

Tìm Độ Dài Lớn Nhất Của Dãy Con Liên Tiếp Có Tổng Bằng Giá Trị Cho Trước

Nguyễn Hoàng Tân

Ngày 2 tháng 1 năm 2025

Tóm Tắt Nội Dung

Bài toán yêu cầu tìm độ dài lớn nhất của một dãy con liên tiếp trong mảng sao cho tổng của dãy con bằng một giá trị mục tiêu. Báo cáo này trình bày chi tiết giải pháp sử dụng **Dynamic Programming** với kỹ thuật **Prefix Sum** và **Hash Map**. Giải pháp đảm bảo hiệu quả về thời gian ($O(n)$) và bộ nhớ ($O(n)$).

1 Giới Thiệu

Mô Tả Bài Toán

Cho một mảng số nguyên `arr` gồm n phần tử và một số nguyên `target`. Tìm độ dài lớn nhất của dãy con liên tiếp trong mảng sao cho tổng các phần tử của dãy bằng `target`.

Lịch Sử và Đặc Điểm Chính

Lịch sử: Bài toán này xuất hiện nhiều trong lĩnh vực xử lý dữ liệu, tối ưu hóa và thuật toán lập trình. Kỹ thuật sử dụng tổng tích lũy (Prefix Sum) kết hợp với bảng băm (Hash Map) đã được áp dụng để giải quyết bài toán hiệu quả.

Đặc điểm chính:

- Giải thuật sử dụng Dynamic Programming để tối ưu hóa thời gian và bộ nhớ.

- Tìm ra cấu trúc của bài toán bằng cách phân tích quan hệ giữa tổng tích lũy và giá trị mục tiêu.
- Đảm bảo thời gian thực thi tuyến tính $O(n)$ nhờ sử dụng bảng băm.

2 Động Lực

Phát Biểu Bài Toán

Cho mảng $\text{arr}[0 \dots n - 1]$ và giá trị mục tiêu target , tìm độ dài lớn nhất của dãy con liên tiếp sao cho:

$$\sum_{i=j}^k \text{arr}[i] = \text{target}, \quad \text{với } 0 \leq j \leq k < n.$$

Ứng Dụng Thực Tế

- **Phân tích chuỗi sự kiện:** Xác định khoảng thời gian có tổng giá trị bằng một ngưỡng mục tiêu.
- **Quản lý chi phí:** Tìm dãy con tối ưu để kiểm soát ngân sách.
- **Tối ưu hóa dữ liệu:** Áp dụng trong lĩnh vực xử lý tín hiệu và dữ liệu lớn.

3 Bối Cảnh và Công Việc Liên Quan

Bài toán liên quan đến các kỹ thuật:

- **Dynamic Programming trên đoạn:** Giải quyết bài toán với cấu trúc đoạn như bài toán nhân ma trận, chia đa giác.
- **Thuật toán tham lam:** Không hiệu quả trong bài toán này vì không tối ưu khi phải xử lý dãy con liên tiếp.
- **Prefix Sum và Hash Map:** Phương pháp hiện đại để giải bài toán hiệu quả.

4 Phân Tích: Giải Pháp Dynamic Programming

Công Thức Truy Hồi

Gọi $\text{prefix_sum}[i]$ là tổng tích lũy từ đầu mảng đến chỉ số i :

$$\text{prefix_sum}[i] = \text{arr}[0] + \text{arr}[1] + \dots + \text{arr}[i].$$

Nếu $\text{prefix_sum}[i] - \text{target}$ đã tồn tại trong bảng băm tại chỉ số j , thì:

$$\text{Độ dài dãy con} = i - j.$$

Mã Giải

```
function findMaxLengthSubarray(arr, n, target):
    prefix_sum = 0
    max_length = 0
    map = {}

    for i = 0 to n-1:
        prefix_sum += arr[i]

        if prefix_sum == target:
            max_length = i + 1

        if (prefix_sum - target) in map:
            max_length = max(max_length, i - map[prefix_sum - target])

        if prefix_sum not in map:
            map[prefix_sum] = i

    return max_length
```

5 Ví Dụ Minh Họa

Đầu Vào:

- $\text{arr} = [1, -1, 5, -2, 3]$
- $\text{target} = 3$

Diễn Giải:

- Tổng tích lũy: `prefix_sum` = [1, 0, 5, 3, 6]
- Khi `prefix_sum` = 3 tại chỉ số 3, độ dài dãy con là 4.

Đầu Ra:

- `max_length` = 4

6 Kết Luận

Phương pháp sử dụng *Prefix Sum* và *Hash Map* đã giải quyết bài toán với thời gian $O(n)$ và bộ nhớ $O(n)$. Đây là một minh chứng rõ ràng cho việc sử dụng cấu trúc dữ liệu hiệu quả để giải quyết các bài toán liên quan đến mảng.