

Universidade Federal do Ceará - Campus Quixadá
QXD0010 – Estruturas de Dados – Turma 05A – 2019.1
Prof. Atílio Gomes

Recursividade

1. Crie um programa em C++ que receba um vetor de números reais com n elementos. Escreva uma função recursiva que inverta a ordem dos elementos presentes no vetor.
2. Faça uma função recursiva que calcule e retorne o fatorial de um número inteiro n .
3. A função fatorial duplo é definida como o produto de todos os números naturais de 1 até algum número natural ímpar n . Assim, o fatorial duplo de 5 é: $5!! = 1 * 3 * 5 = 15$. Faça uma função que receba um número inteiro positivo ímpar n e retorne o fatorial duplo desse número.
4. Faça uma função recursiva que calcule e retorne o n -ésimo termo da sequência Fibonacci. Alguns números desta sequência são: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, ...
5. Faça uma função recursiva que inverta uma string de tamanho n . Ex.: *casa* \implies *asac*.
6. Faça uma função recursiva que receba um número inteiro positivo n e imprima todos os números naturais de 0 até n em ordem decrescente.
7. Faça uma função recursiva que receba um número inteiro positivo n e imprima todos os números pares de 0 até n em ordem decrescente.
8. Faça uma função recursiva que receba um número inteiro positivo n e imprima todos os números primos de 0 até n em ordem decrescente.
9. Crie um programa em C++, que contenha uma função recursiva para encontrar o menor elemento em um vetor. A leitura dos elementos do vetor e impressão do menor elemento devem ser feitas no programa principal.
10. Dado um número n na base decimal, escreva uma função recursiva que converte este número para binário.
11. Crie um programa em C++, que contenha uma função recursiva que receba dois inteiros positivos k e n e calcule k^n . Utilize apenas multiplicações. O programa principal deve solicitar ao usuário os valores de k e n e imprimir o resultado da chamada da função.
12. O máximo divisor comum dos inteiros x e y é o maior inteiro que é divisível por x e y . Escreva uma função recursiva `mdc`, que retorna o máximo divisor comum de x e y . O `mdc` de x e y é definido como segue: se y é igual a 0, então `mdc(x, y)` é x ; caso contrário, `mdc(x, y)` é `mdc(y, x % y)`, onde `%` é o operador resto.
13. A multiplicação de dois números inteiros pode ser feita através de somas sucessivas. Implemente um algoritmo recursivo `multRec(n1, n2)` que calcule a multiplicação de dois inteiros.
14. A multiplicação à Russa consiste em:

1. Escrever os números A e B, que se deseja multiplicar na parte superior das colunas.
2. Dividir A por 2, sucessivamente, ignorando o resto até chegar à unidade, escrever os resultados da coluna A.
3. Multiplicar B por 2 tantas vezes quantas se haja dividido A por 2, escrever os resultados sucessivos na coluna B.
4. Somar todos os números da coluna B que estejam ao lado de um número ímpar da coluna A.

Exemplo: 27×82

A	B	Parcelas
27	82	82
13	164	164
6	328	-
3	656	656
1	1312	1312

Soma: 2214

Escreva uma função recursiva que permita fazer a multiplicação à russa de duas entradas.

15. Implemente uma função recursiva para resolver o seguinte problema: dado um vetor ordenado com n números inteiros positivos e um outro número inteiro positivo x , determine se existem ou não dois elementos cuja soma é igual a x .
16. Escreva uma função recursiva que calcule a soma dos dígitos de um número inteiro. Por exemplo, se a entrada for 357, a saída deverá ser $3+5+7 = 15$.
17. Faça uma função recursiva que permita somar os elementos de um vetor de inteiros.
18. Escreva uma função recursiva que recebe como entrada uma string e um caractere c e retorna a quantidade de vezes que c apareceu na string.
19. Faça uma função recursiva que recebe um vetor ordenado e um inteiro x e retorna 1 se x pertencer ao vetor e 0 caso contrário.

Seu programa deve ter complexidade da ordem de $\log n$.

Noções de Análise de Complexidade

20. Considere o seguinte Algoritmo 1 que apresenta a função SOMA-MATRIZES, que calcula a matriz quadrada C de dimensão $n \times n$ que é a soma de duas matrizes A e B ambas de mesma dimensão $n \times n$. Determine a função $f(n)$ que dá o número de passos que esse algoritmo executa em função do parâmetro n . Qual a complexidade deste algoritmo? Justifique a sua resposta?

Algoritmo 1 Soma de duas matrizes quadradas

```
1: Função SOMA-MATRIZES(int *A, int *B, int *C, int n)
2:   para  $i \leftarrow 0$  até  $n - 1$  faça
3:     para  $i \leftarrow 0$  até  $n - 1$  faça
4:        $C[i][j] \leftarrow A[i][j] + B[i][j]$ 
5:     end para
6:   end para
7:   Retorne  $C$ 
8: end Função
```

21. Considere o Algoritmo 2 que multiplica duas matrizes quadradas A e B . Determine a função $f(n)$ que dá o número de passos que esse algoritmo executa em função do parâmetro n . Qual a complexidade deste algoritmo? Justifique a sua resposta?

