Nhập môn phân tích độ phức tạp thuật toán-21TN

TRẦN MINH HOÀNG-21120075

BÀI TẬP LÝ THUYẾT LẦN 3

Câu 1: Ký hiệu $d(x)$ là số lượng ước dương của x,xét đẳng thức:

$$d(1)+d(2)+\cdots+d(n)=\lfloor\tfrac{n}{1}\rfloor+\lfloor\tfrac{n}{2}\rfloor+\cdots+\lfloor\tfrac{n}{n}\rfloor \text{ với } n\in\mathbb{Z}^+.(*)$$

Chẳng hạn với n=4 thì ta có:

$d(1)+d(2)+d(3)+d(4)=1+2+2+3=8$ và $\lfloor\frac{4}{1}\rfloor+\lfloor\frac{4}{2}\rfloor+\lfloor\frac{4}{3}\rfloor+\lfloor\frac{4}{4}\rfloor=8$.

a) Dựa vào hằng đẳng thức (*) ở trên,hãy ước lượng big-Θ theo n cho tổng $d(1)+d(2)+\cdots+d(n)$.Ngoài ra ,hãy chứng minh đẳng thức (*) đúng với mọi n nguyên dương.

Ước lượng độ lớn theo Θ:

$$T(n)=d(1)+d(2)+\ldots+d(n)=\left\lfloor\frac{n}{1}\right\rfloor+\left\lfloor\frac{n}{2}\right\rfloor+\ldots+\left\lfloor\frac{n}{n}\right\rfloor$$

- $T(n)=\lfloor\frac{n}{1}\rfloor+\lfloor\frac{n}{2}\rfloor+\ldots+\lfloor\frac{n}{n}\rfloor\le\frac{n}{1}+\frac{n}{2}+\ldots+\frac{n}{n}=n\left(\frac{1}{1}+\frac{1}{2}+\ldots+\frac{1}{n}\right)$

  Mà $\frac{1}{1}+\frac{1}{2}+\ldots+\frac{1}{n}\in\Theta(\log n)$
  Vậy $n\left(\frac{1}{1}+\frac{1}{2}+\ldots+\frac{1}{n}\right)\in\Theta(n\log n)$.
  Suy ra $T(n)\in O(n\log n)$ (1)
- $T(n)=\lfloor\frac{n}{1}\rfloor+\lfloor\frac{n}{2}\rfloor+\ldots+\lfloor\frac{n}{n}\rfloor\ge\lfloor\frac{n}{1}\rfloor+\lfloor\frac{n}{2}\rfloor+\ldots+\left\lfloor\frac{n}{\lfloor n/2\rfloor}\right\rfloor\ge\left(\frac{n}{1}+\frac{n}{2}+\ldots+\frac{n}{\lfloor n/2\rfloor}\right)-\lceil\frac{n}{2}\rceil$
  Mà $\frac{1}{1}+\frac{1}{2}+\ldots+\frac{1}{\lfloor n/2\rfloor}\in\Theta(\log\frac{n}{2})=\Theta(\log n-\log 2)=\Theta(\log n)$
  Vậy $n\left(\frac{1}{1}+\frac{1}{2}+\ldots+\frac{1}{\lfloor n/2\rfloor}\right)\in\Theta(n\log n)$
  Suy ra $n\left(\frac{1}{1}+\frac{1}{2}+\ldots+\frac{1}{\lfloor n/2\rfloor}\right)-\lceil\frac{n}{2}\rceil\in\Theta(n\log n)$
  Hay ra $T(n)\in\Omega(n\log n)$ (2)

Từ (1) và (2) suy ra: $T(n)\in\Theta(n\log n)$

Chứng minh đẳng thức đúng với mọi n nguyên dương:

Ta sẽ chứng minh bằng phương pháp quy nạp:

- Với n=1:
  Ta có $d(1)=1=\lfloor\frac{1}{1}\rfloor$
- Giả sử (*) đúng đến $n=k(\forall k\ge 1):d(1)+d(2)+\cdots+d(k)=\lfloor\frac{k}{1}\rfloor+\lfloor\frac{k}{2}\rfloor+\cdots+\lfloor\frac{k}{k}\rfloor$
- Ta sẽ chứng minh đẳng thức trên đúng với $n=k+1$: Với $a\in\mathbb{Z}^+$ và $a\le k$:
  - $k+1\vdots a\Rightarrow\lfloor\frac{k+1}{a}\rfloor=\lfloor\frac{k}{a}\rfloor+1$
  - $k+1\not\vdots a\Rightarrow\lfloor\frac{k+1}{a}\rfloor=\lfloor\frac{k}{a}\rfloor$

