TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA KHOA HỌC MÁY TÍNH
BÀI TẬP MÔN PHÂN TÍCH VÀ THIẾT KẾ THUẬT TOÁN
HOMEWORK #01: ĐÁNH GIÁ THUẬT TOÁN DÙNG KỸ THUÂT TOÁN SƠ CẤP

GV hướng dẫn: Huỳnh Thị Thanh Thương (viết sai tên trừ điểm)
Nhóm thực hiện:
1. Đoàn Thanh Tùng 21521646
2. Đàm thành Nam 21522354
3. Lê Phan Hiển 21520839
4. Lê Khai Trí 21521565

TP.HCM, ngày 5 tháng 10 năm 2023

Bài 1:

Thành lập phương trình đệ quy, kèm giải thích cách thành lập. Không giải phương trình

a.

a). Gửi ngân hàng 1000 USD, lãi suất 12%/năm. Số tiền có được sau 30 năm là bao nhiêu?

Gọi T(n) là số tiền nhận được sau n năm

⇨ T(n-1) là số tiền nhận được sau n - 1 năm

Khi n = 0 thì số tiền nhận được là 1000

⇨ T(0) = 1000

Khi n > 0 thì số tiền nhận được là T(n) = T(n-1) + T(n-1)*12% = T(n-1) * (1 + 0.12)

Vậy phương trình đệ quy cho bài toán này là:

$$T(n) = \begin{cases} 1000 & khi\ n = 0 \\ \mathrm{T(n-1)} * 1.12 & khi\ \ n > 0 \end{cases}$$

b.

```
long Fibo(int n)
{
  if (n == 0 || n == 1)
    return 1;
  return Fibo(n–
1)+Fibo(n–2);
}
```

Gọi T(n) là chi phí thực hiện bài toán khi đầu vào bằng n

- ⇨ T(n-1) là chi phí thực hiện bài toán khi đầu vào bằng n – 1
- ⇨ T(n-2) là chi phí thực hiện bài toán khi đầu vào bằng n – 2

Khi n = 0 hoặc n = 1 thì kết quả trả về của hàm là 1

- ⇨ T(0) = T (1) = C1 (C1 nhỏ, hằng số)

Khi n > 1 thì kết quả trả về của hàm là Fibo(n-1) + Fibo(n-2)

- ⇨ T(n) = T(n-1) + T(n-2) + C2 (với C2 là chi phí tổng hợp kết quả)

Vậy phương trình đệ quy cho bài toán này là:

$$T(n) = \begin{cases} C1 & khi\ n \leq 1 \\ \mathrm{T(n-1)\ +\ T(n-2)\ +\ C2} & khi\ \ n > 1 \end{cases}$$

c.

```
public int g(int n) {
    if (n == 1)
        return 2;
    else
        return 3 * g(n / 2) + g( n / 2) + 5;
}
```

Gọi T(n) là chi phí thực hiện bài toán khi đầu vào bằng n

⇨ T(n-1) là chi phí thực hiện bài toán khi đầu vào bằng n – 1

Khi n = 1 thì kết quả của hàm trả về là 2

⇨ T(1) = C_1 (C1 nhỏ, hằng số)

Khi n > 1 thì kết quả trả về của hàm là $3 * g\left(\frac{n}{2}\right) + g\left(\frac{n}{2}\right) + 5$

⇨ T(n) = $2T\left(\frac{n}{2}\right) + C_2$ (với C_2 là chi phí tổng hợp kết quả)

Vậy phương trình đệ quy cho bài toán này là:

T(n) = $\begin{cases} C_1 \; khi \; n = 1 \\ 2T\left(\frac{n}{2}\right) + C_2 \; khi \; n > 1 \end{cases}$

d.

```
long xn(int n)
{
   if (n == 0) return 1;
   long s = 0;
   for (int i=1; i<=n; i++)
      s = s + i*i*xn(n–i);
   return s;
}
```

Gọi T(n) là chi phí thực hiện bài toán khi đầu vào bằng n

Khi n = 0 thì kết quả của hàm trả về là 1

⇨ T(0) = C_1 (C_1 nhỏ, hằng số)

Khi n >= 1 thì chi phí là

T(n) = $\sum_{i=1}^{n} T(n-i) + C_2 n$ (C_2 là chi phí tổng hợp kết quả)

Vậy phương trình đệ qui cho bài toán này là:

T(n) = $\begin{cases} C_1 \; khi \; n = 0 \\ \sum_{i=1}^{n} T(n-i) + C_2 n \; khi \; n \geq 1 \end{cases}$

e.

```
Draw (n)
{    if (n < 1) return 0;
     for (i = 1 ; i <= n; i++ )
          for (j = 1 ; j <=n ; j++)
               print ("*");
     Draw (n-3);
}
```

Gọi T(n) là chi phí thực hiện bài toán khi đầu vào bằng n

Khi n < 1 thì kết quả của hàm trả về là 0

- ⇨ $T(n < 1) = C_1$ (C_1 nhỏ, hằng số)

Khi n >= 1 thì kết quả trả về là

$T(n) = T(n-3) + C_2\, n^2$ (C_2 là chi phí tổng hợp kết quả)

Vậy phương trình đệ qui cho bài toán này là:

$$T(n) = \begin{cases} C_1 \; khi \; n < 1 \\ T(n-3) + C_2\, n^2 khi \; n \geq 1 \end{cases}$$

f.

f)

Gọi T(n) là số phép cộng cần thực hiện khi gọi Zeta (k). Hãy thiết lập công thức truy hồi cho T(n)

```
Cho hàm:
Zeta (n)
{   if (n == 0) Zeta = 6;
    else
    {   k = 0;
        Ret = 0;
        while  (k<=n-1)
        {       Ret = Ret +Zeta(k) ;
                k = k+1;
        }
        Zeta = Ret;
}
```

Gọi T(n) là số phép cộng cần thực hiện khi gọi Zeta(k).

Khi n < 1 thì kết quả của hàm trả về là 6, không có phép cộng nào cần thực hiện

⇨ T(n < 1) = 0

Khi n >= 1 thì vòng lặp while thực hiện n lần, số phép cộng cần thực hiện là n(2 + T(k))

⇨ T(n) = 2n + nT(n-1)

Vậy phương trình đệ qui cho bài toán này là:

T(n) = $\begin{cases} 0 \ khi \ n < 1 \\ \sum_{k=0}^{n-1} T(n-i) + 2nC_2 \ khi \ n \geq 1 \end{cases}$

g)

Giải thuật chia để trị

X · Y = AC · 10n + (A – B)(D – C) + AC + BD · 10n/2 + BD

Với X, Y có 1 chữ số, nghĩa là n = 1, ta có T(n) = 1 = C1.

