

Sắp Xếp Mảng Chỉ Chứa 0, 1 Và 2 (Dutch National Flag Problem)

Nguyễn Hoàng Tân

Ngày 8 tháng 1 năm 2025

Tóm Tắt Nội Dung

Bài toán yêu cầu sắp xếp một mảng chỉ chứa các phần tử 0, 1 và 2 sao cho các phần tử được sắp xếp theo thứ tự tăng dần. Báo cáo này trình bày chi tiết giải pháp sử dụng thuật toán tối ưu **Dutch National Flag Algorithm**, đảm bảo thời gian chạy $O(n)$.

1 Giới Thiệu

1.1 Mô Tả Bài Toán

Cho một mảng `arr` chỉ chứa các phần tử 0, 1, và 2, yêu cầu sắp xếp mảng theo thứ tự tăng dần mà không sử dụng hàm sắp xếp tích hợp.

1.2 Lịch Sử và Đặc Điểm Chính

Lịch sử: Bài toán này được biết đến như *Dutch National Flag Problem*, được đề xuất bởi Edsger Dijkstra. Thuật toán giải quyết bài toán dựa trên việc phân vùng ba phần (three-way partitioning).

Đặc điểm chính:

- Không sử dụng bộ nhớ phụ: Chỉ thao tác trực tiếp trên mảng đầu vào.
- Thời gian thực thi tối ưu $O(n)$: Duyệt mảng một lần duy nhất.
- Giải pháp đơn giản, dễ triển khai.

2 Động Lực

2.1 Phát Biểu Bài Toán

Sắp xếp mảng `arr[0...n-1]` sao cho tất cả các phần tử 0 nằm trước, tiếp theo là 1 và cuối cùng là 2. Ví dụ:

- **Đầu vào:** `arr = [0, 1, 2, 1, 0, 2]`
- **Đầu ra:** `arr = [0, 0, 1, 1, 2, 2]`

2.2 Ứng Dụng Thực Tế

- **Phân loại dữ liệu:** Sắp xếp dữ liệu theo ba trạng thái hoặc nhóm khác nhau.
- **Tối ưu hóa bộ nhớ:** Thuật toán này không sử dụng thêm bộ nhớ phụ.
- **Hệ thống nhúng:** Áp dụng trong các bài toán xử lý dữ liệu hạn chế tài nguyên.

3 Phân Tích Giải Pháp

3.1 Ý Tưởng Thuật Toán

- Sử dụng ba con trỏ: `low`, `mid` và `high` để phân vùng mảng thành ba phần:
 - `arr[0...low-1]`: Chỉ chứa 0.
 - `arr[low...mid-1]`: Chỉ chứa 1.
 - `arr[mid...high]`: Chưa được phân loại.
- Khi `mid` vượt qua `high`, mảng sẽ được sắp xếp hoàn chỉnh.

3.2 Mã Giải

```
1 function sortArray(arr, n):  
2     low = 0  
3     mid = 0  
4     high = n - 1
```

```

5
6     while mid <= high:
7         if arr[mid] == 0:
8             swap(arr[low], arr[mid])
9             low += 1
10            mid += 1
11        elif arr[mid] == 1:
12            mid += 1
13        else:
14            swap(arr[mid], arr[high])
15            high -= 1

```

4 Ví Dụ Minh Họa

Đầu Vào:

- arr = [2, 0, 2, 1, 1, 0]

Quá Trình Thực Hiện:

- Ban đầu: low = 0, mid = 0, high = 5
- Lặp 1: Đổi arr[mid] với arr[high], arr = [0, 0, 2, 1, 1, 2]
- Lặp 2: Đổi arr[mid] với arr[low], arr = [0, 0, 2, 1, 1, 2]
- Tiếp tục cho đến khi mảng được sắp xếp.

Đầu Ra:

- arr = [0, 0, 1, 1, 2, 2]

5 Cài Đặt Bằng C++

```

1  #include <iostream>
2  #define MAX_SIZE 1000
3
4  void printArray(int arr[], int n) {
5      for (int i = 0; i < n; i++) {
6          std::cout << arr[i] << " ";
7      }
8      std::cout << std::endl;

```

```

9  }
10
11 void sortArray(int arr[], int n) {
12     int low = 0, mid = 0;
13     int high = n - 1;
14
15     while (mid <= high) {
16         if (arr[mid] == 0) {
17             int temp = arr[low];
18             arr[low] = arr[mid];
19             arr[mid] = temp;
20             low++;
21             mid++;
22         } else if (arr[mid] == 1) {
23             mid++;
24         } else {
25             int temp = arr[mid];
26             arr[mid] = arr[high];
27             arr[high] = temp;
28             high--;
29         }
30     }
31 }
32
33 int main() {
34     int n;
35     int arr[MAX_SIZE];
36
37     std::cout << "Nhap so phan tu cua mang: ";
38     std::cin >> n;
39
40     std::cout << "Nhap cac phan tu cua mang (chi gom 0, 1, 2): ";
41     for (int i = 0; i < n; i++) {
42         std::cin >> arr[i];
43     }
44
45     std::cout << "Mang truoc khi sap xep: ";
46     printArray(arr, n);
47
48     sortArray(arr, n);
49
50     std::cout << "Mang sau khi sap xep: ";

```

```
51     printArray(arr, n);  
52  
53     return 0;  
54 }
```

6 Kết Luận

Thuật toán Dutch National Flag giúp sắp xếp mảng chứa ba loại phần tử một cách tối ưu cả về thời gian và bộ nhớ. Đây là bài toán kinh điển minh họa cho phương pháp phân vùng ba phần (three-way partitioning), áp dụng rộng rãi trong xử lý dữ liệu.