Đặt $A = \left(\left\lfloor \frac{k+1}{1} \right\rfloor + \left\lfloor \frac{k+1}{2} \right\rfloor + \ldots + \left\lfloor \frac{k+1}{k+1} \right\rfloor\right)$

Đặt $B = \left(\left\lfloor \frac{k}{1} \right\rfloor + \left\lfloor \frac{k}{2} \right\rfloor + \ldots + \left\lfloor \frac{k}{k} \right\rfloor\right)$

Với $d(k+1)$ là số ước dương của k+1 thì:

Với mỗi a thỏa $(k+1) \vdots a$ thì độ lệch của A và B sẽ lệch đi 1 tương ứng với số ước của $(k+1) \vdots a$

$$\Leftrightarrow \left(\left\lfloor \frac{k+1}{1} \right\rfloor + \left\lfloor \frac{k+1}{2} \right\rfloor + \ldots + \left\lfloor \frac{k+1}{k+1} \right\rfloor\right) - [d(1) + d(2) + \ldots + d(k)] = d(k+1)$$

$$\Leftrightarrow \left(\left\lfloor \frac{k+1}{1} \right\rfloor + \left\lfloor \frac{k+1}{2} \right\rfloor + \ldots + \left\lfloor \frac{k+1}{k+1} \right\rfloor\right) = d(1) + d(2) + \ldots + d(k) + d(k+1)$$

Đẳng thức trên đúng với n=k+1

Vậy theo nguyên lý quy nạp ta có điều phải chứng minh.

b) Từ tính chất được nêu ở trên, hãy ước lượng độ phức tạp của đoạn code sau (viết bằng C++) dựa trên việc đếm số phép S:

```
int happy(int n){
    int i = 1, res = 0;
    while(i <= n){
        int k = 1, j = 1;
        while(k <= 2*i){
            res = res - k*i;
            k = k + 1;
        }
        while(j <= i){
            if(i % j == 0){
                int t = 1;
                while(t <= n){
                    res = res + j*t;
                    t = t + 1;
                }
            }
            j = j + 1;
        }
        i = i + 1;
    }
    return res;
}
```

Độ phức tạp của đoạn code:

- Tổng số phép S : $S = S(Q) = 2 + \sum_{i \le n} S(P_i)$
- $S(P_i) = S(A_{k(i)}) + S(B_{j(i)}) + 1$
- Lại có $S(A_{k(i)}) = 2 + 2i + 2i = 4i + 2$
- $S(B_{j(i)}) = \sum_{j \le i}(S(C_{j(i)}) + 1) = \sum_{j=1}^{i} S(C_{j(i)}) + \sum_{j=1}^{i} 1$
- Ta lại có $S(C_{j(i)}) = \begin{cases} 0, & i \not\vdots j \\ 2n+1, & i \vdots j \end{cases}$
- $S(B_{j(i)}) = \sum_{j;i \not\vdots j} 0 + \sum_{j;i \vdots j} (2n+1) + \sum_{j=1}^{i} 1 = 0 + (2n+1) \cdot d(i) + i$
- $S(P_i) = 3 + 4i + (2n+1) \cdot d(i) + i = 3 + 5i + (2n+1) \cdot d(i)$

Tính tổng số phép gán:

$$S = 2 + \sum_{i=1}^{n} S(P_i) = 2 + \sum_{i=1}^{n}(3 + 5i + (2n+1)\cdot d(i))$$
$$= 2 + \sum_{i=1}^{n} 3 + \sum_{i=1}^{n}(5i) + \sum_{i=1}^{n}((2n+1)\cdot d(i))$$
$$= 2 + 3n + 5\cdot\frac{n(n+1)}{2} + (2n+1)\cdot\sum_{i=1}^{n}(d(i))$$

Mà ta có: $\sum_{i=1}^{n} d(i) = d(1) + d(2) + \cdots + d(n) \in \Theta(n\log n)$

$\Rightarrow (2n+1)\cdot\sum_{i=1}^{n} d(i) \in \Theta(n^2\log n)$

$$\Rightarrow S = 2 + 3n + 5\cdot\frac{n(n+1)}{2} + (2n+1)\cdot\sum_{i=1}^{n}(d(i)) \in \Theta(n^2\log n)$$

Kết luận: Độ phức tạp của thuật toán $\in \Theta(n^2\log n)$.

