

Cắt xâu

Giới hạn thời gian: 1.0s Giới hạn bộ nhớ: 256M

Bài toán cắt xâu sẽ thành các số tăng nghiêm ngặt: cho xâu S chỉ gồm ký tự '0'..'9'. Hãy đếm số cách cắt S thành các đoạn liên tiếp $S_1+S_2+\dots+S_k$ ($k>0$) sao cho mỗi đoạn khi diễn giải là một **số thập phân không có số 0 ở đầu** và dãy các số này **tăng nghiêm ngặt**.

Yêu cầu

Cho xâu S chỉ gồm ký tự '0'..'9'. Đếm số cách tách $S=S_1+S_2+\dots+S_k$ ($k>0$) thoả mãn:

- Ký tự đầu tiên của mỗi S_i khác '0' ($i=1..k$).
- Khi coi S_i là giá trị số nguyên dương ở cơ số 10, dãy $S_1, S_2, \dots, S_k$ tăng nghiêm ngặt.

Kết quả lấy theo modulo 10^9+7 .

Dữ liệu

- Một dòng duy nhất chứa xâu S **không bắt đầu bằng '0'**, với $1 \leq |S| \leq 5000$.

Kết quả

- Một số nguyên là số cách cắt thoả mãn, tính theo modulo 10^9+7 .

Ví dụ

Ví dụ 1

Input

112323

Output

5

Ví dụ 2

Input

10203

Output

2

Giải thích

Ví dụ 1

Các cách hợp lệ của $\sim S = \text{"112323"} \sim$ là:

- $\sim \text{"1"} + \text{"12"} + \text{"323"} \sim$
- $\sim \text{"1"} + \text{"12323"} \sim$
- $\sim \text{"11"} + \text{"2323"} \sim$
- $\sim \text{"112"} + \text{"323"} \sim$
- $\sim \text{"12323"} \sim$

Ví dụ 2

Với $\sim S = \text{"10203"} \sim$, các cách hợp lệ:

- $\sim \text{"10"} + \text{"203"} \sim$
- $\sim \text{"10203"} \sim$ Lưu ý: các đoạn không được bắt đầu bởi '0', do đó tách kiểu $\sim \text{"1"} + \underbrace{\text{"02"}}_{\text{không hợp lệ}} + \text{"03"} \sim$ bị loại.

Gợi ý (có thể đúng hoặc sai)

- 100% số điểm cho lời giải $\sim \mathcal{O}(n^2) \sim$ với $n = |S| \leq 5000$, dựa trên:
 - So sánh hai **số** dạng xâu $\sim X, Y \sim$: nếu $\sim |X| \leq |Y| \sim$ thì số có độ dài lớn hơn là lớn hơn; nếu bằng nhau, so sánh từ trái sang phải. Để tối ưu so sánh khi $\sim |X| = |Y| \sim$, tiền xử lý **bảng LCP** (Longest Common Prefix) giữa mọi cặp vị trí trong $\sim S \sim$ bằng kỹ thuật Z/Suffix và quy hoạch động, cho phép so sánh hai đoạn độ dài bằng nhau trong $\sim \mathcal{O}(1) \sim$.
 - Quy hoạch động theo điểm cắt cuối: đặt $\sim \text{dp}[i] \sim$ là số cách cắt tiền tố $\sim S[1..i] \sim$ kết thúc bằng một đoạn cuối $\sim S[j..i] \sim \sim (1 \leq j \leq i, S[j] \neq '0') \sim$; khi xét hai đoạn liên tiếp $\sim S[t..j-1] \sim$ và $\sim S[j..i] \sim$, điều kiện tăng nghiêm ngặt kiểm bằng quy tắc so sánh ở trên. Có thể tổ chức theo độ dài đoạn cuối và dùng **tổng lũy tích** để giảm từ $\sim \mathcal{O}(n^3) \sim$ tới $\sim \mathcal{O}(n^2) \sim$.
 - Lấy modulo $\sim 10^9 + 7 \sim$ ở mọi phép cộng.
- Mẹo triển khai:
 - Tiền xử lý $\sim \text{LCP}[i][j] \sim$ cho mọi cặp vị trí $\sim i < j \sim$ bằng kỹ thuật Z trên từng suffix (hoặc chuẩn bị mảng Z/Kasai tùy chọn), bộ nhớ $\sim \sim n^2 \sim$ với $n = 5000$ vẫn chấp nhận được trong giới hạn thi

- ($\sim \text{approx } 25 \times 10^6$ số nguyên). Có thể tối ưu bộ nhớ bằng so sánh trực tiếp khi độ dài nhỏ.
- Khi so sánh $S[a..b]$ và $S[c..d]$ có cùng độ dài ℓ , nếu $\text{LCP}[a][c] \geq \ell$ thì hai đoạn bằng nhau (không chấp nhận vì cần **tăng nghiêm ngặt**); ngược lại so ký tự ngay sau LCP để quyết định lớn/nhỏ.