

# BT03: ĐẾM VỚI YẾU TỐ NGẪU NHIÊN

## INSERTION SORT

Nhập môn Phân tích độ phức tạp thuật toán - CNTN2013

Nguyễn Đức Thân      Trương Toàn Thịnh  
Nguyễn Sơn Hoàng Quốc      Nguyễn Vinh Tiệp

## 1 Yếu tố ngẫu nhiên

### 1.1 Thành phần tất định

Trong bài tập trước, sinh viên đã thực nghiệm với thuật toán tính tổng dưới đây.

```
1 int SomeSum(int n)
2 {
3     int sum = 0, i = 1;
4     int j;
5     while (i <= n)
6     {
7         j = n - i;
8         while (j <= i*i)
9         {
10            sum = sum + i*j;
11            j = j + 1;
12        }
13        i = i + 1;
14    }
15    return sum;
16 }
```

Mặc dù các kết quả đo có vẻ ngẫu nhiên, thực chất đây là thuật toán có độ phức tạp tất định. Cụ thể, qua tính toán lý thuyết, hoàn toàn có thể xác

định được chính xác số phép gán  $G(n)$  và số phép so sánh  $S(n)$  của thuật toán.

$$\begin{aligned}
 G(n) &= 4 + 3n - n^2 + \frac{n(n+1)(2n+1)}{3} \\
 &\quad - \frac{(i_0-1)i_0(2i_0-1)}{3} - i_0^2 - i_0 + 2ni_0 \\
 S(n) &= 2 + 2n - n^2 + \frac{n(n+1)(2n+1)}{6} + \frac{n(n+1)}{2} \\
 &\quad - \frac{(i_0-1)i_0(2i_0-1)}{6} - \frac{(i_0-1)i_0}{2} + (n-1)i_0
 \end{aligned}$$

Trong đó

$$i_0 = \left\lceil \frac{-1 + \sqrt{1 + 4n}}{2} \right\rceil$$

*Sinh viên có thể so sánh công thức lý thuyết với thực nghiệm. Sau khi được trang bị các kiến thức lý thuyết cần thiết, sinh viên có thể tự suy ra công thức trên.*

## 1.2 Thành phần ngẫu nhiên

Bây giờ ta xét một thuật toán với độ phức tạp có thành phần ngẫu nhiên - nghĩa là độ phức tạp của thuật toán là một biến ngẫu nhiên phụ thuộc vào dữ liệu đầu vào. Thuật toán Insertion Sort:

```

1 void insertion_sort(vector<int>& arr)
2 {
3     int i, j, key, numLength;
4
5     numLength = arr.size();
6
7     for(j = 1; j < numLength; j = j + 1)
8     {
9         key = arr[j];
10
11         for(i = j - 1; (i >= 0) && (arr[i] < key); i = i - 1)
12         {
13             arr[i+1] = arr[i];
14         }
15

```

```
16 |     arr[i+1] = key;  
17 | }  
18 | }
```

Thành phần ngẫu nhiên của thuật toán phát sinh ở bước so sánh `arr[i] < key` - số vòng lặp trong cùng được xác định tùy vào dữ liệu đầu vào có thứ tự sắp xếp như thế nào.

Khi thuật toán có thành phần ngẫu nhiên, ta chú ý đến các trường hợp độ phức tạp tốt nhất, xấu nhất và trung bình, tương ứng với các trường hợp của dữ liệu khiến cho các biến đếm đạt giá trị nhỏ nhất, lớn nhất và trung bình.

Với thuật toán Insertion Sort, độ phức tạp tốt nhất, xấu nhất và trung bình theo tính toán lý thuyết lần lượt là  $O(n)$ ,  $O(n^2)$  và  $O(n^2)$  với  $n$  là kích thước của mảng sắp xếp.

## 2 Bài tập

### 2.1 Yêu cầu

- Xác định xem thuật toán Insertion Sort có độ phức tạp tốt nhất, xấu nhất và trung bình khi nào.
- Viết bổ sung mã nguồn đếm số phép gán và phép so sánh cho thuật toán.
- Viết mã nguồn sinh dữ liệu *ngẫu nhiên* cho các trường hợp tốt nhất, xấu nhất và trung bình của thuật toán.
- Với mỗi trường hợp tốt nhất, xấu nhất và trung bình, phát sinh 1000 bộ dữ liệu cho từng kích thước bài toán  $n = 100, 200, \dots, 900, 1000$ ; tổng cộng  $3 \times 1000 \times 10 = \mathbf{30000}$  bộ dữ liệu.
- Chạy thuật toán và thống kê kết quả đếm theo từng trường hợp và từng kích thước  $n$ .
- Đối chiếu kết quả thực nghiệm với lý thuyết.

## 2.2 Tổ chức mã nguồn

Sinh viên được cung cấp sẵn khung mã nguồn bằng ngôn ngữ C++ (nếu sinh viên sử dụng ngôn ngữ khác, phải tổ chức mã nguồn theo cấu trúc tương tự như mã nguồn được cung cấp):

- `main.cpp`: mã nguồn chương trình chính, gọi các hàm sinh dữ liệu, đọc dữ liệu, chạy thuật toán và lưu kết quả đếm.
- `generators.h` và `generators.cpp`: khai báo và cài đặt các hàm sinh dữ liệu.
- `file_readers.h` và `file_readers.cpp`: khai báo và cài đặt các hàm đọc dữ liệu.
- `algorithms.h` và `algorithms.cpp`: khai báo và cài đặt các thuật toán.

## 2.3 Tài liệu nộp

- Các tập tin mã nguồn.
- Tập tin Excel chứa kết quả đếm và biểu đồ, theo các trường hợp tốt nhất, xấu nhất và trung bình (tham khảo tập tin Excel `insertion_sort_count.xlsx`).
- Tập tin Word mô tả dữ liệu trong các trường hợp độ phức tạp tốt nhất, xấu nhất và trung bình; nhận xét, so sánh đánh giá lý thuyết và thực nghiệm.