

Tuyển tập 20 Bài tập Quy hoạch động trên Codeforces (Rating 800-1000)

Phân tích Hệ thống và Hướng giải quyết

Biên soạn từ tài liệu

Ngày 18 tháng 1 năm 2026

Giới thiệu chung

Mục tiêu

Nghiên cứu và áp dụng thuật toán Quy hoạch động (DP) trong phạm vi xếp hạng 800-1000 trên Codeforces. Đây là bước chuyển đổi quan trọng từ tư duy lập trình cơ bản sang tư duy thuật toán chuyên sâu.

Đặc điểm

Các bài toán thường yêu cầu sự kết hợp giữa kỹ năng triển khai mã nguồn và khả năng nhận diện các cấu trúc con tối ưu.

- **Phương pháp:** Memoization hoặc Tabulation đơn giản.
- **Dạng bài:** DP một chiều hoặc đếm trạng thái.

Phân loại theo mức xếp hạng

Rating	Đặc điểm chính	Kỹ năng yêu cầu
800	Truy hồi đơn giản, đếm trạng thái tuyến tính.	Cấu trúc vòng lặp, mảng một chiều.
900	Tối ưu hóa lựa chọn, mảng cộng dồn.	Tư duy tham lam kết hợp lưu trữ trạng thái.
1000	Bài toán cái túi đơn giản, DP trên chuỗi/số học.	Nhận diện cấu trúc con tối ưu.

Bảng: Tổng hợp đặc điểm bài toán DP theo Rating

1. The Way to Home (910A)

- **Link:** <https://codeforces.com/problemset/problem/910/A>
- **Rating:** 800

Đề bài

Một chú ếch ở vị trí 1 muốn đến vị trí n . Đường đi có các vị trí '1' (có hoa súng) hoặc '0'. Ếch nhảy tối đa d đơn vị vào nơi có hoa súng. Tìm số bước tối thiểu.

Hướng giải quyết

Bài toán tìm đường ngắn nhất trên mảng một chiều.

- Gọi $dp[i]$ là số bước tối thiểu đến vị trí i .
- Khởi tạo $dp[1] = 0$, còn lại là vô cùng.
- Với mỗi i có hoa súng: $dp[i] = \min(dp[j]) + 1$ với mọi j thỏa mãn $i - d \leq j < i$ và j có hoa súng.
- Độ phức tạp $O(N \cdot d)$ hoặc $O(N^2)$.

2. Subtract or Divide (1451A)

- **Link:** <https://codeforces.com/problemset/problem/1451/A>
- **Rating:** 900

Đề bài

Cho số nguyên n . Có thể giảm n đi 1 hoặc thay n bằng n/d (với d là ước của n). Tìm số bước tối thiểu để đưa n về 1.

Hướng giải quyết

Mối quan hệ giữa các số mô tả qua trạng thái $dp[i]$ (số bước tối thiểu biến i thành 1).

- Cân nhắc lựa chọn tối ưu giữa trừ 1 và chia.
- Thực tế có thể giải bằng Toán học/Tham lam dựa trên tính chẵn lẻ, nhưng tư duy DP giúp hiểu về chuyển trạng thái.

3. Alex and a Rhombus (1180A)

- **Link:** <https://codeforces.com/problemset/problem/1180/A>
- **Rating:** 800

Đề bài

Tính số ô vuông trong hình thoi bậc n được tạo ra bằng cách thêm các ô vào cạnh hình thoi bậc $n - 1$.

Hướng giải quyết

Minh họa cấu trúc truy hồi cơ bản:

- Gọi $f(n)$ là số ô vuông bậc n .
- Công thức truy hồi: $f(n) = f(n - 1) + 4(n - 1)$.
- Cơ sở: $f(1) = 1$.

4. Non-Substring Subsequence (1451B)

- **Link:** <https://codeforces.com/problemset/problem/1451/B>
- **Rating:** 900

Đề bài

Cho chuỗi s và truy vấn (l, r) . Kiểm tra xem có tồn tại chuỗi con bằng $s[l \dots r]$ nhưng không phải là chuỗi con liên tiếp tại vị trí đó không.

Hướng giải quyết

Lưu trữ trạng thái tồn tại của ký tự:

- Để chuỗi không liên tiếp, chỉ cần tìm xem có ký tự $s[l]$ nào ở vị trí $< l$ không, hoặc $s[r]$ nào ở vị trí $> r$ không.
- Sử dụng tư duy tiền xử lý (Preprocessing) giống DP để trả lời truy vấn nhanh.

5. Fair Division (1472B)

- **Link:** <https://codeforces.com/problemset/problem/1472/B>
- **Rating:** 800

Đề bài

Chia kẹo loại 1g và 2g thành hai phần có tổng trọng lượng bằng nhau.