Với n > 1, giải thuật sẽ phải tính tích các giá trị AC; (A – B)(D – C) + AC + BD; BD với mỗi giá trị có số chữ số là n/2.

Trường hợp 1: Trong mã giả có 2 lần gọi hàm multi(A,C) riêng biệt thì ta phải gọi đệ quy 5 lần để tính các giá trị là AC; (A - B)(D - C); AC; BD; BD

⇨ T(n) = 5 T(n / 2) + C2

Vậy phương trình đệ quy:

T(n) = $\begin{cases} 0\ khi\ n = 1 \\ 5\,T\left(\frac{n}{2}\right) + C_2\ khi\ n > 1 \end{cases}$

Trường hợp 2: Trong mã giả khi ta có lưu lại kết quả của các phép tính trước nên ta chỉ cần gọi đệ quy 3 lần để tính các tích là AC; (A - B)(D - C) và BD (các tích AC, BD đã được lưu lại nên chỉ cần lấy ra dùng không cần tính nữa (Dynamic Programming ?))

⇨ T(n) = 3 T(n / 2) + C2

Vậy phương trình đệ quy:

T(n) = $\begin{cases} 0\ khi\ n = 1 \\ 3\,T\left(\frac{n}{2}\right) + C_2\ khi\ n > 1 \end{cases}$

Bài 2:

Bài tập 2:Giải các phương trình đệ quy sau bằng PP truy hồi

Lưu ý: PHẢI RÚT GỌN T(n), GỌN NHẤT CÓ THỂ

1. $T(n) = T(n-1) + n$ $\quad T(1) = 1$
2. $T(n) = 2T(n/2) + n$ $\quad T(1) = 1$
3. $T(n) = 2T(n/2) + n^2$ $\quad T(1) = 1$
4. $T(n) = 2T\left(\frac{n}{2}\right) + \log n$ $\quad T(1) = 1$
5. $T(n) = 8T(n/2) + n^3$ $\quad T(1) = 1$
6. $T(n) = 4T(n/3)+n$ $\quad T(1) = 1$
7. $T(n) = 9T(n/3) + n^2$ $\quad T(1) = 1$
8. $T(n) = 2T(\sqrt{n}) + 1$ $\quad T(2) = 0$

1. T(n) = $T(n-1) + n$ $\qquad$ T(1) = 1

$= [\,T(n-2) + n-1\,] + n$

$= T(n-2) + 2n-1$

$= [\,T(n-3) + n-2\,] + 2n-1$

$= T(n-3) + 3n-3$

$= \mathrm{T(n-i)} + \mathrm{i*n} - \sum_{k=0}^{i-1} k$

Quá trình dừng lại khi n – i = 1 ⇔ i = n - 1

T(n) = $\mathrm{T(1)} + \mathrm{n(n-1)} - \sum_{k=0}^{n-2} k$

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

$= \frac{n^2}{2} + \frac{n}{2}$

2. T(n) = $2T\left(\frac{n}{2}\right) + n$ T(1) = 1

$= 2\left[2T\left(\frac{n}{4}\right) + \frac{n}{2}\right] + n$

$= 4T\left(\frac{n}{4}\right) + 2n$

$= 4\left[2T\left(\frac{n}{8}\right) + \frac{n}{4}\right] + 2n$

$= 8T\left(\frac{n}{8}\right) + 3n$

$= 2^iT\left(\frac{n}{2^i}\right) + i*n$

Quá trình dừng lại khi $\frac{n}{2^i} = 1 \Leftrightarrow n = 2^i \Leftrightarrow$ i = $\log n$

T(n) = $2^{\log n}\, T(1) + n*\log n$

= n*T(1) + $n*\log n$

= n + $n*\log n$

$= n(1+\log n)$

3. T(n) = $2T\left(\frac{n}{2}\right) + n^2$ T(1) = 1

$= 2\left[2T\left(\frac{n}{4}\right) + \frac{n^2}{4}\right] + n^2$

$= 4T\left(\frac{n}{4}\right) + \frac{n^2}{2} + n^2$

$= 8T\left(\frac{n}{8}\right) + \frac{n^2}{4} + \frac{n^2}{2} + n^2$

$= 2^iT\left(\frac{n}{2^i}\right) + \sum_{k=0}^{i-1}\frac{1}{2^k}*n^2$

Quá trình dừng lại khi $\frac{n}{2^i} = 1 \Leftrightarrow$ i = $\log n$

T(n) = $2^{\log n} * T(1) + n^2\sum_{k=0}^{i-1}\frac{1}{2^k}$

$= n + \frac{n^2\left(\left(\frac{1}{2}\right)^{\log n - 1} - 1\right)}{\frac{1}{2} - 1}$

4. T(n) = $2T\left(\frac{n}{2}\right) + \log n$ T(1) = 1

= $2[2T\left(\frac{n}{4}\right) + \log\frac{n}{2}] + \log n$

= $4T\left(\frac{n}{4}\right) + 2\log\frac{n}{2} + \log n$

= $4[2T\left(\frac{n}{8}\right) + \log\frac{n}{4}] + 2\log\frac{n}{2} + \log n$

= $8T\left(\frac{n}{8}\right) + 4\log\frac{n}{4} + 2\log\frac{n}{2} + \log n$

= $2^i * T\left(\frac{n}{2^i}\right) + \sum_{k=0}^{i-1} 2^k * \log\frac{n}{2^k}$

Quá trình dừng lại khi $\frac{n}{2^i} = 1 \Leftrightarrow i = \log n$

T(n) = $2^{\log n} * T(1) + \sum_{k=0}^{\log n - 1} 2^k * \log\frac{n}{2^k}$

= $n + \sum_{k=0}^{\log n - 1} 2^k * \log\frac{n}{2^k}$

= $n + \sum_{k=0}^{\log n - 1} 2^k(\log n - k)$

= $n + \sum_{k=0}^{\log n - 1} 2^k \log n - \sum_{k=0}^{\log n - 1} 2^k * k$

= $n + \log n\left(2^{\log n} - 1\right) - (\log n - 2) * 2^{\log n} + 2$

= $n + \log n\left(2^{\log n} - 1\right) - n(\log n - 2) + 2$

5. T(n) = $8T\left(\frac{n}{2}\right) + n^3$ T(1) = 1

= $8\left[8T\left(\frac{n}{4}\right) + \frac{n^3}{8}\right] + n^3$

= $64T\left(\frac{n}{4}\right) + 2n^3$

= $64\left[8T\left(\frac{n}{8}\right) + \frac{n^3}{64}\right] + 2n^3$

= $512T\left(\frac{n}{8}\right) + 3n^3$

= $8^i T\left(\frac{n}{2^i}\right) + i * n^3$

Quá trình dừng lại khi $\frac{n}{2^i} = 1 \Leftrightarrow i = \log n$

T(n) = $8^{\log n} * T(1) + n^3 \log n$

= $\left(2^{\log n}\right)^3 + n^3 \log n$

= $n^3 + n^3 \log n$

= $n^3(1 + \log n)$

6. T(n) = $4T\left(\frac{n}{3}\right) + n$ T(1) = 1

= $4\left[4T\left(\frac{n}{9}\right) + \frac{n}{3}\right] + n$

= $16T\left(\frac{n}{9}\right) + \frac{4n}{3} + n$

= $16\left(T\left(\frac{n}{27}\right) + \frac{n}{9}\right] + \frac{4n}{3} + n$

= $64T\left(\frac{n}{27}\right) + \frac{16n}{9} + \frac{4n}{3} + n$

= $4^i T\left(\frac{n}{3^i}\right) + \sum_{k=1}^{i-1} \left(\frac{4}{3}\right)^k n$

