

BT01: PHÉP GÁN VÀ PHÉP SO SÁNH

Nhập môn Phân tích độ phức tạp thuật toán

CNTN2013

Nguyễn Đức Thân Trương Toàn Thịnh

Nguyễn Sơn Hoàng Quốc Nguyễn Vinh Tiệp

Ngày 7 tháng 10 năm 2015

1 Đếm số phép gán và so sánh

Trong phần này ta cần đếm số phép gán và phép so sánh được thực hiện trong một thuật toán được cài đặt bởi một đoạn mã nguồn, từ đó có thể kiểm chứng được độ phức tạp của thuật toán đối với cài đặt đó.

Ta xét ví dụ đơn giản - một thuật toán tính tổng bình phương từ 1 đến n :

```
1 // Thuật toán tính tổng bình phương
2 int SquareSum(int n)
3 {
4     int i = 1;
5     int sum = 0;
6
7     while (i <= n)
8     {
9         sum += i*i;
10        i++;
11    }
12
13    return sum;
14 }
```

Ta có thể thấy ngay độ phức tạp của thuật toán là tuyến tính, tức là $O(n)$. Cụ thể, với số vòng lặp là n , số phép gán sẽ là $2n + 2$ và số phép so sánh là $n + 1$. Ta có thể bổ sung vào mã nguồn các biến đếm để kiểm chứng điều này:

```
1 // Thuật toán tính tổng bình phương, có biến đếm
2 int SquareSum(int n, int &count_assign, int &count_compare)
3 {
4     count_assign = 0;
5     count_compare = 0;
6
7     int i = 1; ++count_assign;
8     int sum = 0; ++count_assign;
9
10    while (++count_compare && i <= n)
11    {
12        sum += i*i; ++count_assign;
13        i++; ++count_assign;
14    }
15
16    return sum;
17 }
```

Bằng việc chạy thuật toán với một vài n , ta kiểm chứng được tính toán lý thuyết ban đầu.

2 Bài tập

Trong các bài tập dưới đây, sinh viên so sánh kết quả chạy thực nghiệm với tính toán lý thuyết:

- Cài đặt bổ sung các biến đếm số phép gán và phép so sánh vào mã nguồn của các thuật toán.
- Bổ sung các giá trị trả về là giá trị của các biến đếm.
- Dựa trên công thức suy ra từ tính toán lý thuyết, viết các bộ kiểm thử hộp đen¹ tương ứng, đầu vào là các giá trị đầu vào của thuật toán gốc, đầu ra là đầu ra của thuật toán gốc và **giá trị của các biến đếm**. Yêu cầu tối thiểu **10** bộ kiểm thử cho từng thuật toán.

¹black-box test cases

- Chạy các thuật toán (đã cài đặt bổ sung) và so khớp kết quả với các bộ kiểm thử.

Ví dụ: với thuật toán tính tổng bình phương ở Phần 1, ta có thể có các bộ kiểm thử sau:

Input	Output		
5	55	12	6
10	385	22	11
20	2870	42	21
50	42925	102	51

2.1 Bài tập lý thuyết

Thực hiện các yêu cầu trên với các bài tập trên lớp lý thuyết.

2.2 Dãy số Fibonacci

Thực hiện các yêu cầu trên với 2 thuật toán sinh số Fibonacci thứ n dưới đây. Sau đó so sánh độ phức tạp của 2 thuật toán.

Dãy số **Fibonacci** được định nghĩa đệ quy, với phần tử thứ n là $F_n = F_{n-1} + F_{n-2}$, và các giá trị khởi đầu là $F_0 = 1, F_1 = 1$.

2.2.1 Thuật toán đệ quy

```

1 int fibonacci(int n)
2 {
3     if (n <= 1)      1 so sánh
4         return 1;
5     else
6         return fibonacci(n - 1) + fibonacci(n - 2);
7 }
```

2.2.2 Thuật toán không đệ quy

```
1 int fibonacci(int n)
2 {
3     if(n <= 1)           1 so sánh
4         return 1;
5
6     int last = 1;         1 gán
7     int nextToLast = 1;   1 gán
8     int answer = 1;       1 gán
9
10    for(int i = 2; i <= n; i++)    n gán    n so sánh
11    {
12        answer = last + nextToLast; n-1 gán
13        nextToLast = last;         n-1 gán
14        last = answer;             n-1 gán
15    }
16
17    return answer;
18 }
```

Nộp bài

- **Nội dung nộp:** Xem trên moodle nội dung yêu cầu nộp ở phần **Báo cáo bài Ex1_Counting**.
- **Deadline:** Xem trên moodle.