Hướng giải quyết

Biến thể của bài toán Subset Sum (Cái túi):

- Gọi $dp[w]$ là trạng thái boolean: có thể tạo ra tổng trọng lượng w hay không.
- Do giá trị nhỏ, có thể kiểm tra tính chẵn lẻ của số lượng kẹo 1g và 2g kết hợp tổng trọng lượng.

6. Beat The Odds (1691A)

- **Link:** <https://codeforces.com/problemset/problem/1691/A>
- **Rating:** 800

Đề bài

Tìm số phần tử tối thiểu cần loại bỏ để tổng mọi cặp liên tiếp là chẵn.

Hướng giải quyết

- Tổng hai số là chẵn khi cả hai cùng chẵn hoặc cùng lẻ.
- Để mọi cặp có tổng chẵn, dãy còn lại phải toàn số chẵn hoặc toàn số lẻ.
- Đếm số lượng số chẵn (n_{even}) và số lẻ (n_{odd}).
- Kết quả: $\min(n_{\text{even}}, n_{\text{odd}})$.

7. Even Subset Sum Problem (1323A)

- **Link:** <https://codeforces.com/problemset/problem/1323/A>
- **Rating:** 800

Đề bài

Tìm một tập con (các chỉ số) sao cho tổng các phần tử là số chẵn.

Hướng giải quyết

- Nếu có phần tử chẵn: Chọn chính nó (tập kích thước 1).
- Nếu không có phần tử chẵn (toàn lẻ): Chọn 2 phần tử lẻ bất kỳ (Tổng: Lẻ + Lẻ = Chẵn).
- Nếu $n = 1$ và phần tử đó lẻ: Vô nghiệm.

8. Ordinary Numbers (1520B)

- **Link:** <https://codeforces.com/problemset/problem/1520/B>
- **Rating:** 800

Đề bài

Đếm số lượng các số nguyên dương từ 1 đến n có tất cả các chữ số giống nhau.

Hướng giải quyết

Tiền đề cho Digit DP:

- Liệt kê các số có dạng 1, 2...9, 11, 22...99, 111...
- Đếm trực tiếp các số này thỏa mãn $\leq n$.
- Có tổng cộng rất ít số như vậy trong phạm vi bài toán.

9. Squares and Cubes (1619B)

- **Link:** <https://codeforces.com/problemset/problem/1619/B>
- **Rating:** 800

Đề bài

Đếm các số từ 1 đến n là số chính phương (k^2) hoặc lập phương (k^3).

Hướng giải quyết

Sử dụng Set để tránh trùng lặp hoặc Nguyên lý bao hàm-loại trừ:

- Số lượng = (Số số $x^2 \leq n$) + (Số số $x^3 \leq n$) - (Số số $x^6 \leq n$).
- x^6 là số vừa là chính phương vừa là lập phương.

10. Extremely Round (1766A)

- **Link:** <https://codeforces.com/problemset/problem/1766/A>
- **Rating:** 800

Đề bài

Đếm các số $\leq n$ chỉ có đúng một chữ số khác 0 (Ví dụ: 5, 10, 900).

Hướng giải quyết

Dạng Digit DP đơn giản hoặc liệt kê:

- Các số có dạng $d \times 10^k$ (với $d \in [1, 9]$).
- Duyệt qua độ dài và chữ số đầu tiên để đếm.

11. 2-3 Moves (1716A)

- **Link:** <https://codeforces.com/problemset/problem/1716/A>
- **Rating:** 800

Đề bài

Đứng tại 0, muốn đến n . Có thể nhảy $+2, +3, -2, -3$. Tìm số bước tối thiểu.

Hướng giải quyết

DP trạng thái hoặc Tham lam theo modulo:

- Nếu $n = 1$: Cần 2 bước $(+3, -2)$.
- Nếu $n \% 3 == 0$: $n/3$ bước.
- Nếu $n \% 3 == 1$: $n/3 + 1$ bước (bớt một bước $+3$, thay bằng hai bước $+2$).
- Nếu $n \% 3 == 2$: $n/3 + 1$ bước (thêm một bước $+2$).

12. Maximum GCD (1370A)

- **Link:** <https://codeforces.com/problemset/problem/1370/A>
- **Rating:** 800

Đề bài

Tìm gcd(a, b) lớn nhất với $1 \leq a < b \leq n$.

Hướng giải quyết

Tìm giá trị tối ưu:

- Để GCD lớn nhất là g , ta cần ít nhất g và $2g$ nằm trong khoảng $[1, n]$.
- Điều kiện $2g \leq n \Rightarrow g \leq n/2$.
- Đáp án luôn là $\lfloor n/2 \rfloor$.

13. Team Training (73B)

- **Link:** <https://codeforces.com/problemset/problem/73/B>
- **Rating:** 800 (Lưu ý: Tên bài có thể là 519C, ID 73B là bài khác, nhưng giải theo logic Team Training)

Đề bài

Có n lập trình viên kinh nghiệm và m người mới. Mỗi đội cần (1 kinh nghiệm, 2 mới) hoặc (2 kinh nghiệm, 1 mới). Tối đa bao nhiêu đội?

