

Bài 9.8: Thuật toán sắp xếp đếm

- ✓ Tổng quan về thuật toán
- ✓ Thuật toán sắp xếp đếm
- ✓ Ví dụ minh họa
- ✓ Bài tập thực hành

Tổng quan

- Counting sort-sắp xếp đếm, là một thuật toán dùng để sắp xếp tập hợp các **số nguyên dương nhỏ**.
- Thuật toán sử dụng giá trị của các phần tử mảng làm khóa.
- Thuật toán hoạt động bằng cách đếm số lần xuất hiện của các khóa và áp dụng tổng tiền tố trên các số đếm đó để xác định vị trí của từng giá trị khóa trong mảng kết quả.
- Thuật toán này không sử dụng việc so sánh các phần tử với nhau để xác định vị trí phần tử.
- Thay vào đó nó sử dụng các khóa làm chỉ số của mảng để sắp xếp.
- Thuật toán này thường được sử dụng làm thành phần của thuật toán radix sort và một số thuật toán sắp xếp khác. Những thuật toán đó xử lý các khóa có giá trị lớn hiệu quả hơn counting sort.

Tổng quan

- Thuật toán chỉ phù hợp với các trường hợp mà sự thay đổi của các giá trị khóa không chênh lệch quá lớn so với số lượng phần tử mảng.
- Độ phức tạp của thuật toán này là $O(n+k)$ với n là số phần tử của mảng và k là giá trị lớn nhất trong mảng.

Thuật toán sắp xếp đếm

➤ Mã giả của thuật toán counting sort:

```
// thuật toán sắp xếp đếm
// input: mảng đầu vào
// n: kích thước mảng input
function countingSort(input[], n):
    k = max(input) // tìm phần tử có giá trị max trong mảng
    count[] = new int[k + 1] // tạo mảng count có k + 1 phần tử
    output[] = new int[n] // mảng lưu kết quả sắp xếp
    for(i từ 0 tới n - 1): // đếm số lần xuất hiện của phần tử
        j = input[i]; // tại vị trí i trong mảng input
        count[j] += 1; // tăng biến đếm tại vị trí j lên 1 đơn vị
    for(i từ 1 tới k): // tìm tổng tiền tố cho phần tử tại vị trí i
        count[i] += count[i - 1]
    for(i từ n - 1 tới 0): // đưa các phần tử vào đúng vị trí của nó
        j = input[i] // lấy phần tử tại vị trí i mảng input
        count[j] = count[j] - 1 // giảm biến đếm của nó đi 1
        output[count[j]] = input[i] // gán phần tử vào đúng vị trí
    for(i từ 0 tới n - 1): // chép các phần tử vào mảng gốc
        input[i] = output[i]
    delete[] count // thu hồi bộ nhớ cấp phát cho mảng count
    delete[] output // thu hồi bộ nhớ cấp phát cho mảng output
```

Thuật toán sắp xếp đếm

➤ Mã thật của thuật toán counting sort:

```
// thuật toán sắp xếp đếm
void countingSort(int* input, int size) {
    int k = findMax(input, size); // tìm giá trị lớn nhất trong mảng input
    int* output = new int[size]; // cấp phát mảng output
    int* count = new int[k + 1](); // cấp phát, khởi tạo giá trị mặc định cho mảng
    for (int i = 0; i < size; i++) // tìm số lần xuất hiện của từng phần tử
    {
        int j = input[i];
        count[j]++;
    }
    for (int i = 1; i <= k; i++) // cộng dồn tổng tiền tố
    {
        count[i] += count[i - 1];
    }
    for (int i = size - 1; i >= 0; i--) // đưa các phần tử vào đúng vị trí của nó
    {
        int j = input[i];
        count[j] = count[j] - 1;
        output[count[j]] = input[i];
    }
    for (int i = 0; i < size; i++) // chép kết quả sang mảng gốc input
    {
        input[i] = output[i];
    }
    delete[] count; // thu hồi bộ nhớ cấp phát động
    delete[] output; // thu hồi bộ nhớ cấp phát động
}
```

Nội dung tiếp theo

Thuật toán sắp xếp radix sort