Quá trình dừng lại khi $\frac{n}{3^i} = 1 \Leftrightarrow i = \log_3 n$

T(n) = $4^{\log_3 n} * T(1) + n \sum_{k=1}^{\log_3 n - 1} \left(\frac{4}{3}\right)^k$

= $4^{\log_3 n} + n \frac{\left(\frac{4}{3}\right)^{\log_3 n} - 1}{\frac{4}{3} - 1}$

7. T(n) = $9T\left(\frac{n}{3}\right) + n^3$ T(1) = 1

= $9\left[9T\left(\frac{n}{9}\right) + \frac{n^3}{27}\right] + n^3$

= $81T\left(\frac{n}{9}\right) + \frac{n^3}{3} + n^3$

= $81\left[9T\left(\frac{n}{27}\right) + \frac{n^3}{729}\right] + \frac{n^3}{3} + n^3$

= $729T\left(\frac{n}{27}\right) + \frac{n^3}{9} + \frac{n^3}{3} + n^3$

= $9^i T\left(\frac{n}{3^i}\right) + \sum_{k=0}^{i-1} \frac{1}{3^i} * n^3$

Quá trình dừng lại khi $\frac{n}{3^i} = 1 \Leftrightarrow i = \log_3 n$

T(n) = $9^{\log_3 n} * T(1) + n^3 \sum_{k=0}^{i-1} \frac{1}{3^i}$

= $\left(3^{\log_3 n}\right)^2 + \frac{n^3\left(\left(\frac{1}{3}\right)^{\log_3 n} - 1\right)}{\frac{1}{3}-1}$

= $n^2 + \frac{n^3\left(\frac{1}{n}-1\right)}{-\frac{2}{3}}$

8. T(n) = $2T\left(\sqrt{n}\right) + 1 = 2T\left(n^{\frac{1}{2}}\right) + 1$ T(2) = 0

= $2\left[2T\left(n^{\frac{1}{4}}\right) + 1\right] + 1$

= $4T\left(n^{\frac{1}{4}}\right) + 2 + 1$

= $4\left[2T\left(n^{\frac{1}{8}}\right) + 1\right] + 2 + 1$

= $8T\left(n^{\frac{1}{8}}\right) + 4 + 2 + 1$

= $2^i T\left(n^{\frac{1}{2^i}}\right) + \sum_{k=0}^{i-1} 2^k$

Quá trình dừng lại khi $n^{\frac{1}{2^i}} = 2 \Leftrightarrow \frac{1}{2^i} = \log_n 2 \Leftrightarrow \frac{1}{\log_n 2} = 2^i \Leftrightarrow i = \log\log_n 2$

T(n) = $2^{\log\log_n 2} T(2) + \sum_{k=0}^{\log\log_n 2-1} 2^k$

= $0 + 2^{\log\log_n 2} - 1$

Bài 3:

Giải các phương trình đệ quy sau dùng phương trình đặc trưng

a. T (n) = 4 T (n − 1) - 3 T (n − 2)
T (0) = 1
T (1) = 2

b. T(n) = 4T(n-1) – 5T(n-2) + 2T(n-3)
T (0) = 0
T (1) = 1
T(2) = 2

c. T (n) = T (n − 1) + T (n − 2)
T (0) = 1
T (1) = 1

a) T(n) = 4 T(n - 1) – 3 T(n - 2) (1)

T(0) = 1

T(1) = 2

Giải:

Đặt T(n) = X^n

1 ⇔ $X^n = 4 X^{n-1} - 3 X^{n-2}$

⇔ $X^n - 4 X^{n-1} + 3 X^{n-2} = 0$

⇔ $X^{n-2} (X^2 - 4X + 3) = 0$

⇔

$$\left[\begin{array}{l} X^{n-2} = 0 \\ X^2 - 4X + 3 = 0\ (*) \end{array}\right.$$

(*) được gọi là phương trình đặc trưng của phương trình truy hồi

(*) ⇔ (X - 1)(X - 3) = 0

(*) có một nghiệm đơn $X_1 = 1$, 1 nghiệm đơn $X_2 = 3$

Dạng của T(n) = $C_1 X_1^n + C_2 X_2^n$
= $C_1 + C_2 3^n$

Ta có T(0) = 1

T(1) = 1

⇨ $\begin{cases} C_1 + C_2 = 1 \\ C_1 + 3C_2 = 2 \end{cases}$

⇔ $\begin{cases} C_1 = 1/2 \\ C_2 = 1/2 \end{cases}$

Vậy T(n) = $\frac{3^n}{2} + \frac{1}{2}$

b) T(n) = 4 T(n - 1) – 5 T(n - 2) + 2 T(n - 3) (*)

T(0) = 0

T(1) = 1

T(2) = 2

Giải:

Đặt T(n) = X^n

1 ⇔ $X^n = 4 X^{n-1} - 5 X^{n-2} + 2 X^{n-3}$

⇔ $X^n - 4 X^{n-1} + 5 X^{n-2} - 2 X^{n-3} = 0$

⇔ $X^{n-3}(X^3 - 4X^2 + 5X - 2) = 0$

⇔

$$\left[\begin{array}{l} X^{n-3} = 0 \\ X^3 - 4X^2 + 3X - 2 = 0\ (*) \end{array}\right.$$

(*) được gọi là phương trình đặc trưng của phương trình truy hồi

	1	-4	5	-2
1	1	-3	2	0

(*) ⇔ $(X - 1)(X^2 - 3X + 2) = 0$

(*) có một nghiệm kép $X_1 = 1$, 1 nghiệm đơn $X_2 = 2$

Dạng của $T(n) = C_1 X_1^n + C_2\, n\, X_1^n + C_3 X^n$
$= C_1 + C_2 n + C_3 2^n$