Câu 2: Xét đoạn code sau viết bằng C++ để kiểm tra số nguyên dương $a_0, a_1, a_2, \cdots, a_{n-1}$ có thỏa mãn ràng buộc: với mọi $i = 0, 1, 2, \cdots, n-1$ thì hai số a_i và i đều có cùng tính chẵn lẻ hay không:

```
bool checkParity(int n, int a[]){
    for(int i = 0; i < n; i++){
        if((a[i] % 2) != (i % 2)) return false;
    }
    return true;
}
```

Giả sử rằng mỗi phần tử của mảng được sinh ngẫu nhiên với xác suất là $\frac{1}{2}$ cho số chẵn và $\frac{1}{2}$ cho số lẻ. Xét $f_n(x)$ là hàm sinh xác suất ứng với số lần lặp của thuật toán.

a) CHứng minh rằng $f_n(x) = \frac{x}{2} + \frac{x^2}{4} + \frac{x^3}{8} + \cdots + \frac{x^{n-2}}{2^{n-2}} + \frac{x^{n-1}}{2^{n-1}} + \frac{x^n}{2^n}$
Gọi P_k là xác suất vòng lặp dừng lại ở lần thứ k.

- $1 \le k \le n-1$:
$$P_k = (\prod_{n=1}^{k-1}\frac{1}{2}) * \frac{1}{2} = (\frac{1}{2})^k$$
- k=n:
Vòng lặp bắt buộc phải dừng: $P_k = P_n = \prod_{n=1}^{k-1}\frac{1}{2} = (\frac{1}{2})^{k-1}$

Vậy khi đó hàm sinh xác suất:

$$f_n(x) = \sum_{k=1}^{n} P_k x^k = \sum_{k=1}^{n-1} P_k x^k + p_n x^n = \sum_{k=1}^{n-1}(\frac{1}{2})^k x^k + (\frac{1}{2})^{n-1}x^n$$
$$= \sum_{k=1}^{n-1}\frac{x^k}{2^k} + \frac{x^n}{2^n - 1} = \frac{x^2}{4} + \frac{x^3}{8} + \cdots + \frac{x^{n-2}}{2^{n-2}} + \frac{x^{n-1}}{2^{n-1}} + \frac{x^n}{2^n}$$

b) Hãy tính độ phức tạp trung bình của thuật toán bằng cách xác định kỳ vọng của số lần lặp dựa theo hàm sinh trên.
Kỳ vọng của hàm sinh là số lần lặp trung bình của thuật toán hay nói cách khác kỳ vọng của hàm sinh là độ phức tạp trung bình của thuật toán:$\Leftrightarrow E_n(X) = f_n'(1)$

$$f_n'(x) = (\frac{x^2}{4} + \frac{x^3}{8} + \cdots + \frac{x^{n-2}}{2^{n-2}} + \frac{x^{n-1}}{2^{n-1}} + \frac{x^n}{2^n})'$$
$$= (\sum_{k=1}^{n-1}\frac{x^k}{2^k} + \frac{x^n}{2^n - 1})'$$

$$= \sum_{k=1}^{n-1} \frac{k}{2^k} x^{k-1} + \frac{nx^{n-1}}{2^n - 1}$$

Vậy $E_n(X) = f'_n(1) = \sum_{k=1}^{n-1} \frac{k}{2^k} + \frac{n}{2^n - 1} = \sum_{k=1}^{n-1} \frac{k}{2^k} + \frac{n}{2^{n-1}}$

Ta sẽ tính toán $S = \sum_{k=1}^{n-1} \frac{k}{2^k}$

$$2S = \sum_{k=1}^{n-1} \frac{2k}{2^k} = \sum_{k=1}^{n-1} \frac{k}{2^{k-1}} = 1 + \sum_{k=2}^{n-1} \frac{k}{2^{k-1}}$$

$$\Leftrightarrow 2S = 1 + \sum_{j=1}^{n-2} \frac{j+1}{2^j} = 1 + \sum_{j=1}^{n-2} \frac{j}{2^j} + \sum_{j=1}^{n-2} \frac{1}{2^j}$$

