Thực hiện so sánh thuật toán Fibonacci đệ quy và không đệ quy

Sinh viên thực hiện: Nguyễn Trần Hậu – 1612180

1/ Mã nguồn

Mã nguồn thuật toán tính số Fibonacci **đệ quy**

```
int fibonaci_dquy(int n, int &count_assign, int &count_compare){
    // Phần 1
    // 1 lần so sánh
    if(++count_compare && n <= 1){
        // 1 phép gán
        ++count_assign;
        return 1;
    }

    // Phần 2
    // 1 lần gán
    ++count_assign;
    return fibonaci_dquy(n - 1, count_assign, count_compare) +
        fibonaci_dquy(n - 2, count_assign, count_compare);
}
```

Mã nguồn thuật toán tính số Fibonacci **không đệ quy**

```
int fibonaci_kdquy(int n, int &count_assign, int &count_compare){
    // Phần 1
    if (++count_compare && n <= 1){
        ++count_assign;
        return 1;
    }

    // Phần 2
    // 3 lần gán
    count_assign += 3;
    int last = 1;
    int nextToLast = 1;
    int answer = 1;

    // Phần 3
    // i gán từ 2 đến n + 1 -> n lần
    // trong vòng lặp có 3 lần gán, mà vòng lặp chạy với i từ 2 đến n -> 3(n - 1) lần
    // i so sánh từ 2 đến n + 1 -> n lần
    ++count_assign;
    for(int i = 2; ++count_compare && i <= n; ++count_assign && ++i){
```

```
        count_assign += 3;
        answer = last + nextToLast;
        nextToLast = last;
        last = answer;
    }

    // Phần 4
    // 1 lần gán
    ++count_assign;
    return answer;
}
```

2/ Kết quả thực thi

Kết quả thuật toán tính số Fibonacci **đệ quy**

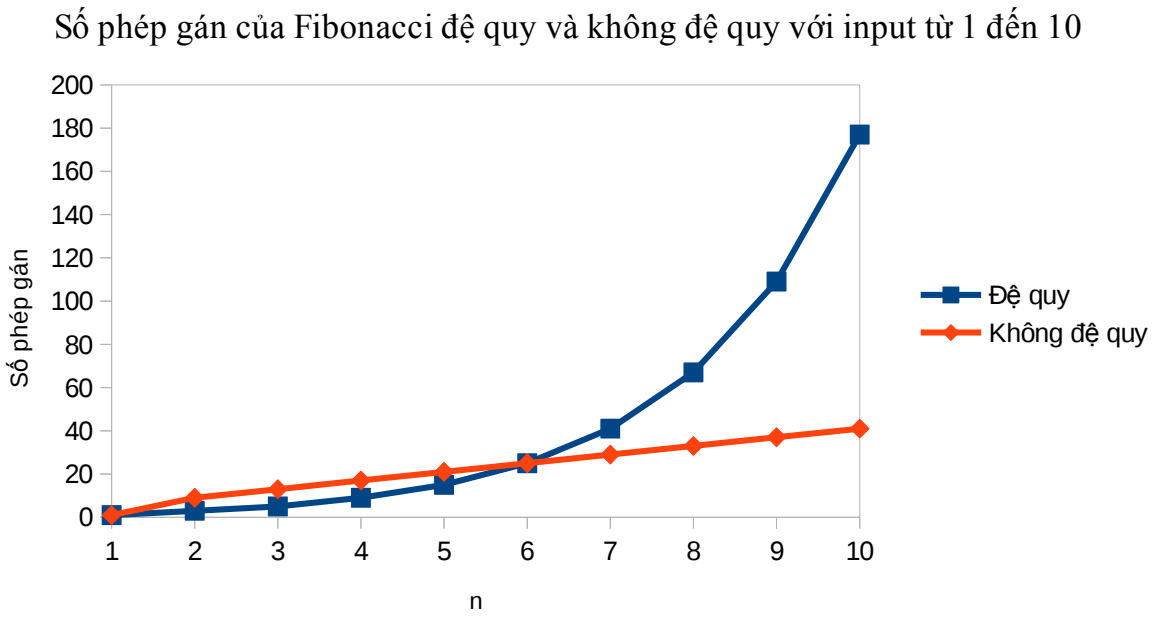
Input	Output		
	Kết quả	Số phép gán	Số phép so sánh
1	1	1	1
2	2	3	3
3	3	5	5
4	5	9	9
5	8	15	15
6	13	25	25
7	21	41	41
8	34	67	67
9	55	109	109
10	89	177	177
11	144	287	287
12	233	465	465
13	377	753	753
14	610	1219	1219
15	987	1973	1973

