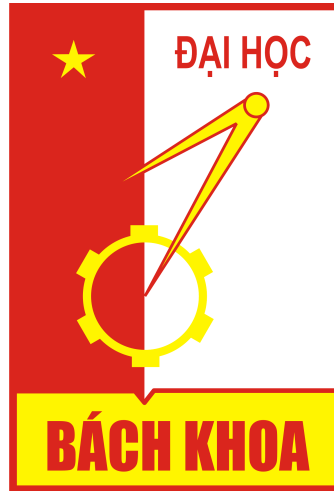


TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG
_____*



TÌM HIỂU VỀ FLEX

Sinh viên thực hiện : **Nguyễn Quang Huy**

MSSV : 20151690

Lớp : KSTN-CNTT-K60

HÀ NỘI
Ngày 14 tháng 11 năm 2018

Mục lục

1	Thuật toán Mo	3
1.1	Thuật toán cơ bản	3
1.2	Thuật toán Mo	3
2	Tài liệu tham khảo	4

1 Thuật toán Mo

Cho một dãy số A gồm N phần tử. Cần thực hiện Q truy vấn, mỗi truy vấn (i, j) yêu cầu tìm $\text{mode}(A_i, \dots, A_j)$. (Mode của một tập hợp là giá trị xuất hiện nhiều lần nhất trong tập hợp đó). Giới hạn: $N, Q, A_i \leq 10^5$.

1.1 Thuật toán cơ bản

Có thể giải bài toán bằng thuật toán đơn giản như sau:

- Với mỗi truy vấn, duyệt từ trái sang phải theo độ dài của câu truy vấn, đếm số lần xuất hiện của các phần tử.
- Trong khi đếm thực hiện cập nhật kết quả.

Code đơn giản như sau:

```
void mode(1,r):
    res = -1;
    for i = 1 .. r :
        count[A[i]] += 1;
        if res == -1 or count[A[i]] > count[res]:
            res = A[i];
    return res;
```

Dễ thấy, thuật toán này có độ phức tạp $\mathcal{O}(N*Q)$. Có 2 lí do chính khiến thuật toán này chạy chậm:

1. Khởi tạo mảng count mỗi lần mất $\mathcal{O}(N)$.
2. Với mỗi truy vấn, phải tính lại mảng count từ đầu.

Có thể cải tiến thuật toán:

Sau khi trả lời truy vấn $[l_1, r_1]$, để trả lời truy vấn $[l_2, r_2]$, chỉ cần thay đổi mảng đếm một cách phù hợp. Cụ thể:

- Nếu $l_2 > l_1$, giảm số lần xuất hiện của $A_{l_1}, \dots, A_{l_2-1}$
- Nếu $l_2 < l_1$, tăng số lần xuất hiện của $A_{l_2}, \dots, A_{l_1-1}$
- Tương tự với r_1 và r_2 . Để cập nhật số lần xuất hiện lớn nhất có thể dùng thêm set. Như vậy, độ phức tạp thuật toán là tổng $|l_i - l_{i-1}| + |r_i - r_{i-1}|$, nhân thêm $\mathcal{O}(\log N)$ để đếm và tìm phần tử lớn nhất của mảng đếm.

1.2 Thuật toán Mo

Thuật toán Mo là một cách sắp xếp lại các truy vấn, sao cho tổng $|l_i - l_{i-1}| + |r_i - r_{i-1}|$ không quá $\mathcal{O}(N * \sqrt{N} + Q * \sqrt{N})$.

Thứ tự các truy vấn được định nghĩa qua hàm so sánh dưới đây:

```
S = sqrt(N);
bool cmp(Query A, Query B){
    if(A.l / S != B.l / S)
```

2 Tài liệu tham khảo

- Lexical Analysis with Flex
- Flex & Bison
- [https://en.wikipedia.org/wiki/Flex_\(lexical_analyser_generator\)](https://en.wikipedia.org/wiki/Flex_(lexical_analyser_generator))
- ftp://ftp.gnu.org/old-gnu/Manuals/flex-2.5.4/html_mono/flex.html