

Exercícios: Recursão - Recorrências

Ronaldo Hashimoto

March 28, 2016

1. Considere a função abaixo que calcula o n -ésimo termo $F(n)$ da sequência de Fibonacci:

```
1 int fibonacci(int n) {  
2     printf("*");  
3     if (n == 1) || (n == 2)  
4         return 1;  
5     return (fibonacci(n-1) + fibonacci(n-2));  
6 }
```

- Se $T(n)$ denota a quantidade de asteriscos impressos no cálculo de `fibonacci(n)`, escreva a equação de recorrência para $T(n)$.
- Dado n , prove (por indução em n) que a quantidade de asteriscos ‘*’ impressos no cálculo de `fibonacci(n)` é $T(n) = 2 \times F(n) - 1$.

2. Considere a função abaixo:

```
1 int X (int n) {  
2     if ( n <= 0 )  
3         return 0;  
4     return n + X(n-1);  
5 }
```

- O que faz esta função?
- Se $T(n)$ denota o número de operações de adição ‘+’ feito pela chamada $X(n)$, escreva a fórmula de recorrência para $T(n)$.
- Prove (por indução) que $T(n) = n$.
- Escreva uma função não-recursiva que resolve o mesmo problema. Qual é a ordem de complexidade de tempo de sua função? Explique.

3. Considere a função abaixo:

```

1 double f(double x, double y) {
2     if (x >= y)
3         return (x+y)/2;
4     return f(f(x+2, y-1), f(x+1, y-2));
5 }

```

- Qual é o valor de $f(1, 10)$?
- Como se poderia calcular $f(a, b)$ de maneira mais simples?

4. Considere o programa abaixo:

```

1 #include <stdio.h>
2 int fusc(int n) {
3     printf ("*");
4     if (n <= 1) return (1);
5     if (n % 2 == 0)
6         return fusc(n / 2) ;
7     return fusc((n-1)/2) + fusc((n+1)/2);
8 }
9
10 int main() {
11     int m = 7;
12     printf("Fusc = %d\n", fusc(m));
13     return 0;
14 }

```

- Simule a execução do programa.
- Se $T(n)$ denota o número de de asteriscos impressos na chamada $fusc(n)$, escreva a fórmula de recorrência para $T(n)$.

5. Considere a seguinte função:

```

1 void misterio (int A[], int inic, int fim) {
2     int aux;
3     while (A[fim] % 2 == 0 && inic < fim) fim--;
4     while (A[inic] % 2 == 1 && inic < fim) inic++;
5     if (inic < fim) {
6         aux = A[inic];
7         A[inic] = A[fim];
8         A[fim] = aux;
9         misterio(A, inic, fim);
10    }
11 }

```

- Simule a função `misterio` para

$A =$

0	1	2	3	4	5	6	7	8
8	10	3	6	5	2	9	1	4

 Início= 0 e Fim = 8.

- O que faz a função `misterio`?
- Quantas comparações envolvendo elementos do vetor A são feitas? Justifique a sua resposta.
- Escreva um algoritmo que faz a mesma coisa com um número menor de comparações.