Ta có $T(0) = 0$

$T(1) = 1$

$T(2) = 2$

$$\Rightarrow \begin{cases} C_1 + C_3 = 0 \\ C_1 + C_2 + 2C_3 = 1 \\ C_1 + 2C_2 + 4C_3 = 2 \end{cases}$$

$$\Leftrightarrow \begin{cases} C_1 = 0 \\ C_2 = 1 \\ C_3 = 0 \end{cases}$$

Vậy $T(n) = n$

c) $T(n) = T(n - 1) + T(n - 2)$ (1)

$T(0) = 1$

$T(1) = 1$

<u>Giải:</u>

Đặt $T(n) = X^n$

1 ⇔ $X^n = X^{n-1} + X^{n-2}$

⇔ $X^n - X^{n-1} - X^{n-2} = 0$

⇔ $X^{n-2}(X^2 - X - 1) = 0$

⇔
$$\left[\begin{array}{l} X^{n-2} = 0 \\ X^2 - X - 1 = 0 \ (*) \end{array}\right.$$

(*) được gọi là phương trình đặc trưng của phương trình truy hồi

(*) ⇔ $\left(X - \frac{1+\sqrt{5}}{2}\right)\left(X - \frac{1-\sqrt{5}}{2}\right)$

(*) có một nghiệm đơn $X_1 = \frac{1+\sqrt{5}}{2}$, 1 nghiệm đơn $X_2 = \frac{1-\sqrt{5}}{2}$

Dạng của $T(n) = C_1 X_1^n + C_2 X_2^n$

$= C_1\left(\frac{1+\sqrt{5}}{2}\right)^n + C_2\left(\frac{1-\sqrt{5}}{2}\right)^n$

Ta có $T(0) = 1$

$T(1) = 1$

⇨ $\begin{cases} C_1 + C_2 = 1 \\ C_1\left(\frac{1+\sqrt{5}}{2}\right) + C_2\left(\frac{1-\sqrt{5}}{2}\right) = 2 \end{cases}$

⇔ $\begin{cases} C_1 = \frac{\sqrt{5}+1}{2\sqrt{5}} \\ C_2 = \frac{\sqrt{5}-1}{2\sqrt{5}} \end{cases}$

Vậy $T(n) = \frac{\sqrt{5}+1}{2\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^n + \frac{\sqrt{5}+1}{2\sqrt{5}}\left(\frac{1-\sqrt{5}}{2}\right)^n$

Bài 4:

Giải phương trình đệ quy sau dùng phương pháp hàm sinh:

$$a.\, T(n) = \begin{cases} 1 & khi\ n = 0 \\ 2T(n-1) + 7 & khi\ n > 0 \end{cases}$$

$$b.\, T(n) = \begin{cases} 1 & khi\ n = 0, \\ 2 & khi\ n = 1, \\ 7T(n-1) - 12T(n-2) & khi\ n \geq 2 \end{cases}$$

$$c.\, T(n+1) = T(n) + 2(n+2)$$
$$T(0) = 3$$

a)

$$T(n) = \begin{cases} 1 & khi\ n = 0 \\ 2T(n-1) + 7 & khi\ \ n > 0 \end{cases}$$

<u>Giải:</u>

Hàm sinh của dãy vô hạn $\{T(n)\}_{n=0}^{\infty}$ có dạng:

f(X) = $\sum_{n=0}^{\infty} T(n)\, X^n$ = T(0) X^0 + T(1) X^1 + …

= $\sum_{n=1}^{\infty}[2T(n-1)+7]X^n$ + T(0) X^0

= 2 $\sum_{n=1}^{\infty} T(n-1)\, X^n$ + 7 $\sum_{n=1}^{\infty} X^n$ + 1 (*)

Đặt A = $\sum_{n=1}^{\infty} T(n-1)\, X^n$, B = $\sum_{n=1}^{\infty} X^n$, ta đi tính A và B

A = $\sum_{n=1}^{\infty} T(n-1)\, X^n$ = X $\sum_{n=1}^{\infty} T(n-1)\, X^{n-1}$ = X (T(0) X^0 + T(1) X^1 + …)

= X f(X)

B = $\sum_{n=1}^{\infty} X^n$ = $\sum_{n=0}^{\infty} X^n$ – X^0

= $\frac{1}{1-X} - 1$

Khi đó thì (*) ⇔ f(X) = 2X f(X) + 7 ($\frac{1}{1-X} - 1$) + 1

⇔ f(X) (1 – 2X) = $\frac{7}{1-X} - 6$

⇔ f(X) = $\frac{1+X}{(1-X)(1-2X)}$

⇔ f(X) = $\frac{-2}{1-X} + \frac{3}{1-2X}$

= -2 $\sum_{n=0}^{\infty} X^n$ + 3 $\sum_{n=0}^{\infty} (2X)^n$

=$\sum_{n=0}^{\infty} X^n \, (3\,.\,2^n - 2)$

Vậy nên T(n) = $3\,.\,2^n - 2$

b)

$$T(n) = \begin{cases} 1 & khi\ n = 0 \\ 2 & khi\ n = 1 \\ 7T(n-1) - 12T(n-2) & khi\ \ n \geq 2 \end{cases}$$

<u>Giải:</u>

Hàm sinh của dãy vô hạn $\{T(n)\}_{n=\,0}^{\infty}$ có dạng:

f(X) = $\sum_{n=0}^{\infty} T(n)\, X^n$ = T(0) X^0 + T(1) X^1 + …

= $\sum_{n=2}^{\infty} [7T(n-1) - 12T(n-2)]X^n$ + T(0) X^0 + T(1) X^1

= 7 $\sum_{n=2}^{\infty} T(n-1)\, X^n$ - 12 $\sum_{n=2}^{\infty} T(n-2)X^n$ + 2X + 1 (*)

Đặt A = $\sum_{n=2}^{\infty} T(n-1)\, X^n$, B = $\sum_{n=2}^{\infty} T(n-2)X^n$, ta đi tính A và B

A = $\sum_{n=2}^{\infty} T(n-1)\, X^n$ = X $\sum_{n=2}^{\infty} T(n-1)\, X^{n-1}$ = X (T(1) X^1 + T(2) X^2 + …)

= X [f(x) - T(0) X^0] = X (f(X) - 1) = X f(X) – X

B =$\sum_{n=2}^{\infty} T(n-2)X^n$ = X^2 $\sum_{n=2}^{\infty} T(n-2)\, X^{n-2}$

= X^2 (T(0) X^0 + T(1) X^1 + …)

= X^2 f(X)

