47003 - Algoritmos e Complexidade

Universidade de Aveiro

António Rafael da Costa Ferreira
NMec: 67405 | P1
rafael.ferreira@ua.pt

Rodrigo Lopes da Cunha
NMec: 67800 | P1
rodrigocunha@ua.pt

Pretendemos com este trabalho analisar e comparar o desempenho computacional de duas estratégias distintas para o cálculo dos números de Catalan com base na relação de recorrência seguinte:

$$\text{Catalan}(n) = \begin{cases} 1, & \text{se } n = 0 \\ \sum_{i=0}^{n-1} \text{Catalan}(i) \times \text{Catalan}(n-i-1), & \text{se } n > 0 \end{cases}$$

1. Funções de Catalan desenvolvidas:

Função recursiva:

```
int catalanRecursive(unsigned int n){
    int i;
    int sum = 0;

    if (n == 0) {
        return 1;
    }else{
        for(i = 0; i < n; i++){
            nMults++;
            sum += catalanRecursive(i) * catalanRecursive(n - i - 1);
        }
    }

    return sum;
}
```

Função repetitiva:

```
int catalanDinamic(unsigned int n){
    int i, j;
    int array[n];

    array[0] = 1;
    for(i = 1; i <= n; i++){
        array[i] = 0;
        for(j = 0; j < i; j++){
            nMults++;
            array[i] += array[j] * array[i - j - 1];
        }
    }
    return array[n];
}
```

2. Explicação da estratégia de programação dinâmica usada:

A estratégia de programação dinâmica é usada no armazenamento dos números de Catalan desde 0 até n num array de inteiros. O primeiro valor, número de Catalan de 0, é sempre 1, portanto é logo inicializado como ***array[0] = 1***.

Os números de Catalan são calculados de 1 até n, baixo para cima, e colocados no ***array[]***, sendo devolvido o número de Catalan n, ***array[n]*** (o número pretendido).

3. Análise formal das funções recursiva e repetitiva:

Sendo ***M(n)*** o número de multiplicações que as funções têm de efetuar para calcular o n-ésimo número de Catalan.

Função recursiva:

Analisando a função recursiva verificamos que a cada iteração no ciclo for, existem duas chamadas recursivas e uma multiplicação, concluindo-se assim, que o número de multiplicações pode ser obtido por:

$$M(n) = \begin{cases} 0, & \text{se } n=0 \\ \sum_{i=0}^{n-1} (M(i) + M(n-i-1) + 1), & \text{se } n>0 \end{cases}$$

Sabendo que:

$$\sum_{i=0}^{n-1} M(i) = \sum_{i=0}^{n-1} M(n-i-1)$$

Temos que:

$$M(n) = 2\sum_{i=0}^{n-1} M(i) + n$$

Comparando o número de multiplicações de *n* com *n-1* temos que:

$$M(n) - M(n-1) = 2\sum_{i=0}^{n-1} M(i) + n - \left[2\sum_{i=0}^{n-2} M(i) + (n-1)\right] = 2M(n-1) + 1$$

$$
\begin{aligned}
M(n) &= 3M(n-1) + 1 \\
&= 3[3M(n-2) + 1] + 1 = 3^2M(n-2) + 3 + 1 \\
&= \ldots
\end{aligned}
$$

$$= 3^kM(n-k) + 3^{k-1} + 3^{k-2} + 3^{k-3} + \ldots + 3 + 3^0 = 3^kM(n-k) + \sum_{i=0}^{k-1} 3^i$$

terminando, quando n = k, M(0);

A complexidade da função é exponencial, O(3^n) e o número de multiplicações efetuadas pode ser dado por:

$$M(n) = \frac{1}{2}(3^n - 1)$$

Função repetitiva:

Analisando a função sabemos que as multiplicações estão dentro de dois ciclos ***for***, sendo que o primeiro vai desde ***i=1*** até ***i=n*** e o segundo vai desde ***j=0*** até ***j=i-1***. portanto número de mutlicações de n será:

$$\sum_{i=1}^{n} \sum_{j=0}^{i-1} 1 = \sum_{i=1}^{n} i = \frac{n(n+1)}{2} = \frac{1}{2}(n^2+n)$$

Portanto, a complexidade desta função repetitiva é de O(n^2), complexidade quadrática.

4. Análise do número de multiplicações efetuadas pelas funções anteriores:

Na seguinte tabela estão os resultados obtidos experimentalmente:

n	Catalan(n)	M(n)	
		Recursiva	Dinâmica
0	1	0	0
1	1	1	1
2	2	4	3
3	5	13	6
4	14	40	10
5	42	121	15
6	132	364	21
7	429	1093	28
8	1430	3280	36
9	4862	9841	45
10	16796	29524	55

Comparação dos resultados obtidos:

Analisando a tabela sabemos que a função recursiva cresce de forma exponencial O(3^n) (complexidade exponencial) e a repetitiva de forma quadrática O(n^2) (complexidade quadrática).

Podemos observar que os resultados obtidos na análise computacional são idênticos aos da análise formal.

Verifica-se também, que a função repetitiva é muito mais eficiente do que a função recursiva.

Referências:

[1] A. Adrego da Rocha, "Análise da Complexidade de Algoritmos" FCA, 2014
[2] A. Adrego da Rocha, *"Estruturas de Dados e Algoritmos em C"* FCA, 2008
[3] Apontamentos das aulas práticas