Kết quả thuật toán tính số Fibonacci **không đệ quy**

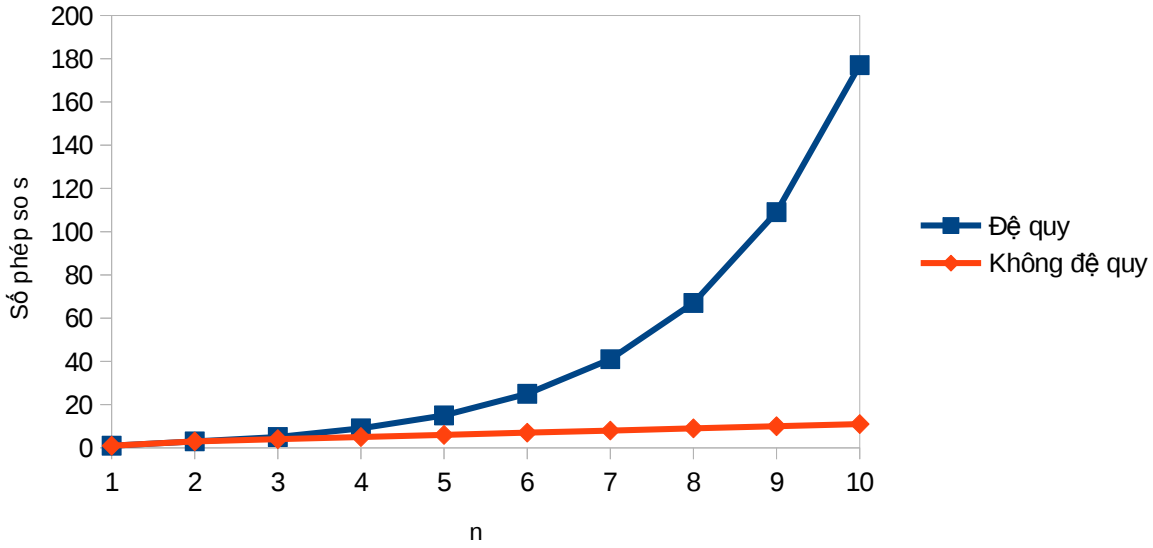
Input	Output		
	Kết quả	Số phép gán	Số phép so sánh
1	1	1	1
2	2	9	3
3	3	13	4
4	5	17	5

5	8	21	6
6	13	25	7
7	21	29	8
8	34	33	9
9	55	37	10
10	89	41	11
11	144	45	12
12	233	49	13
13	377	53	14
14	610	57	15
15	987	61	16

3/ Đồ thị thể hiện



Số phép so sánh của Fibonacci đệ quy và không đệ quy với input từ 1 đến 10



Đối với phép gán, n < 6 thì số phép gán của thuật toán đệ quy ít hơn thuật toán không đệ quy. Từ n > 6 trở đi, số phép gán của thuật toán đệ quy lớn hơn thuật toán không đệ quy và khoảng cách tăng nhanh.

Đối với phép so sánh, lúc nào số phép so sánh của thuật toán không đệ quy cũng ít hơn thuật toán đệ quy, và khoảng cách này càng lúc càng lớn dần

4/ Công thức

Thuật toán Fibonacci không đệ quy:

Số phép gán:

- Phần 1: với n <= 1 thì số phép gán là 1, n > 1 thì số phép gán là 0

- Phần 2: 3 phép gán

- Phần 3: i gán n lần, lặp n – 1 lần và trong vòng lặp gán 3 lần => n + 3(n – 1) phép gán

- Phần 4: 1 phép gán

=> Số phép gán với n <= 1 là 1, với n > 1 là 3 + n + 3(n – 1) + 1 = 4n + 1

Số phép so sánh:

- Phần 1: 1 phép so sánh

- Phần 2: 0 phép so sánh

- Phần 3: với n > 1, i so sánh từ 2 đến n + 1 => n phép so sánh

- Phần 4: 0 phép so sánh

=> Số phép so sánh với n <= 1 là 0, với n > 1 là 1 + n

Đối với thuật toán Fibonacci đệ quy:

Số phép gán:

- Phần 1: với n <= 1 có 1 phép gán, với n > 1 không có phép gán

- Phần 2: với n > 1, số phép gán = 1 + số phép gán của (n – 1) và số phép gán của (n - 2)

=> Số phép gán là hàm $f_n = f_{n-1} + f_{n-2} + 1$ với $f_0 = 1$ và $f_1 = 1$

Số phép so sánh:

- Phần 1: có 1 phép so sánh

- Phần 2: với n > 1, số phép so sánh = số phép so sánh của (n – 1) và số phép so sánh của (n – 2)

=> Số phép so sánh là hàm $f_n = f_{n-1} + f_{n-2} + 1$ với $f_0 = 1$ và $f_1 = 1$

Hàm này giải ra là $f_n =$

$$(1 - \frac{1}{\sqrt{5}})(\frac{1-\sqrt{5}}{2})^n + (1 + \frac{1}{\sqrt{5}})(\frac{1+\sqrt{5}}{2})^n - 1$$

5/ Độ phức tạp thuật toán

Với thuật toán Fibonacci không đệ quy:

Số phép gán với n > 1 là 4n + 1

Số phép so sánh với n > 1 là n + 1

=> Thuật toán có độ phức tạp Θ(n)

Với thuật toán Fibonacci đệ quy

Số phép gán và số phép so sánh đều là hàm f_n như được tính ở phần 4

=> Thuật toán có độ phức tạp O(2^n)