Ta có:

$$\sum_{j=1}^{n-2} \frac{1}{2^j} = \frac{1}{2} \left(\frac{1 - (\frac{1}{2})^{n-2}}{1 - \frac{1}{2}} \right) = 1 - \frac{1}{2^{n-2}} (*)$$

Từ (*) ta có:

$$2S = 1 + \sum_{j=1}^{n-2} \frac{j}{2^j} + 1 - \frac{1}{2^{n-2}} = 2 + \sum_{j=1}^{n-2} \frac{j}{2^j} - \frac{1}{2^{n-2}}$$

$$\Leftrightarrow 2S = 2 + S - \frac{n-1}{2^{n-1}} \Leftrightarrow S = 2 - \frac{n-1}{2^{n-1}}$$

$\Rightarrow E(X) = 2 - \frac{n+1}{2^{n-1}} + \frac{n}{2^{n-1}} = 2 - \frac{1}{2^{n-1}} \in \theta(2) = \theta(1)$
Vậy độ phức tạp trung bình của thuật toán là $\theta(1)$

Câu 3: Trong bài này, ta sẽ đánh giá độ phức tạp thuật toán bằng thực nghiệm.

a) Cài đặt bổ sung các biến đếm số phép S count_assign và số phép so sánh count_compare vào mã nguồn của các thuật toán sau đây,thực hiện đếm rồi lập bảng thống kê hoặc vẽ đồ thị mô tả sự thay đổi của các giá trị đó theo n (SV có thể cài đặt lại bằng Python cũng được,bài làm trên notebook hoặc chụp code chèn vào file báo cáo đều được).

- Thuật toán tính số Fibonaci dùng đệ quy:

```
int fibonacci(int n)
{
    if (n <= 1)
        return 1;
    else
        return fibonacci(n - 1) + fibonacci(n - 2);
}
```

- Thuật toán tính số Fibonaci không đệ quy:

```
int fibonacci(int n)
{
    if(n <= 1)
        return 1;

    int last = 1;
    int nextToLast = 1;
    int answer = 1;

    for(int i = 2; i <= n; i++)
    {
        answer = last + nextToLast;
        nextToLast = last;
        last = answer;
    }

    return answer;
}
```