Khi đó thì (*) ⇔ f(X) = 7(X f(X) - X) - 12 X^2 f(X) + 2X + 1

⇔ f(X) (1 – 7X + 12X^2) = 1 – 5X

⇔ f(X) = $\frac{1-5X}{(1-3X)(1-4X)}$

⇔ f(X) = $\frac{2}{1-3X} - \frac{1}{1-4X}$

= 2 $\sum_{n=0}^{\infty}(3X)^n$ - $\sum_{n=0}^{\infty}(4X)^n$

=$\sum_{n=0}^{\infty} X^n\,(2.3^n - 4^n)$

Vậy nên T(n) = $2.3^n - 4^n$

c)

$$\begin{cases} T(0) = 3 & khi\ n = 0 \\ T(n+1) = T(n) + 2(n+2) \quad (1) & khi\ \ n > 0 \end{cases}$$

<u>Giải:</u>

<u>Ta có : (1) ⇔ T (n + 1) = T(n + 1 - 1) + 2 (n + 1 + 1)</u>

<u>⇔ T(n) = T(n - 1) + 2 (n + 1)</u>

Hàm sinh của dãy vô hạn $\{T(n)\}_{n=0}^{\infty}$ có dạng:

f(X) = $\sum_{n=0}^{\infty} T(n)\, X^n$ = T(0) X^0 + T(1) X^1 + …

= $\sum_{n=1}^{\infty}[T(n-1) + 2(n+1)]X^n$ + T(0) X^0

= $\sum_{n=1}^{\infty} T(n-1)\, X^n$ + 2 $\sum_{n=1}^{\infty}(n+1)X^n$ + 3 (*)

Đặt A = $\sum_{n=1}^{\infty} T(n-1)\, X^n$, B = $\sum_{n=1}^{\infty}(n+1)X^n$, ta đi tính A và B

A = $\sum_{n=1}^{\infty} T(n-1)\, X^n$ = X $\sum_{n=1}^{\infty} T(n-1)\, X^{n-1}$ = X (T(0) X^0 + T(1) X^1 + …)

= X f(X)

B = $\sum_{n=1}^{\infty}(n+1)X^n$ = $\sum_{n=0}^{\infty}(n+1)X^n$ – (0 + 1)X^0

= $\frac{1}{(1-x)^2} - 1$

Khi đó thì (*) ⇔ f(X) = X f(X) + 2 ($\frac{1}{(1-x)^2} - 1$) + 3

⇔ f(X) (1 – X) = $\frac{2}{(1-x)^2}$ + 1

⇔ f(X) = $\frac{2}{(1-x)^3}$ + $\frac{1}{1-x}$ (2)

Ta có $\sum_{n=0}^{\infty}\ (n+1)X^n$ = $\frac{1}{(1-x)^2}$

Lấy đạo hàm 2 vế, ta được:

$\sum_{n=1}^{\infty}\ n(n+1)X^{n-1}$ = $\frac{2}{(1-x)^3}$ (3)

Đặt m = n – 1

⇨ n = m + 1

Đổi biến biểu thức (3) ta được:

$\sum_{m=0}^{\infty} (m+1)(m+2)X^m = \frac{2}{(1-x)^3}$

⇔ $\sum_{n=0}^{\infty} (n+1)(n+2)X^n = \frac{2}{(1-x)^3}$

Thay vào biểu thức (2) ta được:

f(X) = $\frac{2}{(1-x)^3} + \frac{1}{1-x}$

= $\sum_{n=0}^{\infty} (n+1)(n+2)X^n + \sum_{n=0}^{\infty} X^n$

= $\sum_{n=0}^{\infty} X^n (n^2 + 3n + 3)$

Vậy T(n) = $n^2 + 3n + 3$

Bài 5:

5a:

$$\begin{cases} \boldsymbol{T(1) = C_1} \\ \boldsymbol{T(n) = 4T(n/2) + n \text{ nếu } n \geq 2} \end{cases}$$

i. Dự đoán: $f(n) = an^3$

$CM: T(n) \leq f(n)\ \forall n$

Bước 1:

Ta cần chứng minh: $T(1) \leq f(1) \Leftrightarrow C_1 \leq a.$

Nếu ta chọn a sao cho $C_1 \leq a$, ta có dpcm

Bước 2:

Giả thiết quy nạp $T(k) \leq f(k)\ \forall k < n$

Bước 3:

Chứng minh $T(n) \leq f(n)\ tại\ n$

Ta có: $T(n) = 4T\left(\frac{n}{2}\right) + n \leq 4f\left(\frac{n}{2}\right) + n$

$T(n) \leq 4a\left(\frac{n}{2}\right)^3 + n$

$\Leftrightarrow T(n) \leq \frac{1}{2}an^3 + n$

$$\Leftrightarrow T(n) \leq \frac{1}{2}f(n) + n$$

$$\Leftrightarrow T(n) \leq \frac{1}{2}f(n) + n$$

Để $T(n) \leq f(n) \Leftrightarrow \frac{1}{2}f(n) + n \leq f(n)$

$$\Leftrightarrow n \leq \frac{1}{2}f(n) \text{ } hay \text{ } n \leq \frac{1}{2}an^3$$

$$\Leftrightarrow n(an^2 - 2) \geq 0, n \geq 2$$

Chọn a phải thỏa mãn $\begin{cases} an^2 - 2 \geq 2, n \geq 2 \\ a \geq C_1 \end{cases}$. Vậy ta có thể chọn $a = 1 + |C_1|$

khi đó $T(n) \leq (1 + |C_1|)n^3$

ii. Dự đoán: $f(n) = an^2$

$CM: T(n) \leq f(n) \; \forall n$

Bước 1:

Ta cần chứng minh: $T(1) \leq f(1) \Leftrightarrow C_1 \leq a.$

Nếu ta chọn a sao cho $C_1 \leq a$, ta có dpcm

Bước 2:

Giả thiết quy nạp $T(k) \leq f(k) \; \forall \; k < n$

Bước 3:

Chứng minh $T(n) \leq f(n) \text{ } tại \text{ } n$

Ta có: $T(n) = 4T\left(\frac{n}{2}\right) + n \leq 4f\left(\frac{n}{2}\right) + n$

$$T(n) \leq 4a\left(\frac{n}{2}\right)^2 + n$$

$$\Leftrightarrow T(n) \leq an^2 + n$$

$$\Leftrightarrow T(n) \leq f(n) + n$$

$$\Leftrightarrow T(n) \leq f(n) + n$$

Vì $f(n) + n > f(n)$ nên chúng ta không thể chứng minh được dự đoán đúng

iii. Dự đoán: $f(n) = an^2 - bn$

$CM: T(n) \leq f(n) \; \forall n$

Bước 1:

Ta cần chứng minh: $T(1) \leq f(1) \Leftrightarrow C_1 \leq a.$

Nếu ta chọn a sao cho $C_1 \leq \ a - b$, ta có dpcm

Bước 2:

