

Sắp xếp đối xứng

Giới hạn thời gian: 1.0s Giới hạn bộ nhớ: 254M

Cho xâu $\sim S \sim$ độ dài $\sim N \sim$ chỉ gồm các ký tự từ $\sim 'a'..'z' \sim$. Bạn cần xử lý $\sim Q \sim$ truy vấn, mỗi truy vấn cho một đoạn $\sim [L..R] \sim$. Với mỗi truy vấn, **nếu có thể**, hãy **sắp xếp lại** (hoán vị) các ký tự trong xâu con $\sim S[L..R] \sim$ để thu được **một xâu đối xứng** (palindrome) và trong số các cách sắp xếp đối xứng hợp lệ, hãy chọn cách cho thứ tự **từ điển nhỏ nhất** của **toàn bộ** xâu sau khi thay thế đoạn đó. Nếu **không thể** tạo palindrome cho đoạn $\sim [L..R] \sim$ (do ràng buộc tần suất ký tự), **giữ nguyên** đoạn này.

Sau khi thực hiện **tuần tự** tất cả $\sim Q \sim$ truy vấn, in ra xâu $\sim S \sim$ cuối cùng.

Yêu cầu

Thực hiện lần lượt $\sim Q \sim$ truy vấn trên xâu $\sim S \sim$ như mô tả.

- Nếu đoạn $\sim [L..R] \sim$ có thể sắp xếp thành palindrome, thay $\sim S[L..R] \sim$ bởi **palindrome có thứ tự từ điển nhỏ nhất** (trong phạm vi đoạn) và tiếp tục truy vấn tiếp theo dựa trên xâu đã cập nhật.
- Nếu không thể, **không thay đổi** đoạn và tiếp tục.

Dữ liệu

- Dòng 1:** Hai số nguyên $\sim N \sim, \sim Q \sim \sim (1 \leq N, Q \leq 10^5) \sim$.
- Dòng 2:** Xâu $\sim S \sim$ có độ dài $\sim N \sim$, chỉ gồm ký tự $\sim 'a'..'z' \sim$.
- Dòng 3..Q+2:** Mỗi dòng chứa hai số nguyên $\sim L, R \sim \sim (1 \leq L \leq R \leq N) \sim$ — đoạn cần xử lý.

Kết quả

- In ra **một dòng** chứa xâu $\sim S \sim$ sau khi thực hiện xong $\sim Q \sim$ truy vấn.

Ví dụ 1

Input

```
7 3
aabbcbba
1 4
2 6
3 7
```

Output

```
ababcba
```

Giải thích

- Bước 1: $S[1..4] = \text{"aabb"}$ có thể xếp thành palindrome, chọn từ điển nhỏ nhất $\sim\text{"abba"}\sim \rightarrow \sim\text{"aabbcbba"}\sim \rightarrow \sim\text{"abbacba"}\sim$.
- Bước 2: $S[2..6] = \text{"bbacb"}$ có thể xếp thành palindrome nhỏ nhất là $\sim\text{"abcba"}\sim$, nhưng cần thay **trên đoạn** và giữ phần còn lại $\rightarrow$ xâu vẫn $\sim\text{"abbacba"}\sim$ (vì thay $\sim\text{"bbacb"}$ bằng $\sim\text{"abcba"}\sim$ cho kết quả giống).
- Bước 3: $S[3..7] = \text{"bacba"}$ $\rightarrow$ palindrome nhỏ nhất $\sim\text{"abcba"}\sim \rightarrow$ kết quả cuối $\sim\text{"ababcba"}\sim$.

Ví dụ 2

Input

```
8 2
abcdzzdc
1 8
2 7
```

Output

```
abcdzzdc
```

Giải thích

- Với đoạn $\sim[1..8]\sim$, tần suất ký tự có **2** ký tự lẻ (hai chữ $\sim\text{'z'}$ là chẵn, nhưng $\sim\text{'a','b','c','d'}$ đều lẻ) $\rightarrow$ **không thể** thành palindrome $\rightarrow$ bỏ qua.
- Đoạn $\sim[2..7]\sim$ cũng không thể $\rightarrow$ xâu giữ nguyên.

Gợi ý & Ràng buộc thuật toán (không bắt buộc khi cài đặt)

- Một đoạn $\sim[L..R]\sim$ có thể sắp xếp thành palindrome **khi và chỉ khi** số lượng ký tự có tần suất **lẻ** trong đoạn ≤ 1 .
- Palindrome **từ điển nhỏ nhất** trên đoạn thu được bằng cách:
 - Đặt lần lượt từ ký tự $\sim\text{'a'}$ đến $\sim\text{'z'}$ vào **nửa trái** (mỗi ký tự lấy $\lfloor \text{cnt}[c]/2 \rfloor$ lần),

- **Soi gương** nửa trái sang nửa phải theo thứ tự đối xứng,
- Nếu tồn tại ký tự có tần suất lẻ, đặt **một** ký tự đó ở **giữa**.
- Để đạt hiệu năng với $\sim N, Q \leq 10^5$, tham khảo cấu trúc dữ liệu:
 - **Segment tree** lưu **26** tần suất/ký tự cho mỗi nút, hỗ trợ: (i) **đếm tần suất theo đoạn**, (ii) **gán đoạn** thành một ký tự duy nhất (range assign) với **lazy propagation**.
 - Với mỗi truy vấn hợp lệ, bạn sẽ thực hiện **tối đa 26** phép gán đoạn để dựng nửa trái theo thứ tự từ $\sim 'a' \sim$ lên và phần soi gương cho nửa phải (cộng thêm gán 1 ký tự giữa nếu cần).
 - Bộ nhớ $\sim \mathcal{O}(26 \cdot 4N)$, thời gian xấp xỉ $\sim \mathcal{O}(26 \log N)$ cho mỗi truy vấn hợp lệ, $\sim \mathcal{O}(\log N)$ để kiểm tra tần suất.

Chấm điểm

Tổng ~ 100 điểm, gồm:

- **Subtask 1 (20 điểm):** $\sim 1 \leq N, Q \leq 2000$ — có thể thao tác trực tiếp trên mảng/ký tự.
- **Subtask 2 (30 điểm):** Chỉ có truy vấn trên các đoạn có độ dài $\sim \leq 2000$.
- **Subtask 3 (50 điểm):** Ràng buộc đầy đủ $\sim (1 \leq N, Q \leq 10^5)$ — yêu cầu cấu trúc dữ liệu như gợi ý để đạt thời gian hợp lệ.

Lưu ý cài đặt (tùy chọn):

- Có thể nén trạng thái **đoạn toàn ký tự** $\sim c \sim$ bằng **mask 26-bit** trong lazy tag để **gán nhanh**.
- Lúc dựng palindrome, dùng hai con trỏ $\sim i=L, \sim j=R$; với mỗi chữ $\sim c \sim$ từ $\sim 'a' \sim \dots \sim 'z' \sim$, gán đoạn $\sim [i, i+k-1] \sim$ là $\sim c \sim$ với $\sim k=\lfloor \text{cnt}[c]/2 \rfloor$, đồng thời gán đối xứng $\sim [j-k+1, j] \sim$ là $\sim c \sim$, rồi cập nhật $\sim i \leftarrow i+k, \sim j \leftarrow j-k$. Nếu còn ký tự lẻ, gán vị trí giữa $\sim i=j$ bằng chữ đó.