Thuật toán tính số Fibonaci dùng đệ quy với thêm biến đếm

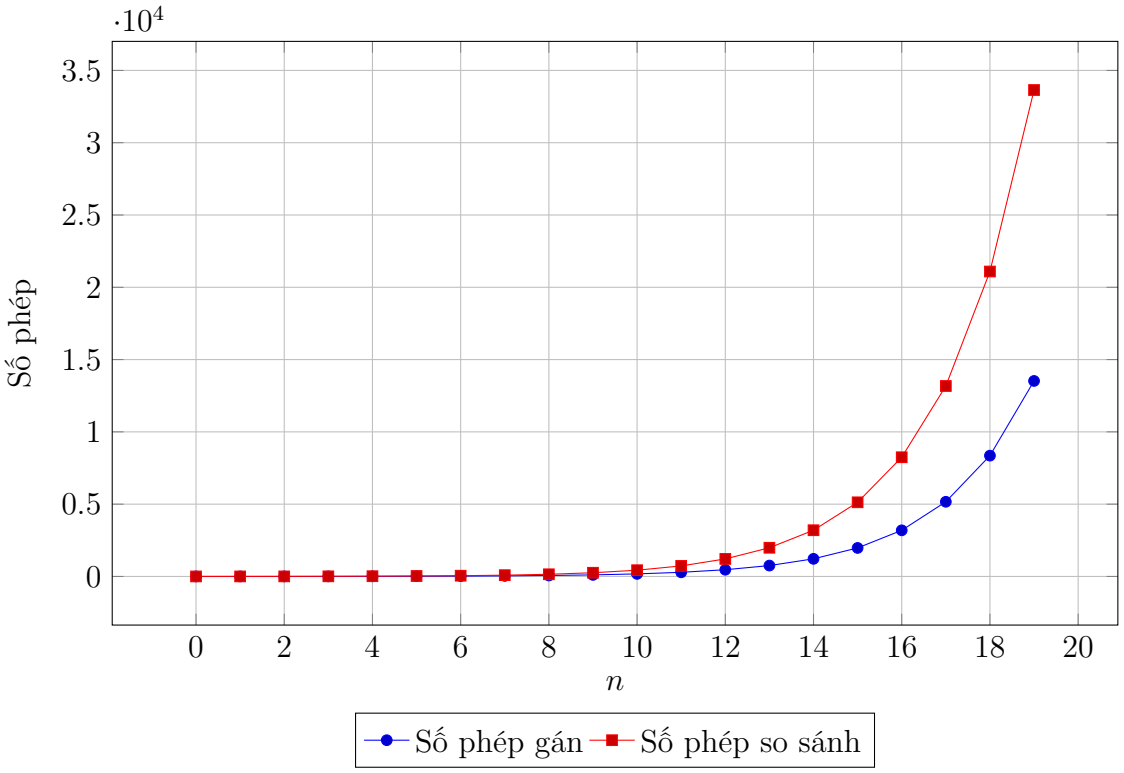
Đây là mã nguồn Python triển khai thuật toán Fibonacci dùng đệ quy với các biến đếm số phép gán (assignments) và số phép so sánh (comparisons)

```
count_assign = 0
count_compare = 0

def fibo(n):
    global count_assign, count_compare
    count_compare += 1
    if n <= 1:
        count_assign += 1
        return n
    count_assign += 1
    return fibo(n - 1) + fibo(n - 2)

# Test with different values of n and store the results
results = []
for n in range(20):  # Change the value in range to test with different n values
    count_assign = 0
    count_compare = 0
    result = fibo(n)
    results.append((n, count_assign, count_compare))

import pandas as pd
import matplotlib.pyplot as plt

# Create the statistics table
df = pd.DataFrame(results, columns=['n', 'count_assign', 'count_compare'])
print(df)

# Plot the graph
plt.figure(figsize=(10, 5))
plt.plot(df['n'], df['count_assign'], label='Number of assignments')
plt.plot(df['n'], df['count_compare'], label='Number of comparisons')
plt.xlabel('n')
plt.ylabel('Number of operations')
plt.title('Number of assignments and comparisons vs. n')
plt.legend()
plt.grid(True)
plt.show()
```

- **Kết quả thống kê**
  Bảng đây mô tả số phép gán và số phép so sánh cho các giá trị Fibonacci từ 0 đến 9.
- **Đồ thị biểu diễn**
  Đồ thị dưới đây biểu diễn sự thay đổi của số phép gán và số phép so sánh theo giá trị n.

n	Số phép gán	Số phép so sánh
0	1	1
1	1	2
2	2	4
3	4	8
4	8	16
5	14	30
6	24	52
7	40	88
8	66	150
9	108	256
10	176	432
11	286	728
12	464	1212
13	752	1984
14	1218	3196
15	1972	5128
16	3190	8244
17	5164	13172
18	8356	21092
19	13522	33644