Giả thiết quy nạp $T(k) \leq \ f(k) \ \forall k < \ n$

Bước 3:

Chứng minh $T(n) \leq \ f(n) \ tại \ n$

Ta có: $T(n) = 4T\left(\frac{n}{2}\right) + n \leq 4f\left(\frac{n}{2}\right) + \ n$

$$T(n) \leq 4(a\left(\frac{n}{2}\right)^2 - b\left(\frac{n}{2}\right)) + \ n$$

$$\Leftrightarrow T(n) \leq an^2 - bn + (1 + \frac{b}{2})n$$

$$\Leftrightarrow T(n) \leq f(n) + (1 + \frac{b}{2})n$$

$$\Leftrightarrow T(n) \leq f(n) + n$$

$$\Leftrightarrow T(n) \leq f(n) + n$$

Nếu chọn b sao cho $1 + \frac{b}{2} \leq 0$ thì $T(n) \leq \ f(n)$

Chọn a,b phải thỏa mãn $\begin{cases} b = -2 \\ a + b \geq C_1 \end{cases}, chọn \begin{cases} b = -2 \\ a = C_1 + 2 \end{cases}$ ta xác định được $T(n) \leq (C_1 + 2)n^2 + 2n$

5b:

$$\begin{cases} \boldsymbol{1 \ khi \ n = 1} \\ \boldsymbol{T(n) = 3T\left(\frac{n}{2}\right) + n^2 \ \ nếu \ n \geq 2} \end{cases}$$

i. Dự đoán: $f(n) = an^2 + b$

$CM: T(n) \leq f(n) \ \forall n$

Bước 1:

Ta cần chứng minh: $T(1) \leq \ f(1) \Leftrightarrow 1 \leq a + b.$

Nếu ta chọn a,b sao cho $1 \leq \ a + b$, ta có dpcm

Bước 2:

Giả thiết quy nạp $T(k) \leq \ f(k) \ \forall k < \ n$

Bước 3:

Chứng minh $T(n) \leq \ f(n) \ tại \ n$

Ta có: $T(n) = 3T\left(\frac{n}{2}\right) + n^2 \leq 3f\left(\frac{n}{2}\right) + \ n^2$

$$T(n) \leq 3(a\left(\frac{n}{2}\right)^2 + b) + \ n$$

$$\Leftrightarrow T(n) \leq \frac{3}{4}an^2 + n + b$$

$$\Leftrightarrow T(n) \leq \frac{3}{4}f(n) + n + \frac{b}{4}$$

Để $T(n) \leq \ f(n) \Leftrightarrow \frac{3}{4}f(n) + n + \frac{b}{4} \leq \ f(n)$

$$\Leftrightarrow n + \frac{b}{4} \leq \frac{f(n)}{4}\ hay\ n + \frac{b}{4} \leq \frac{an^2 + b}{4}$$

$$\Leftrightarrow 4n + b \leq an^2 + b\ hay\ an^2 + 4n \geq 0, n > 1$$

$$\Leftrightarrow an + 4 \geq 0\ , n > 1$$

Chọn a phải thỏa mãn $\begin{cases} a + b \geq 1 \\ an + 4 \geq 0, n > 1 \end{cases}, chọn \begin{cases} b = 1 \\ a = \ 1 \end{cases}$. ta xác định được $T(n) \leq n^2 + 1$

5c:

$$\boldsymbol{T(n)} = \begin{cases} \boldsymbol{1\ khi\ n \leq 5} \\ \boldsymbol{T\left(\frac{n}{2}\right) + T\left(\frac{n}{4}\right)\ + n\ \ n\text{ế}u\ n > 5} \end{cases}$$

i. Dự đoán: $f(n) = an$

$CM: T(n) \leq f(n)\ \forall n$

Bước 1:

Ta cần chứng minh: $T(n \leq 5) \leq \ f(n \leq 5) \Leftrightarrow 1 \leq \ f(n \leq 5) \Leftrightarrow 1 \leq f(1)$

Nếu ta chọn a sao cho $1 \leq a$, ta có dpcm

Bước 2:

Giả thiết quy nạp $T(k) \leq \ f(k)\ \forall\ k < \ n$

Bước 3:

Chứng minh $T(n) \leq \ f(n)\ tại\ n$

Ta có: $T(n) = T\left(\frac{n}{2}\right) + T\left(\frac{n}{4}\right)\ + n \leq f\left(\frac{n}{2}\right) + f\left(\frac{n}{4}\right) + n$

$$T(n) \leq \frac{an}{2} + \frac{an}{4} + n$$

$$\Leftrightarrow T(n) \leq \frac{3}{4}an + n$$

$$\Leftrightarrow T(n) \le \frac{3}{4} f(n) + n$$

Để $T(n) \le f(n) \Leftrightarrow \frac{3}{4} f(n) + n \le f(n)$

$$\Leftrightarrow n \le \frac{f(n)}{4} \; hay \; n \le \frac{an}{4}$$

Chọn a phải thỏa mãn $\begin{cases} a \ge 1 \\ an - 4n \ge 0, n > 5 \end{cases}$, $chọn\ a = 5$. ta xác định được $T(n) \le 5n$

Bài Bonus:

2.4

Bài 1.

1. Solve the following recurrence relations.

a. $x(n) = x(n-1) + 5$ for $n > 1$, $x(1) = 0$

b. $x(n) = 3x(n-1)$ for $n > 1$, $x(1) = 4$

c. $x(n) = x(n-1) + n$ for $n > 0$, $x(0) = 0$

d. $x(n) = x(n/2) + n$ for $n > 1$, $x(1) = 1$ (solve for $n = 2^k$)

e. $x(n) = x(n/3) + 1$ for $n > 1$, $x(1) = 1$ (solve for $n = 3^k$)

Giải:

a) X(n) = X(n - 1) + 5 for n > 1
X(1) = 0

Ta có X(n) = X(n - 1) + 5
= (X(n - 2) + 5) + 5
= X(n - 2) + 2*5
= (X(n - 3) + 5) + 2*5
= X(n - 3) + 3*5
=
…
..
= X(n - i) + i*5

