

# Lista de Exercícios 2

## projeto de algoritmos



AE22CP

Prof. Jefferson T. Oliva



## 1 Método Guloso

1. Dado um grande número de arquivos digitais em um computador. Cada arquivo ocupa um certo número de KB. Deseja-se armazenar a maior quantidade de arquivos em uma mídia com capacidade  $C$ . O problema pode ser modelado na seguinte maneira: dados números naturais representando o tamanho dos arquivos  $A = \{a_1, a_2, \dots, a_n\}$  e um valor  $C$ . Deseja-se encontrar o maior subconjunto  $S$  que satisfaça a restrição  $\max \sum_i a_i \leq C$ . Em seguida, faça a análise de complexidade da solução.
2. Dado um conjunto de livros numerados de 1 a  $n$ , sendo que o  $i$ -ésimo livro tem peso  $0 < p[i] \leq 10$ . Implemente um algoritmo guloso para definir a quantidade mínima de envelopes seguindo as seguintes restrições: cada envelope deve ter  $\leq 2$  livros e o peso do conteúdo de cada envelope deve ser  $\leq 10$ . Em seguida, faça a análise de complexidade da solução.
3. Dado conjunto de objetos  $S = \{a, b, c, d, e, f, g, h, i, j\}$ , sendo cada com um determinado peso e valor  $(p, v)$  e uma mochila com capacidade máxima de peso 22. Para cada objeto, o peso e valor foram definidos da seguinte forma:  $a : (14, 7)$ ,  $b : (3, 6)$ ,  $c : (18, 6)$ ,  $d : (5, 5)$ ,  $e : (8, 24)$ ,  $f : (6, 10)$ ,  $g : (10, 25)$ ,  $h : (4, 16)$ ,  $i : (2, 7)$ ,  $j : (5, 14)$ . Considerando o **problema da mochila fracionária**, qual é a solução ótima (melhor/menor custo benefício) para o armazenamento de objetos? Mostre o desenvolvimento da solução utilizando um dos métodos de construção de algoritmos vistos em sala de aula (não é necessária a implementação de código).

## 2 Divisão e Conquista

4. Descreva diferenças entre método guloso e divisão e conquista.
5. Implemente uma função, utilizando divisão e conquista, que receba um número natural  $n$  e retorne a soma de 0 até  $n$ . Por exemplo: para  $n = 5$ , o resultado é  $0 + 1 + 2 + 3 + 4 + 5 = 15$ . A estratégia de divisão e conquista é eficiente para o problema deste exercício? Explique. **Dica (implementação e análise de complexidade):** sequência de Fibonacci.
6. Aplique a estratégia da divisão e conquista para calcular a soma dos elementos de um vetor de inteiros. Em seguida, calcule a complexidade de tempo do algoritmo.
7. A multiplicação entre números inteiros  $a$  e  $b$  pode ser dada por sucessivas somas. Por exemplo:  $3 * 5 = 3 + 3 + 3 + 3 + 3$ . Implemente um algoritmo por divisão e conquista para realização de somas sucessivas.

### 3 Programação Dinâmica

8. Qual a principal diferença entre divisão e conquista e programação dinâmica.
9. Implemente uma solução baseada em programação dinâmica para a solução do problema de potenciação: dado dois números inteiros ( $a$  e  $n$ ), calcular  $2^n$ . Dica: o problema pode ser resolvido com base na solução apresentada em aula para a sequência de Fibonacci.
10. Dadas as seguintes matrizes que devam ser multiplicadas:  $M_1[5, 10]$ ,  $M_2[10, 3]$ ,  $M_3[3, 12]$ ,  $M_4[12, 5]$ ,  $M_5[5, 50]$  e  $M_6[50, 6]$ . Encontre, utilizando a solução por programação dinâmica apresentada em sala de aula, a quantidade mínima de operações necessárias para multiplicar a cadeia de 6 matrizes. Durante o cálculo dos parâmetros, preencha a matriz de custos.

### 4 Força-bruta, *backtracking* e *branch-and-bound*

11. Defina e apresente semelhanças e diferenças entre força-bruta, *backtracking* e *branch-and-bound*.
12. Cite exemplos de algoritmos em que são aplicadas a força-bruta.
13. Por que *backtracking* é geralmente mais eficiente que força-bruta?
14. Quais as consequências da definição inadequada de limitantes no *branch-and-bound*?