

Quy tắc tính độ phức tạp trên mã nguồn

Các quy tắc nên tảng

Quy tắc bỏ hằng số

Big O chỉ quan tâm đến tốc độ tăng trưởng khi n rất lớn. Do đó, các hệ số hằng số không có ý nghĩa

Ví dụ

- $O(2n) \rightarrow O(n)$
- $O(500) \rightarrow O(1)$
- $O(n/3) \rightarrow O(n)$

Quy tắc lấy thành phần worst case

Khi một thuật toán có nhiều thành phần, độ phức tạp tổng thể được quyết định bởi thành phần chậm nhất

Ví dụ

- $O(n^2 + n) \rightarrow O(n^2)$
- $O(2n + n^2) \rightarrow O(n^2)$
- $O(n \log n + n \log n) \rightarrow O(n \log n)$

Các thao tác cơ bản là $O(1)$

Một thao tác cơ bản không phụ thuộc vào kích thước đầu vào được coi là có độ phức tạp hằng số $O(1)$

Phân tích mã nguồn

Các câu lệnh thường gặp

```
public void DoSomething(int[] numbers)
{
    // Phân tích từng dòng:
```

```

    Console.WriteLine("B?t ??u...");           // O(1)

    int sum = numbers[0] + numbers[1];         // O(1)

    for (int i = 0; i < numbers.Length; i++)   // O(n)
    {
        Console.WriteLine(numbers[i]);
    }

    Console.WriteLine("K?t thúc.");           // O(1)
}

```

Tổng độ phức tạp = $O(1) + O(1) + O(n) + O(1) = O(n+3)$

Áp dụng quy tắc loại bỏ hằng số và lấy phần vượt trội ta có độ phức tạp là $O(n)$

Vòng l?p

```

// n = items.Count
public void LoopExample(List<string> items)
{
    // Vòng l?p này ch?y n l?n
    foreach (var item in items)
    {
        // Kh?i l?nh bên trong ch? có các thao tác O(1)
        Console.WriteLine("Item: " + item); // O(1)
        int length = item.Length;           // O(1)
    }
}

```

Mỗi vòng lặp tốn $O(1) + O(1) = O(1)$

Vòng lặp này lặp lại n lần → độ phức tạp là $O(n)$

Vòng l?p l?ng nhau

```

// n = numbers.Count
public void NestedLoopExample(List<int> numbers)
{
    // Vòng l?p ngoài ch?y n l?n
    foreach (var num1 in numbers)
    {
        // Vòng l?p trong c?ng ch?y n l?n v?i m?i l?n l?p c?a vòng ngoài
        foreach (var num2 in numbers)
        {
            // Thao tác bên trong là O(1)
            Console.WriteLine($"{num1}, {num2}");
        }
    }
}

```

Độ phức tạp của mỗi vòng lặp trong là $O(n)$

Vòng lặp trong được lặp lại n lần.

Độ phức tạp của thuật toán là $O(n*n) = O(n^2)$

Câu lệnh rẽ nhánh

```
public void ConditionalExample(int condition, int[] data)
{
    if (condition == 0)
    {
        // Nhánh này có ?? ph?c t?p O(1)
        Console.WriteLine("Condition is true.");
    }
    else if (condition == 1)
    {
        // Nhánh này có ?? ph?c t?p O(n)
        foreach (var item in data)
        {
            Console.WriteLine(item);
        }
    }
    else
    {
        // Nhánh này có ?? ph?c t?p là O(n2)
        foreach (var num1 in numbers)
        {
            foreach (var num2 in numbers)
            {
                Console.WriteLine($"{num1}, {num2}");
            }
        }
    }
}
```

Với câu lệnh điều kiện độ phức tạp của thuật toán là độ phức tạp của nhánh tệ nhất.

Trong trường hợp này là $O(n^2)$

Revision #1

Created 2025-07-13 14:20:20 UTC by ThanhDV

Updated 2025-07-13 15:05:46 UTC by ThanhDV