CẤU TRÚC DỮ LIỆU

Kim Minh Thắng - B2007210

Bài tập 1: Tính độ phức tạp của đoạn chương trình sau:

```
1 Sum1 = 0;
2 k = 1;
3 while (k ⩽ n){
4     for (j = 1; j ⩽ n; j++)
5         Sum1 = Sum1 + 1;
6     k = k * 2;
7 }
```

Bài làm:

Dòng 1: phép gán ⇒ độ phức tạp là O(1)	
Dòng 2: phép gán ⇒ độ phức tạp là O(1)	
⇒ Độ phức tạp của dòng 1 và dòng 2 là 1 (Quy tắc cộng)	*(1)*
Dòng 4 và 5: vòng lặp thực hiện n lần mỗi lần O(1) ⇒ độ phức tạp là O(n)	
Dòng 3 đến 7: vòng lặp dừng khi $2^k > n \Leftrightarrow k > log(n)$ ⇒ độ phức tạp là $O(n) * O(log(n)) = O(n * log(n))$	*(2)*
Từ *(1) và (2):* ⇒ độ phức tạp $T(n) = max(O(1), O(n * log(n))) = O(n * log(n))$	

Vậy: độ phức tạp của giải thuật là $T(n) = O(n * log(n))$.

Bài tập 2: Hãy viết hàm tính tổng sau.

$$\sum_{i=0}^{n} r^i = r^0 + r^1 + r^2 + r^3 + ... + r^n$$

Hãy tính độ phức tạp của hàm? Bạn có thể cải tiến hàm đã viết?

Bài làm:

Viết hàm: [Source code]

```
1  double sum(double r, int n)
2  {
3      double s, power;
4      s = 0;
5      for (int i = 0; i <= n; i++)
6      {
7          power = 1;
8          for (int j = 1; j <= i; j++)
9              power *= r;
10         s += power;
11     }
12     return s;
13 }
```

Tính độ phức tạp:

Dòng 3, 4: Độ phức tạp: $O(1)$	*(1)*
Dòng 8, 9: độ phức tạp là $O(i) * O(1) = O(i)$với i tăng từ 0 đến n ⇒ độ phức tạp theo quy tắc cộng là $O(n)$	
Dòng 7 đến dòng 9: độ phức tạp là $O(1) + O(n) + O(1) = O(n)$	
Dòng 5 đến dòng 11: độ phức tạp là $O(n) * O(n) = O(n * n) = O(n^2)$	*(2)*
Dòng 12: độ phức tạp là $O(1)$	*(3)*
Từ (1), (2) và (3): ⇒ Độ phức tạp $T(n) = O(1) + O(n^2) + O(1) = O(n^2)$	

Vậy: độ phức tạp của giải thuật là: $T(n) = O(n^2)$

Có thể cải tiến giải thuật bằng công thức tính tổng cấp số nhân:

- Số hạng đầu: $u_1 = r^0 = 1$
- Công bội: $q = r$

$$\Rightarrow \sum_{i=0}^{n} r^i = r^0 + r^1 + r^2 + r^3 + ... + r^n = \frac{1(r^{n+1} - 1)}{r - 1} = \frac{r^{n+1} - 1}{r - 1}$$

Hàm sau khi đã cải tiến giải thuật: [Source code]

```
double sum(double r, int n)
{
    double power = 1;
    for (int i = 1; i <= n + 1; i++)
        power *= r;
    return (power - 1) / (r - 1);
}
```

Độ phức tạp là: $T(n) = O(n + 1) = O(n)$.

Bài tập 3: hãy viết hàm tính e^x theo công thức gần đúng sau với x và n là tham số đầu vào.

$$e^x = \sum_{i=0}^{n} \frac{x^i}{i!} = 1 + \frac{x}{1!} + \frac{x^2}{2!} + ... + \frac{x^n}{n!}$$

Hãy tính độ phức tạp của hàm? Bạn có thể cải tiến hàm đã viết?

Bài làm:

Viết hàm: [Source code]

```
1  double power(double x, int n);
2  long long factorial(int n);
3
4  // hàm tính e^x
5  double exp(double x, int n)
6  {
7      double e = 1;
8      for (int i = 1; i <= n; i++)
9          e += (double)power(x, i) / factorial(i);
10     return e;
11 }
12
13 // hàm tính lũy thừa và giai thừa
14 double power(double x, int n)
15 {
16     double p = 1;
17     for (int i = 1; i <= n; i++)
18     {
19         p *= x;
20     }
21     return p;
22 }
23 long long factorial(int n)
24 {
25     long long f = 1;
26     for (int i = 2; i <= n; i++)
27         f *= i;
28     return f;
29 }
```

Tính độ phức tạp:

Hàm power: có độ phức tạp là $O(n)$	
Hàm factorial: có độ phức tạp là $O(n)$	
Dòng 7: có độ phức tạp là $O(1)$	*(1)*

Dòng 9: có độ phức tạp là $O(i) + O(i) = O(i)$với i từ 1 đến n ⇒độ phức tạp theo quy tắc cộng là $O(n)$	
Dòng 8, 9: có độ phức tạp là $O(n) * O(n) = O(n * n) = O(n^2)$	*(2)*
Dòng 10: có độ phức tạp là $O(1)$	*(3)*
Từ (1), (2) và (3): ⇒Độ phức tạp: $T(n) = O(1) + O(n^2) + O(1) = O(n^2)$	

Vậy: độ phức tạp của giải thuật là: $O(n^2)$.

Có thể cải tiến giải thuật bằng cách:

$$e^x = \sum_{i=0}^{n} \frac{x^i}{i!} = \underset{a0}{1} + \underset{a1}{\frac{x}{1!}} + \underset{a2}{\frac{x^2}{2!}} + \ldots + \underset{an}{\frac{x^n}{n!}}$$

Ta dễ thấy

$$a_i = a_{i-1} * \frac{x}{i}$$

Hàm sau khi đã cải tiến giải thuật: [Source code]

```
double exp(double x, int n)
{
    double e = 1, //giá trị khởi tạo ứng với n == 0
        a = 1;    //a_0
    for (int i = 1; i <= n; i++)
    {
        a *= x / i; //a_i = a_(i-1) * x/i
        e += a;
    }
    return e;
}
```

Độ phức tạp của giải thuật này là:$T(n) = O(n)$

Bài tập 4: để giải một bài toán thực tế, ta có thể sử dụng một trong hai giải thuật A hoặc B.

- Giải thuật A có $TA(n) = 100 * n * n\, ms = O(n^2)$

- Giải thuật B có $TB(n) = 5 * n * n * n\, ms = O(n^3)$

Ta nên chọn giải thuật A hay giải thuật B?

Bài làm:

Ta chọn giải thuật A vì nó có độ phức tạp nhỏ hơn giải thuật B.