

BÀI 1. TÌM SỐ

Ta có một nhận xét: Với một số nguyên dương x tồn tại trong dãy các bội số nguyên dương của 1 hoặc 2 số a hoặc b , thì số lượng số nguyên dương trong dãy nhỏ hơn hoặc bằng x sẽ là $\frac{x}{a} + \frac{x}{b} - \frac{x}{[a,b]}$ với $[a, b]$ là BCNN của hai số a và b (quy tắc bao hàm loại trừ).

→ Tìm kiếm nhị phân với số nguyên x (vì a, b cố định) với x là số nhỏ nhất thỏa mãn:

$$\frac{x}{a} + \frac{x}{b} - \frac{x}{[a,b]} \geq n. \text{ Từ đó, } x \text{ sẽ là đáp án của bài toán.}$$

BÀI 2. ĐIỂM THƯỜNG

- *Subtask 1: $N, M \leq 10^3$ [60% số điểm];*

Duyệt toàn bộ mảng B với mỗi lần duyệt các phần tử trong mảng A , nếu $B[j] < A[i]$ thì tăng biến đếm T .

Độ phức tạp: $O(N * M)$

- *Subtask 2, 3: $N, M \leq 10^6$ [90% số điểm];*

Theo tính chất của dãy, ta có nhận xét: Nếu tồn tại một chỉ số j sao cho $B[j] \geq A[i]$ thì các số $B[j+1..n]$ đều cũng $\geq A[i]$. Từ đó ta chỉ có việc tìm j nhỏ nhất sao cho $B[j] \geq A[i]$ bằng cách tìm kiếm nhị phân trên mảng $B \rightarrow$ Khi đó $j - 1$ sẽ là vị trí lớn nhất sao cho $B[j] < A[i]$.

Độ phức tạp: $O(N * \log(M))$

- *Subtask 4: $N, M \leq 5 \times 10^6$*

Sử dụng CTDL stack tối ưu hóa thuật toán: Đẩy toàn bộ các phần tử của mảng B vào dãy A . Duyệt ngược mảng A , lấy ra khỏi stack các phần tử có giá trị $\geq A[i]$, số lượng phần tử còn lại trong stack sẽ là các số thỏa mãn đối với từng phần tử $A[i]$.

Độ phức tạp: $O(2 * M) \approx O(10^7)$.

BÀI 3. GIẢI MÃ

Chú thích: n là số lượng dấu thăng trong bưu thiếp của BigZero, $|s|$ là độ dài của xâu s , t_i là chuỗi các chữ cái có thể thay thế cho dấu # thứ i .

Trước khi xử lý ta cần phải sắp xếp các xâu chứa các ký tự có thể thay thế theo chiều tăng dần của thứ tự từ điển. Lưu mảng $blank[]$ với $blank[i]$ là vị trí của chỗ trống thứ i .

Ta có một nhận xét: Khi mình xét ở dấu # thứ i , thì ở các dấu thăng sau sẽ có $|t_{i+1}| * |t_{i+2}| * ... * |t_n|$ cách điền cho mỗi chữ cái có thể điền trong dấu # thứ i , số lượng cách điền như trên có thể tính bằng mảng tích tiền tố $pref[]$.

Từ đó, với mỗi lần xét, ta duyệt từ đầu đến cuối xâu s_i để tìm ký tự thứ v cần thay thế cho dấu # thứ i , mỗi lần duyệt ta trừ x đi cho một lượng là $pref[n] / pref[i]$, và sẽ dừng lại khi $x \leq pref[n] / pref[i]$.

→ $s[blank[i]] = t[i][v]$.