khi đảo ngược.
- **Thời gian:** Tránh vòng lặp lồng nhau quá lớn ($O(N^2)$) nếu $N > 10^4$.

Lời khuyên

Hãy tự code lại 20 bài này mà không nhìn lời giải để thực sự nắm vững các cấu trúc điều khiển cơ bản.