Bảng 1: Bảng thống kê số phép gán và số phép so sánh theo n



Hình 1: Đồ thị số phép gán và số phép so sánh theo n

Thuật toán tính số Fibonaci không đệ quy với thêm biến đếm

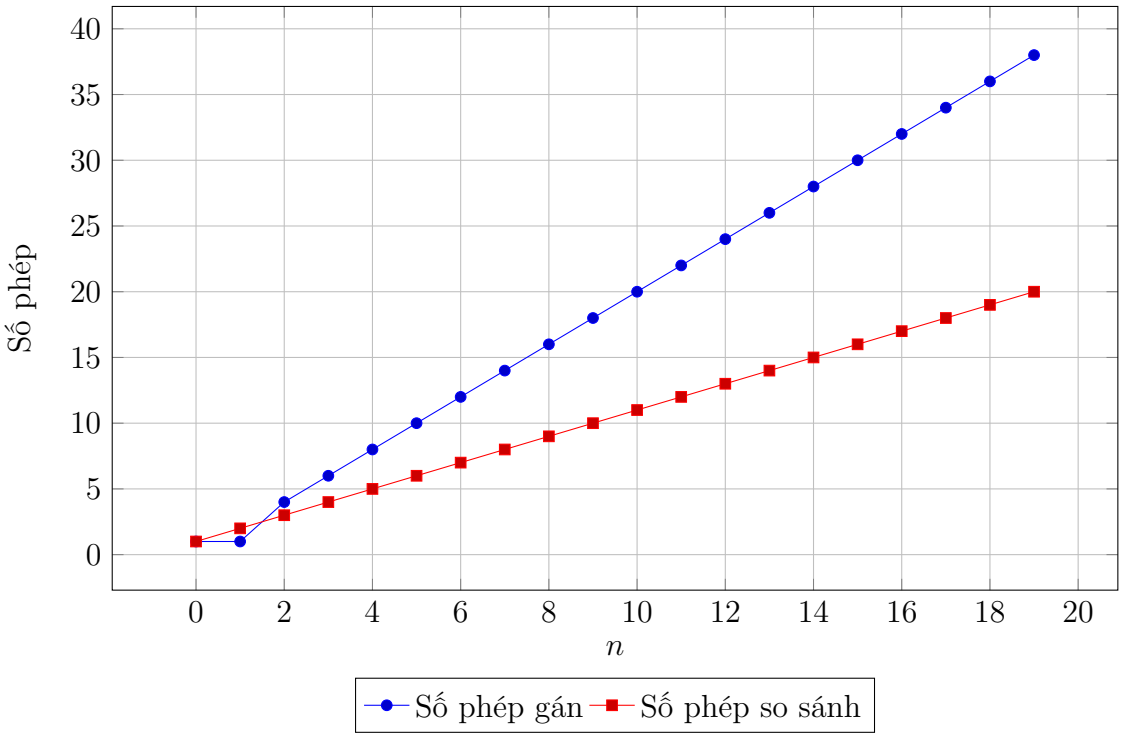
Đây là mã nguồn Python triển khai thuật toán Fibonacci không dùng đệ quy với các biến đếm số phép gán (assignments) và số phép so sánh (comparisons)

```python
import matplotlib.pyplot as plt

def fibonacci(n):
    count_assign = 0  # Variable to count assignment operations
    count_compare = 0  # Variable to count comparison operations

    if n <= 0:
        count_compare += 1
        return 0, count_assign, count_compare
    elif n == 1:
        count_compare += 2
        return 1, count_assign, count_compare

    count_compare += 2  # Two comparisons are already performed above

    fib_0 = 0
    fib_1 = 1
    count_assign += 2  # Initialize first two values

    for i in range(2, n + 1):
        fib_n = fib_0 + fib_1
        fib_0 = fib_1
        fib_1 = fib_n
        count_assign += 3  # Reassign three values
        count_compare += 1  # Comparison in the loop

    return fib_n, count_assign, count_compare

# Count and plot
n_values = range(1, 21)
assign_counts = []
compare_counts = []

for n in n_values:
    _, count_assign, count_compare = fibonacci(n)
    assign_counts.append(count_assign)
    compare_counts.append(count_compare)

# Plot
plt.figure(figsize=(10, 5))
plt.plot(n_values, assign_counts, label='Number of assignments', marker='o')
plt.plot(n_values, compare_counts, label='Number of comparisons', marker='x')
plt.xlabel('n')
plt.ylabel('Operation Count')
plt.title('Number of assignments and comparisons with n in Fibonacci Algorithm')
plt.legend()
plt.grid(True)
plt.show()
```

- **Kết Quả Thống Kê**
  Bảng đây mô tả số phép gán và số phép so sánh cho các giá trị Fibonacci từ 0 đến 9.

n	Số phép gán	Số phép so sánh
0	1	1
1	1	2
2	4	3
3	6	4
4	8	5
5	10	6
6	12	7
7	14	8
8	16	9
9	18	10
10	20	11
11	22	12
12	24	13
13	26	14
14	28	15
15	30	16
16	32	17
17	34	18
18	36	19
19	38	20

Bảng 2: Bảng thống kê số phép gán và số phép so sánh theo n



Hình 2: Đồ thị số phép gán và số phép so sánh theo n

b) Dùng phương pháp lý thuyết để đếm số phép so sánh được sử dụng ở các câu a) và số phép S được sử dụng ở câu a), xét với n tổng quát (có thể đối chiếu lại với thực nghiệm).