Quá trình dừng lại khi X(1)⇔ n – i = 1
⇔ i = n - 1
⇨ Bước cuối:
X(n) = X(1) + (n - 1)*5
= 0 + 5(n - 1)
= 5n – 5

b) X(n) = 3X(n - 1) for n > 1
X(1) = 4

Ta có X(n) = 3X(n - 1)
= 3(3X(n - 2))
= 3^2 X(n - 2)
=
…
..
= 3^i X(n - i)
Quá trình dừng lại khi X(1) ⇔ n – i = 1
⇔ i = n - 1
⇨ Bước cuối:
X(n) = 3^i X(n - i)
= 3^{n-1} X(1)
= 4 * 3^{n-1}

c) X(n) = X(n - 1) + n for n > 0
X(0) = 0

Ta có X(n) = X(n - 1) + n
= (X(n - 2) + n) + n
= X(n - 2) + 2*n
= (X(n - 3) + n) + 2*n
= X(n - 3) + 3*n
=
…
..
= X(n - i) + i*n
Quá trình dừng lại khi X(0)⇔ n – i = 0
⇔ i = n
⇨ Bước cuối:
X(n) = X(0) + (n)*n
= 0 + n^2
= n^2

d) X(n) = X(n / 2) + n for n > 1
X(1) = 1

Ta có X(n) = X(n / 2) + n
= (X(n / 4) + n) + n
= X(n / 4) + 2*n
= (X(n /8) + n) + 2*n
= X(n / 8) + 3*n
=
...
..
= X(n / 2^i) + i*n
Quá trình dừng lại khi X(1)⇔ n / 2^i= 1
⇔ n = 2^i
⇔ i = $\log_2 n$
⇨ Bước cuối:
X(n) = X(1) + ($\log_2 n$)*n
= 1 + ($\log_2 n$)*n

e) X(n) = X(n / 3) + 1 for n > 1
X(1) = 1

Ta có X(n) = X(n / 3) + 1
= (X(n / 9) + 1) + 1
= X(n / 9) + 2
= (X(n /27) + 1) + 2
= X(n / 27) + 3
=
...
..
= X(n / 3^i) + i
Quá trình dừng lại khi X(1)⇔ n / 3^i= 1
⇔ n = 3^i
⇔ i = $\log_3 n$
⇨ Bước cuối:
X(n) = X(1) + ($\log_3 n$)
= 1 + ($\log_3 n$)

Bài 2.

2. Set up and solve a recurrence relation for the number of calls made by $F(n)$, the recursive algorithm for computing $n!$.

<u>Giải:</u>

T(0) = C1
T(n) = T(n - 1) + C2, với n > 0

Ta có T(n) = T(n - 1) + C2
= (T(n - 2) + C2) + C2
= T(n - 2) + 2*C2
= (T(n - 3) + C2) + 2*C2
= T(n - 3) + 3*C2
=
...
..
= T(n - i) + i*C2
Quá trình dừng lại khi T(0)⇔ n – i = 0
⇔ i = n
⇨ Bước cuối:
T(n) = T(0) + (n)*C2
= C1 + nC2

Bài 3.

3. Consider the following recursive algorithm for computing the sum of the first n cubes: $S(n) = 1^3 + 2^3 + \cdots + n^3$.

ALGORITHM $S(n)$

//Input: A positive integer n
//Output: The sum of the first n cubes
if $n = 1$ **return** 1
else return $S(n - 1) + n * n * n$

a. Set up and solve a recurrence relation for the number of times the algorithm's basic operation is executed.

b. How does this algorithm compare with the straightforward nonrecursive algorithm for computing this sum?

Giải:

a) T(1) = C1
   T(n) = T(n - 1) + C2 khi n > 1

Ta có T(n) = T(n - 1) + C2
= (T(n - 2) + C2) + C2
= T(n - 2) + 2*C2
= (T(n - 3) + C2) + 2*C2
= T(n - 3) + 3*C2
=
…
..
= T(n - i) + i*C2
Quá trình dừng lại khi T(0)⇔ n – i = 0
⇔ i = n
⇨ Bước cuối:
   T(n) = T(0) + (n)*C2
   = C1 + nC2

b) Thuật toán giải bài toán này mà không dùng đệ quy mà dùng vòng lặp thì cũng sẽ tốn O(n) thời gian để thực thi nên có thể coi 2 cách làm có Time complexity tương đương nhau

Bài 4.

4. Consider the following recursive algorithm.

ALGORITHM $Q(n)$
//Input: A positive integer n
if $n = 1$ **return** 1
else return $Q(n-1) + 2 * n - 1$

a. Set up a recurrence relation for this function's values and solve it to determine what this algorithm computes.

b. Set up a recurrence relation for the number of multiplications made by this algorithm and solve it.

c. Set up a recurrence relation for the number of additions/subtractions made by this algorithm and solve it.

<u>Giải:</u>

a)

T(1) = C1
T(n) = T(n - 1) + C2 * n   với n > 1

Ta có T(n) = T(n - 1) + C2 * n

$= (T(n - 2) + C2 * n) + C2 * n$
$= T(n - 2) + 2C2*n$
$= (T(n - 3) + C2 * n) + 2C2 * n$
$= T(n - 3) + 3C2 * n$
= …
= …
$= T(n - i) + iC2 * n$
Quá trình dừng lại khi $T(1) \Leftrightarrow n – i = 1$
$\Leftrightarrow i = n – 1$
Bước cuối:
$T(n) = T(1) + (n - 1) C2 n$
$= C1 + (n - 1)n C2$

b) Xem thời gian nhân hay thời gian cộng, trừ đều là C2 hết => chỉ cẩn giải 1 bài là xong

$T(1) = C1$
$T(n) = T(n - 1) + C2$ với $n > 1$

Ta có $T(n) = T(n - 1) + C2$
$= (T(n - 2) + C2) + C2$
$= T(n - 2) + 2*C2$
$= (T(n - 3) + C2) + 2*C2$
$= T(n - 3) + 3*C2$
=
…
..
$= T(n - i) + i*C2$
Quá trình dừng lại khi $T(1)\Leftrightarrow n – i = 1$
$\Leftrightarrow i = n - 1$
⇨ Bước cuối:
$T(n) = T(1) + (n - 1)*C2$
$= C1 + (n - 1)C2$

Bài 5.

5. *Tower of Hanoi*

a. In the original version of the Tower of Hanoi puzzle, as it was published in the 1890s by Édouard Lucas, a French mathematician, the world will end after 64 disks have been moved from a mystical Tower of Brahma. Estimate the number of years it will take if monks could move one disk per minute. (Assume that monks do not eat, sleep, or die.)

b. How many moves are made by the ith largest disk ($1 \le i \le n$) in this algorithm?

c. Find a nonrecursive algorithm for the Tower of Hanoi puzzle and implement it in the language of your choice.

Giải:

a) Ta có phương trình truy hồi của bài toán Tháp Hà Nội là:

T(1) = 1

T(n) = 2 T(n - 1) + 1 với n > 1

(Ta chỉ quan tâm chi phí để chuyển đổi đĩa giữa các tháp)

Ta có T(n) = 2 T(n - 1) + 1

= 2(2 T(n - 2) + 1) + 1

= 4 T(n - 2) + 3

= 4 (2T(n - 3) + 1) +3

= 8 T(n - 3) + 7

= …

..