Hướng giải quyết

Bài toán chia tài nguyên có cấu trúc tối ưu:

- Số đội tối đa không vượt quá n , không vượt quá m .
- Tổng số người dùng là $3 \times k$.
- Kết quả: $\min(n, m, (n + m)/3)$.

14. No Casino in the Mountains (41B)

- **Link:** <https://codeforces.com/problemset/problem/41/B>
- **Rating:** 800

Đề bài

(Bài toán Martian Dollar) Mua đô la ngày i giá a_i và bán ngày $j > i$ giá a_j để tối đa hóa tiền.

Hướng giải quyết

- Duyệt qua từng ngày mua i .
- Tìm giá bán cao nhất trong các ngày sau i (có thể dùng DP hậu tố max suffix để tối ưu, hoặc duyệt trâu vì N nhỏ).
- Tính lợi nhuận tối đa cho từng phương án.

15. Minimum Varied Number (1714C)

- **Link:** <https://codeforces.com/problemset/problem/1714/C>
- **Rating:** 800

Đề bài

Tìm số nhỏ nhất có các chữ số khác nhau và tổng các chữ số bằng s .

Hướng giải quyết

Tham lam/Xây dựng từ cuối lên:

- Để số nhỏ nhất, số lượng chữ số phải ít nhất (các chữ số ở hàng đơn vị phải lớn nhất có thể).
- Lấy các số 9, 8, 7... lần lượt trừ vào s từ hàng đơn vị lên hàng cao hơn.

16. Tanya and Stairways (1005A)

- **Link:** <https://codeforces.com/problemset/problem/1005/A>
- **Rating:** 800

Đề bài

Đếm số cầu thang và số bậc của mỗi cầu thang dựa trên dãy số đếm bậc (1, 2, 3, 1, 2...).

Hướng giải quyết

- Mỗi khi gặp số 1, đó là bắt đầu một cầu thang mới.
- Số bậc của cầu thang trước đó chính là giá trị phần tử ngay trước số 1 mới (hoặc phần tử cuối cùng của dãy).

17. Polycarp's Pockets (1003A)

- **Link:** <https://codeforces.com/problemset/problem/1003/A>
- **Rating:** 800

Đề bài

Có n đồng xu với các mệnh giá. Chia vào các túi sao cho mỗi túi không có 2 đồng cùng mệnh giá. Tìm số túi tối thiểu.

Hướng giải quyết

- Số túi tối thiểu chính bằng tần suất xuất hiện lớn nhất của một mệnh giá bất kỳ trong mảng.
- Dùng mảng đếm (Frequency Array).

18. Construct the String (1335B)

- **Link:** <https://codeforces.com/problemset/problem/1335/B>
- **Rating:** 800

Đề bài

Xây dựng chuỗi độ dài n , mọi chuỗi con độ dài a có đúng b ký tự khác nhau.

Hướng giải quyết

Sử dụng tính chất chu kỳ:

- Tạo một mẫu chu kỳ gồm b ký tự khác nhau (ví dụ: 'abc...').
- Lặp lại mẫu này cho đến khi đủ độ dài n .
- $S[i] = 'a' + (i \bmod b)$.

20. Water Buying (1118A)

- **Link:** <https://codeforces.com/problemset/problem/1118/A>
- **Rating:** 800

Đề bài

Cần mua n lít nước. Chai 1L giá a , chai 2L giá b . Tìm chi phí tối thiểu.

Hướng giải quyết

Bài toán Knapsack cơ bản/Tham lam:

- So sánh giá mua 2 chai 1L ($2 \times a$) và 1 chai 2L (b).
- Nếu $2a \leq b$: Mua toàn bộ bằng chai 1L.
- Nếu $2a > b$: Ưu tiên mua chai 2L (với $n/2$ chai), nếu lẻ thì mua thêm 1 chai 1L.

Phân tích Sâu và Chiến lược Luyện tập

Sự hình thành tư duy

Ở mức 800-1000, khái niệm "trạng thái" và "truy hồi" được lồng ghép trong các bài toán triển khai (Implementation). Hiểu sâu về Prefix Sums (DP đơn giản) là bước đệm quan trọng.

Lộ trình đề xuất

- ➊ **Mức 800:** Chuyển đổi ý tưởng thành mã nguồn sạch.
- ➋ **Mức 900:** Vẽ cây quyết định hoặc bảng trạng thái trên giấy.
- ➌ **Mức 1000:** Làm quen bài toán con chồng lẫn (CP-31 Sheet).

Tương quan xếp hạng

Codeforces	USACO	Trình độ tương ứng
800 - 1000	Beginner / Bronze	Newbie / Pupil
1100 - 1500	Bronze / Silver	Pupil / Specialist

"Quy hoạch động không phải là rào cản, mà là cánh cửa mở ra những giải pháp thông minh."