

Sliding Window

BÁO CÁO KỸ THUẬT SLIDING WINDOW

◆ 1. Giới thiệu

Sliding Window (cửa sổ trượt) là một kỹ thuật lập trình giúp tối ưu các bài toán yêu cầu xét các đoạn con liên tiếp trong mảng hoặc chuỗi. Thay vì kiểm tra mọi đoạn con một cách ngây thơ (brute-force), ta chỉ dịch chuyển giới hạn cửa sổ để tận dụng thông tin đã tính toán trước.

◆ 2. Ứng dụng điển hình

Sliding Window thường dùng để giải các bài toán sau:

- Đếm hoặc tìm đoạn con (subarray) thỏa mãn điều kiện.
- Tìm độ dài đoạn con lớn nhất/nhỏ nhất.
- Tối ưu hóa chi phí, tổng, trung bình,... trên đoạn con liên tiếp.

Ví dụ tiêu biểu:

- Đếm số mảng con có tổng nhỏ hơn hoặc bằng K.
- Tìm xâu con có số lượng ký tự đặc biệt không vượt quá ngưỡng.
- Tìm đoạn con có hiệu giữa phần tử lớn nhất và nhỏ nhất nhỏ hơn hoặc bằng K.

◆ 3. Cách cài đặt

Sliding Window có hai dạng:

◆ a. Cửa sổ cố định

- Dùng khi độ dài cửa sổ không thay đổi.
- Thường áp dụng với bài toán tính trung bình, tổng,... trên đoạn độ dài K.

```
// Tổng các đoạn con dài K
for (int i = 0; i < K; i++) sum += A[i];
for (int i = K; i < N; i++) {
    sum += A[i] - A[i - K];
}
```

◆ b. Cửa sổ động

- Dùng khi cửa sổ thay đổi kích thước để thỏa điều kiện.
- Phổ biến trong các bài toán "đoạn con dài nhất thỏa mãn điều kiện".

```
int left = 0;
for (int right = 0; right < N; right++) {
    // Cập nhật trạng thái khi thêm A[right]
    while (/* điều kiện không thỏa */) {
        // Thu hẹp cửa sổ từ trái
        left++;
    }
}
```

```
// Cửa sổ từ left đến right là hợp lệ  
}
```

◆ 4. Tối ưu với deque và map

Một số bài toán cần duy trì phần tử lớn nhất/nhỏ nhất trong cửa sổ — sử dụng:

- **Deque** (double-ended queue) để lưu chỉ số phần tử cực trị.
- **Multiset / map** nếu cần đếm số lượng phần tử, hoặc duy trì thứ tự.

◆ 5. Độ phức tạp

- **Thời gian:** $O(N)$ – mỗi phần tử vào và ra khỏi cửa sổ chỉ một lần.
- **Bộ nhớ:** $O(K)$ hoặc $O(N)$ tùy bài toán.

◆ 6. Kết luận

Kỹ thuật **Sliding Window** là một công cụ mạnh mẽ để giải quyết các bài toán tối ưu trong mảng và chuỗi. Khi áp dụng đúng, nó giúp giảm độ phức tạp từ $O(N^2)$ xuống $O(N)$, phù hợp với các bài toán dữ liệu lớn trong lập trình thi đấu và thực tế.