

Estruturas de Dados Básicas II

Aula 01 - Apresentação e avaliação inicial

Ementa

1. Complexidade de Algoritmos
 - a. Noções de complexidade assintótica de algoritmos
 - i. Algoritmos e Problemas
 - ii. Noções de análise de complexidade de algoritmos
 - iii. Pior caso, melhor caso e caso médio
 - iv. Notações Assintóticas de Complexidade: Θ (Theta), O (Big-oh), Ω (Big-Omega), o (little o), ω (little omega)
 - v. Uso formal e informal da terminologia
 - b. Análise Assintótica de Algoritmos
 - i. Avaliação assintótica de somatórios e recorrências
 1. Técnicas de solução assintótica de somatórios
 2. Técnicas de solução assintótica de relações de recorrência
 - ii. Técnicas de análise de algoritmos iterativos e recursivos
 - iii. Uso formal e informal da terminologia
 - iv. Introdução à análise do caso médio
 - c. Complexidade de Problemas *versus* Complexidade de Algoritmos

Ementa

2. Árvores

- a. Conceitos de árvores
- b. Implementação de árvores genéricas
- c. Tipos diferenciados de árvores: árvores binárias, árvores de pesquisa

3. Árvores de Pesquisa

- a. Tipos de Dados: Conjuntos, Mapeamentos e Dicionários, Multiconjuntos
- b. Árvores de Pesquisa Binárias
- c. Árvores Balanceadas
 - i. Árvores rubro-negras
 - ii. Outros tipos de árvores balanceadas
- d. Árvore B
- e. Árvores Digitais

4. Heaps

- a. Tipo de Dados: Lista de Prioridades
- b. Heaps binárias e seu uso na implementação de lista de prioridades e Heapsort

5. Implementação de Conjuntos Disjuntos

- a. Tipo de Dados: Conjuntos Disjuntos (*Union-Find*)
- b. Implementação de Conjuntos Disjuntos

Metodologia

- Aulas expositivas (utilização de projetor, quadro branco e slides interativos);
- Discussão dos conceitos e problemas referentes aos assuntos abordados;
- Resolução de exercícios em sala de aula;
- Práticas de laboratório e implementação de programas.

Avaliação da Aprendizagem

3 módulos -> 3 avaliações

Cronograma de aulas - 1o módulo

Apresentação e avaliação inicial

Complexidade local

Complexidade assintótica I

Complexidade assintótica II

Exercícios

Equações de recorrência I

Equações de recorrência II

Complexidade de problemas vs. complexidade de algoritmo

Introdução à análise de caso médio

Exercícios e dúvidas para 1ª avaliação

1ª avaliação

Discussões sobre a 1ª avaliação

Cronograma de aulas - 2o módulo

Conceitos de árvores

Percursos

Conjuntos Dinâmicos/Dicionários e Árvores Binárias de Busca

Exercícios

Heapsort/Lista de prioridades (heap)

Heapsort/Lista de prioridades (heap)

Exercícios

Conjuntos disjuntos I

Conjuntos disjuntos II

Exercícios e dúvidas para 2ª avaliação

2ª avaliação

Discussões sobre a 2ª avaliação

Cronograma de aulas - 3o módulo

Árvores AVL I

Árvores AVL II

Árvores rubro-negras I

Árvores rubro-negras II

Exercícios

Árvores B

Árvores B II

Árvores digitais

Exercícios e dúvidas para 3ª avaliação

3ª avaliação

Discussões sobre a 3ª avaliação

consolidação

Bibliografia

1. Cormen, Thomas H.; Leiserson, Charles E.; Rivest, Ronald L.; Stein, Clifford. *Introduction to Algorithms*. 3ed. MIT Press. 2009.
2. Cormen, Thomas H.; Leiserson, Charles E.; Rivest, Ronald L.; Stein, Clifford. *Algoritmos: teoria e prática*. Campus. 2002. (Tradução da 2nd ed. Introduction to Algorithms)
3. Sedgewick, Robert; Flajolet, Philippe. *An Introduction to the Analysis of Algorithms*. 2nd ed. Addison-Wesley. 2013.
4. Brassard, Gilles; Bratley, Paul. *Fundamentals of Algorithmics*. Prentice Hall. 1996.
5. Szwarcfiter, Jayme Luiz. *Estruturas de Dados e seus Algoritmos*. 3ed. LTC. 2010.
6. Sedgewick, Robert; Wayne, Kevin. *Algorithms*. 4th ed. Addison-Wesley. 2011.

Estruturas de Dados Básicas

Avaliação inicial

Questão

Considere a seguinte sub-rotina `misterio()`.

```
int misterio (int n) {  
    int i = 0;  
    int p = 1;  
    while (p <= n) {  
        i = i + 1;  
        p = p * 2;  
    }  
    return i - 1;  
}
```

- I. A sub-rotina sempre termina? ou pode entrar em loop infinito?
- II. Qual o retorno de `misterio(9)`?

Questão

Considere dois algoritmos que executam n operações:

• Algoritmo 1: $f_1(n) = 2n^2 + 5n$ operações

• Algoritmo 2: $f_2(n) = 500n + 4000$ operações

Qual algoritmo é mais eficiente? Em quais casos?

Questão

Prove que $f(n)$ é $O(n^2)$, sendo:

$$f(n) = 2n^2 + 3n + 4$$

Questão

Prove que:

$$n^2 - 200n - 300 = O(n^2)$$