Algoritmo 2 Multiplicação de duas matrizes quadradas

```
1: Função MULTIPLICA-MATRIZES(int *A, int *B, int *C, int n)
2:   para  $i \leftarrow 0$  até  $n - 1$  faça
3:     para  $i \leftarrow 0$  até  $n - 1$  faça
4:        $C[i][j] \leftarrow 0$ 
5:       para  $i \leftarrow 0$  até  $n - 1$  faça
6:          $C[i][j] \leftarrow C[i][j] + A[i][j] \times B[i][j]$ 
7:       end para
8:     end para
9:   end para
10:  Retorne  $C$ 
11: end Função
```

22. Sejam as funções de complexidade $a(n) = n^2 - n + 549$ e $b(n) = 49n + 49$ referentes a certos algoritmos A e B , respectivamente. Para que valores de n é melhor aplicar o Algoritmo A ?
23. O que é a complexidade de pior caso de uma algoritmo? E o que é a complexidade de melhor caso? Qual a diferença entre elas?
24. Faça um algoritmo que verifique se os elementos de um vetor estão ordenados de forma ascendente. Qual a complexidade de pior caso e melhor caso do seu algoritmo? Justifique suas respostas.

25. Para cada uma das afirmações abaixo, justifique formalmente (usando definições, manipulações algébricas e implicações) se for verdade ou dê um contraexemplo se for falso.
- (a) $3n = O(n)$
 - (b) $2n^2 - n = O(n^2)$
 - (c) $\log 8n = O(\log 2n)$
 - (d) $2^{n+1} = O(2^n)$
 - (e) Se $f(n) = 17$, então $f(n) = O(1)$
 - (f) Se $f(n) = 3n^2 - n + 4$, então $f(n) = O(n^2)$
26. O que significa dizer que um algoritmo executa em tempo proporcional a n ?
27. Por muitas vezes damos atenção apenas à análise do pior caso dos algoritmos. Explique o porquê.
28. Qual algoritmo você prefere: um algoritmo que requer n^5 passos ou um que requer 2^n passos? Justifique sua resposta.
29. O que significa dizer que $g(n)$ é $O(f(n))$?

Tipos Abstratos de Dados

30. Dê o conceito de:
- (a) algoritmo;
 - (b) tipos de dados;
 - (c) tipo abstrato de dados;
 - (d) estrutura de dados;
 - (e) linguagem de programação;
 - (f) programa.
31. Defina um Tipo Abstrato de Dados (TAD) que represente pontos2D. Implemente, na linguagem de programação C++, um módulo que representa o TAD pontos2D e faça um pequeno programa `main.cpp` para testar sua implementação.
32. Defina um Tipo Abstrato de Dados (TAD) que represente pontos3D. Implemente, na linguagem de programação C++, um módulo que representa o TAD pontos3D e faça um pequeno programa `main.cpp` para testar sua implementação.

Listas Sequenciais (Vetores)

33. Faça uma função que encontre o valor mínimo em uma lista sequencial.
34. Escreva uma função que remova de uma lista sequencial o elemento que tem o valor máximo.
35. Escreva uma função que inclui um elemento em uma lista sequencial.
36. Escreva uma função que remove um elemento de uma lista sequencial.
37. Escreva uma função que busca a primeira ocorrência de um elemento x em uma lista sequencial e retorna a posição do elemento na lista caso ele ocorra, ou retorne -1 caso contrário.
38. Escreva algoritmos de inserção e remoção em uma lista sequencial cujos elementos estão ordenados em ordem crescente e devem permanecer ordenados após estas operações. Qual a complexidade dos seus algoritmos? Justifique.
39. Faça uma função para remover de uma lista sequencial todos os elementos com valor x , dado como entrada.
40. Escreva uma função que retorna a quantidade de vezes que o elemento x aparece na lista sequencial.

Listas Encadeadas

41. Faça uma função que encontre uma nó de conteúdo mínimo em uma lista encadeada.
42. Escreva uma função que remova de uma lista encadeada um nó cujo conteúdo tem o valor máximo.
43. Escreva uma função que conte o número de nós de uma lista encadeada.
44. Escreva uma função que concatene duas listas encadeadas (isto é, “amarre” a segunda no fim da primeira).
45. Escreva uma função que remova o k -ésimo nó de uma lista encadeada.
46. Faça uma função para remover de uma lista encadeada todos os elementos que contém x .
47. Escreva uma função para remover elementos repetidos de uma lista encadeada.
48. Escreva uma função que retorna a quantidade de vezes que o elemento x aparece na lista l .
49. Escreva uma função que conta retorna a quantidade de números primos em uma lista l .
50. Escreva uma função que receba uma lista como parâmetro e devolve a soma dos números primos dessa lista.

51. Faça uma função que receba duas listas encadeadas de mesmo tamanho l_1 e l_2 e retorna uma terceira lista de modo que cada elemento i da terceira lista é a soma do elemento i de l_1 com o elemento i de l_2 .
52. Escreva uma função que decida se duas listas dadas tem o mesmo conteúdo (Os elementos de uma lista devem ser iguais aos da outra lista, na mesma ordem).
53. Escreva uma função que inverta a ordem dos nós de uma lista encadeada (o primeiro passe a ser o último, o segundo passe a ser o penúltimo etc.). Faça isso sem criar novos nós; apenas altere os ponteiros.