= $2^i\, T(n-i) + 2^i - 1$

Quá trình dừng lại khi tới T(1) ⇔ n – i = 1

⇔ i = n – 1

Bước cuối

T(n) = $2^{n-1}\, T(1) + 2^{n-1} - 1$

= $2^n - 1$

Vậy nếu với 64 đĩa thì ta có:

T(64) = $2^{64} - 1$ = 18446744073709600000 – 1

= 18446744073709599999

Nếu duy chuyển 1 đĩa mất 1 phút

⇨ Cần 18446744073709599999 phút

⇨ Cần 18446744073709599999 / (365 * 24 * 60) năm

= 35096545041304.4 (năm)

b) T(i) = $2^i - 1$

c) Tháp Hà Nội không đệ quy, Dùng stack, Java

```
import java.util.Stack;

public class HanoiTower {
    public static void main(String[] args) {
```

```
        int numDiscs = 3;
        Tower source = new Tower("A");
        Tower auxiliary = new Tower("B");
        Tower destination = new Tower("C");

        // Khởi tạo tháp nguồn với các đĩa
        for (int i = numDiscs; i >= 1; i--) {
            source.push(i);
        }

        // Sử dụng ngăn xếp để thực hiện di chuyển
        nonRecursiveHanoi(numDiscs, source, auxiliary, destination);
    }

    public static void nonRecursiveHanoi(int numDiscs, Tower source, Tower auxiliary, Tower
destination) {
        int totalMoves = (int) Math.pow(2, numDiscs) - 1;

        // Lặp qua tất cả các bước di chuyển
        for (int move = 1; move <= totalMoves; move++) {
            if (move % 3 == 1) {
                moveBetweenTowers(source, destination);
            } else if (move % 3 == 2) {
                moveBetweenTowers(source, auxiliary);
            } else if (move % 3 == 0) {
                moveBetweenTowers(auxiliary, destination);
            }
        }
    }

    public static void moveBetweenTowers(Tower source, Tower destination) {
        if (!source.isEmpty() && (destination.isEmpty() || source.peek() < destination.peek())) {
            int disk = source.pop();
            destination.push(disk);
            System.out.println("Move disk " + disk + " from " + source.getName() + " to " +
destination.getName());
        }
    }
}

class Tower {
    private Stack<Integer> disks;
    private String name;
```

```
    public Tower(String name) {
        this.disks = new Stack<>();
        this.name = name;
    }

    public void push(int disk) {
        disks.push(disk);
    }

    public int pop() {
        return disks.pop();
    }

    public int peek() {
        return disks.peek();
    }

    public boolean isEmpty() {
        return disks.isEmpty();
    }

    public String getName() {
        return name;
    }
}
```
(Code do ChatGpt gen)

2.5

Bài 1.

1. Find a Web site dedicated to applications of the Fibonacci numbers and study it.

Xong

https://www.investopedia.com/terms/f/fibonaccilines.asp

Bài 2.

2. *Fibonacci's rabbits problem* A man put a pair of rabbits in a place surrounded by a wall. How many pairs of rabbits will be there in a year if the initial pair of rabbits (male and female) are newborn and all rabbit pairs are not fertile during their first month of life but thereafter give birth to one new male/female pair at the end of every month?

Giải:

Gọi F(n) là số cặp thỏ trong chuồng ở thời điểm n, ta có

$$T(0) = T(1)=1$$

$$T(n) = T(n - 1) + T(n - 2)$$

Theo như câu 3c thì ta có:

T(n) = $\frac{\sqrt{5}+1}{2\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^{n} + \frac{\sqrt{5}-1}{2\sqrt{5}}\left(\frac{1-\sqrt{5}}{2}\right)^{n}$

Với n = 12 thì ta có:

T(12) = $\frac{\sqrt{5}+1}{2\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^{12} + \frac{\sqrt{5}-1}{2\sqrt{5}}\left(\frac{1-\sqrt{5}}{2}\right)^{12}$

Bài 3:

3. *Climbing stairs* Find the number of different ways to climb an *n*-stair staircase if each step is either one or two stairs. For example, a 3-stair staircase can be climbed three ways: 1-1-1, 1-2, and 2-1.

Giải:

Đây chính là bài toán Fibonacci. Để cần tính số cách để leo lên bậc thang thứ n thì đầu tiên ta cần tính số cách để leo lên bậc thang thứ n – 1 và bậc thang thứ n – 2 rồi lấy 2 kết quả cộng lại với nhau là ra kết quả bài toán.

Gọi T(n) là số cách để leo bậc thang cấp n

Ta có

$$T(0) = T(1)=1$$

$$T(n) = T(n - 1) + T(n - 2)$$

Theo như câu 3c thì ta có:

T(n) = $\frac{\sqrt{5}+1}{2\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^{n} + \frac{\sqrt{5}-1}{2\sqrt{5}}\left(\frac{1-\sqrt{5}}{2}\right)^{n}$

Bài 4:

4. How many even numbers are there among the first n Fibonacci numbers, i.e., among the numbers $F(0), F(1), \ldots, F(n-1)$? Give a closed-form formula valid for every $n > 0$.

<u>Giải:</u>

Ta có 2 số lẻ thì ghép lại làm 1 số chẵn

1 lẻ 1 chẵn thì ghép lại được 1 số lẻ

- ⇨ Cứ mỗi nhóm gồm 3 số fibonacci liên tục sẽ có 1 số chẵn
- ⇨ Chỉ cần tính có bao nhiêu số fibonacci trong dãy rồi chia 3 là ra kết quả bài toán
- ⇨ Trong n số fibonacci ban đầu sẽ có $\lfloor n / 3 \rfloor$ số chẵn

Bài 6:

6. The maximum values of the Java primitive types `int` and `long` are $2^{31} - 1$ and $2^{63} - 1$, respectively. Find the smallest n for which the nth Fibonacci number is not going to fit in a memory allocated for

a. the type `int`. **b.** the type `long`.

<u>Giải:</u>

Ta sẽ áp dụng công thức Tỷ lệ vàng để tính số Fibinacci thứ n

$F(n) = (\varphi^n - (-\varphi)^(-n)) / \sqrt{5}$

a) F(n) > 2^31 – 1

⇔ $(\varphi^n - (-\varphi)^(-n)) / \sqrt{5} > 2^{31} - 1$ với $\varphi \approx 1.618$

- ⇨ n = 47 (code Python để tính)

b) F(n) > 2^63 – 1

⇔ $(\varphi^n - (-\varphi)^(-n)) / \sqrt{5} > 2^{61} - 1$ với $\varphi \approx 1.618$

- ⇨ n = 93 (code Python để tính)