- Thuật toán dùng đệ quy:
  Gọi a_n là số phép gán của thuật toán fibonaci(n)
  Ta có $\begin{cases} a_0 = 1, a_0 = 1 \\ a_n = a_{n-1} + a_{n-2}, n \geq 2 \end{cases}$

Đặt $f(z) = \sum_{n=0}^{+\infty} a_n z^n$
Giả sử bán kính hội tụ của $f(z)$ là $R > 0$

$$f(z) = \sum_{n=0}^{+\infty} a_n z^n = a_0 z^0 + a_1 z^1 + \sum_{n=2}^{+\infty} (a_{n-1} z^n + a_{n-2} z^n)$$
$$= 1 + z + \sum_{n=2}^{+\infty} (a_{n-1} z^n + a_{n-2} z^n)$$
$$f(z) = 1 + z + z \sum_{n=0}^{+\infty} a_n z^n + z^2 \sum_{n=0}^{+\infty} a_n z^n$$
$$f(z) = 1 + z + z f(z) + z^2 f(z)$$
$$f(z) = 1 + z + f(z)(z + z^2)$$
$$f(z) = \frac{1 - z + z^2}{(1-z)(1-z-z^2)} = \frac{1}{1-z} + \frac{1 - z - z^2}{1 - z}$$
$$= \frac{1}{1-z} + \frac{-1 + 2z^2}{1 - z - z^2}$$
$$= \frac{1}{1-z} + \frac{-1}{1-z} + \frac{2}{1 - z - z^2}$$
$$= \frac{-1}{1-z} + \frac{2}{1 - z - z^2}$$

$$k(z) = \frac{1}{1 - z - z^2} = \frac{1}{(z - \frac{-1+\sqrt{5}}{2})(z - \frac{-1-\sqrt{5}}{2})}$$

$$\Leftrightarrow k(z) = -\frac{1}{(\frac{-1+\sqrt{5}}{2} - z)(\frac{-1-\sqrt{5}}{2} - z)}$$

$$\Leftrightarrow k(z) = -\left(\frac{1}{\frac{-1+\sqrt{5}}{2}\left(1 - \frac{2}{-1+\sqrt{5}} z\right)}\right)\left(\frac{1}{\frac{-1-\sqrt{5}}{2}\left(1 - \frac{2}{-1-\sqrt{5}} z\right)}\right)$$

$$\Leftrightarrow k(z) = -\left(\frac{1}{-1+\sqrt{5}} \frac{1}{1 - \frac{2}{-1+\sqrt{5}} z}\right)\left(\frac{1}{-1-\sqrt{5}} \frac{1}{1 - \frac{2}{-1-\sqrt{5}} z}\right)$$

$$k(z) = \frac{1}{2}\left(\frac{1}{1 - \frac{2}{-1+\sqrt{5}} z}\right)\left(\frac{1}{1 - \frac{2}{-1-\sqrt{5}} z}\right) = \frac{1}{2} \sum_{n=0}^{\infty} \left(\frac{2}{\sqrt{5}-1} z\right)^n \sum_{n=0}^{\infty} \left(\frac{-2}{\sqrt{5}+1} z\right)^n$$

Với $h(z) = \sum_{n=0}^{\infty} \left(\frac{2}{\sqrt{5}-1}\right)^n z^n$ và $g(z) = \sum_{n=0}^{\infty} \left(\frac{-2}{\sqrt{5}+1}\right)^n z^n$

$$k(z) = h(z) \cdot g(z)$$

Đặt $k(z) = \sum_{n=0}^{\infty} d_n z^n$

$$d_n = \sum_{k=0}^{n} b_k \cdot c_{n-k} = \sum_{k=0}^{n} \left(\frac{2}{\sqrt{5}-1}\right)^k \left(\frac{-2}{\sqrt{5}+1}\right)^{n-k}$$

Ta có:

$$f(z) = \sum_{n=0}^{\infty} a_n z^n = \frac{-1}{1-z} + \frac{2}{1-z-z^2} = -1 \sum_{n=0}^{\infty} z^n + 2 \sum_{n=0}^{\infty} d_n z^n$$

Suy ra:

$$a_n = -1 + d_n = -1 + 2\sum_{k=0}^{n} \left(\frac{2}{\sqrt{5}-1}\right)^k \left(\frac{-2}{\sqrt{5}+1}\right)^{n-k}$$