

MC 102 - Algoritmos e Programação de Computadores

Primeiro Semestre de 2014

Lista de Exercícios 6

1. Escreva um algoritmo iterativo em C para avaliar $a * b$ usando a adição, onde a e b são inteiros não negativos.
2. Escreva um algoritmo recursivo para o calculo de $a*b$, onde a e b são inteiros não-negativos.
3. Escreva uma definição recursiva para computar $a+b$, onde a e b são inteiros não-negativos.
4. Faça uma função recursiva em C que calcula o elemento máximo em um vetor.
5. Faça uma função recursiva em C que calcula o elemento mínimo em um vetor.
6. Faça uma função recursiva em C que calcula a média dos elementos de um vetor.
7. Faça uma representação da memória do computador considerando as chamadas das funções recursivas abaixo (vistas em aula). Faça um modelo passo a passo como nos exemplos visto em sala de aula:
 - `fatorial(6)`
 - `fibonacci(5)`
8. Determine o que a seguinte definição recursiva para uma função f calcula. A definição da função f é dada abaixo:
 - Se $n == 0$ retorne 0.
 - Se $n > 0$ retorne $n + f(n - 1)$.
9. Execute a função ff abaixo com os argumentos 7 e 0.

```
int ff(int n, int ind) {
    int i;
    for (i = 0; i < ind; i++)
        printf(" ");
    printf("ff (%d, %d) \n", n, ind);
    if (n == 1)
        return 1;
    if (n % 2 == 0)
        return ff(n/2, ind + 1);
    return ff((n-1)/2, ind + 1) + ff((n+1)/2, ind + 1);
}
```

10. Escreva uma função recursiva que calcule $\lfloor \lg n \rfloor$, ou seja, o *piso* do logaritmo de n na base 2.

11. Escreva uma função recursiva para a busca sequencial em um vetor.
12. Escreva uma função recursiva para a busca binária em um vetor.
13. Escreva uma função recursiva para o cálculo do máximo divisor comum de dois números inteiros não negativos. Lembre-se que
 - $\text{mdc}(x,y) = x$ se y é zero,
 - $\text{mdc}(x,y) = \text{mdc}(y,x\%y)$ caso contrário.

14. Faça uma função recursiva para calcular $\binom{n}{k}$ sabendo que

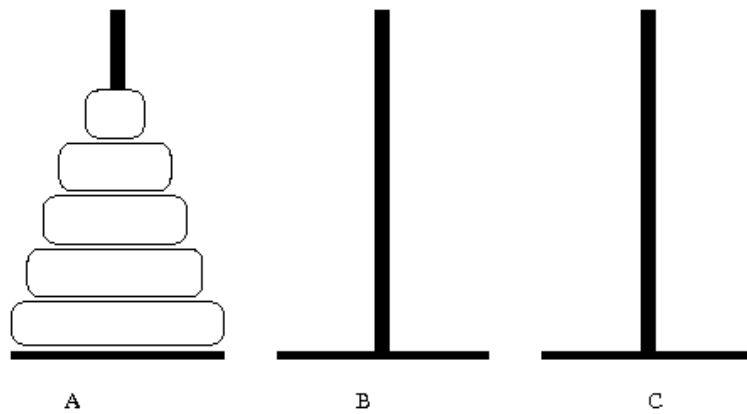
$$\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1},$$

e

$$\binom{n}{n} = 1 \text{ e } \binom{n}{1} = n.$$

15. Aplique o algoritmo de particionamento do quickSort sobre o vetor (13, 19, 9, 5, 12, 21, 7, 4, 11, 2, 6, 6) com pivô igual a 6.
16. Qual o valor retornado pelo algoritmo de particionamento se todos os elementos do vetor tiverem valores iguais?
17. Faça uma execução passo-a-passo do quickSort com o vetor (4, 3, 6, 7, 9, 10, 5, 8).
18. Modifique o algoritmo quickSort para ordenar vetores em ordem decrescente.
19. Mostre passo a passo a execução da função merge considerando dois sub-vetores: (3, 5, 7, 10, 11, 12) e (4, 6, 8, 9, 11, 13, 14).
20. Faça uma execução passo-a-passo do Merge-Sort para o vetor: (30, 45, 21, 20, 6, 715, 100, 65, 33).
21. Re-escreva o algoritmo Merge-Sort para que este passe a ordenar um vetor em ordem decrescente.
22. Considere o seguinte problema: Temos como entrada um vetor de inteiros v (não necessariamente ordenado), e um inteiro x . Desenvolva um algoritmo que determina se há dois números em v cuja soma seja x . Tente fazer o algoritmo o mais eficiente possível. Utilize um dos algoritmos de ordenação na sua solução.

23. Torres de Hanoi: Inicialmente temos 5 discos de diâmetros diferentes na estaca A.



O problema das torres de Hanoi consiste em transferir os cinco discos da estaca A para a estaca C (pode-se usar a estaca B como auxiliar). Porém deve-se respeitar algumas regras:

- Apenas o disco do topo de uma estaca pode ser movido.
- Nunca um disco de diâmetro maior pode ficar sobre um disco de diâmetro menor.

Desenvolva uma solução recursiva para o problema das Torres de Hanoi.