

PROCEDIMENTOS E RECURSOS

Nesta primeira Unidade são introduzidos aspectos básicos da construção de algoritmos, como a contagem de passos e a apresentação dos problemas da construção de algoritmos eficientes. Os somatórios mais comuns nestas aplicações são apresentados.

Os alunos terão contato com taxas de crescimento de funções e as notações empregadas para tratamento assintótico, bem como as propriedades desta notação e seus usos.

São introduzidas as classes de complexidade mais importantes e que devem ser exploradas através de exemplos e comparação com problemas reais. Os problemas P e NP são mostrados sem fazer uso de máquinas de Turing ou de não-determinismo (abordagens que são usadas na parte de teoria da computabilidade), apoiando-se na diferença entre resolver um problema em tempo polinomial (classe P) e verificar uma resposta em tempo polinomial (classe NP). Nesta etapa, deve ser direcionada uma pesquisa em busca de problemas típicos que são P/NP, para que seja exercitada a identificação nestas classes.

As recorrências são tratadas em nível introdutório, mostrando-se como converter problemas do mundo real em recorrências que depois são implementadas, mas sem mostrar o processo algébrico de resolução de recorrências, obtendo-se soluções fechadas.

Nº DA UNIDADE: 02

Nº HORAS AULA EM PERCENTUAL: 17%

CONTEÚDO: Metodologias de desenvolvimento de algoritmos

1. Algoritmos gulosos
2. Divisão e conquista
3. Programação dinâmica
4. Métodos *branch and bound*

PROCEDIMENTOS E RECURSOS

Esta Unidade deve apresentar as metodologias citadas, através de exemplos para que os alunos conheçam as possibilidades de desenvolvimento que podem ser aplicadas quando do início do desenvolvimento de um algoritmo.

Nº DA UNIDADE: 03

Nº HORAS AULA EM PERCENTUAL: 33%

CONTEÚDO: Teoria de Autômatos

1. Linguagens e definições recursivas
2. Autômatos finitos
3. Teorema de Kleene
4. Linguagens regulares e não-regulares

PROCEDIMENTOS E RECURSOS

Nesta Unidade são introduzidos os conceitos fundamentais da teoria de autômatos, incluindo formatos usuais de linguagens usadas para definir autômatos, as formas de representação de reconhecedores de linguagens e o teorema fundamental de Kleene. A diferenciação essencial entre linguagens regulares e não-regulares é explorada através da análise da capacidade de

Emitido em:

Carimbo e Assinatura da Unidade:

representação das mesmas. Recomenda-se o uso abundante de exemplificação para fixar os conceitos teóricos apresentados.

Nº DA UNIDADE: 04

Nº HORAS AULA EM PERCENTUAL: 33%

CONTEÚDO: Teoria de Turing

1. Máquinas de Turing
2. Teorema de Minsky
3. Linguagens recursivamente enumeráveis
4. Hierarquia de Chomsky
5. Computadores

PROCEDIMENTOS E RECURSOS

Nesta Unidade são introduzidos os conceitos fundamentais da teoria de Turing, incluindo a demonstração do poder aumentado de máquinas de Turing em relação a autômatos. A hierarquia de gramáticas de Chomsky deve ser introduzida usando os conceitos de gramáticas de produções e linguagens associadas, com base nos tipos de produções permitidas na gramática associada. Recomenda-se o uso abundante de exemplificação para fixar os conceitos teóricos apresentados, incluindo a re-visita de linguagens utilizadas na Unidade 03.

AVALIAÇÃO

$$G1 = (T1+T2+P1+2*T3) / 5$$

Onde:

- T1, T2 = trabalhos que abrangem as Unidades 01 e 02, respectivamente.
- P1 é uma prova escrita referente aos conteúdos da Unidade 03.
- T3 é uma nota de trabalho prático referente à Unidade 04.

Os trabalhos T1 e T2 constituem-se normalmente de problemas a serem resolvidos e entregues pelos alunos sob a forma de um **artigo** descrevendo a solução obtida por eles e descrevendo os resultados solicitados. Geralmente, não há inspeção da implementação propriamente dita. Uma série de critérios genéricos de avaliação é apresentada na página da disciplina, e a estes se somam os critérios específicos do trabalho em questão.

Embora os critérios de avaliação possam ser alterados ligeiramente de acordo com os trabalhos aplicados a cada semestre, geralmente os aspectos mais importantes na correção são os seguintes:

1. O desenvolvimento completo e coerente da solução apresentada, de forma que ela possa ser compreendida e reproduzida por alguém que leia apenas o relatório apresentado pelo aluno. Para servir de exemplo, um modelo deste tipo de relatório é dado na página da disciplina.

Emitido em:	Carimbo e Assinatura da Unidade:
Página 3 de 5	

2. O relatório inclui eventualmente o uso de exemplos detalhados, diagramas, demonstrações e outros mecanismos que sejam julgados necessários pelo aluno. É importante notar que sejam quais forem os instrumentos escolhidos o principal ponto é a clareza das explicações e do desenvolvimento feitos, e não a quantidade de mecanismos usados.
3. Desencoraja-se FORTEMENTE a apresentação de trechos de programa ou listagens dentro do relatório. Pode-se admitir a apresentação de trechos críticos de algoritmo, e mesmo assim como pseudo-código, e não como linguagens de programação.
4. A limpeza, elegância e generalidade dos algoritmos apresentados também são valorizadas.
5. Embora a obtenção de resultados corretos seja essencial para uma boa avaliação, ela não é suficiente. Neste contexto, não basta obter um resultado correto ou um programa que funcione: é necessário transmitir o conhecimento de como ele foi obtido, e esta transmissão é feita através do relatório.
6. O relatório deve ser tão completo quanto possível, explorar possibilidades diferentes de solução (se houverem), prever e/ou descrever casos em que a solução apresentada pode falhar, bem como levantar novas possibilidades. Naturalmente, o relatório deve apresentar as soluções ou casos de teste solicitados.
7. Português correto É UM CRITÉRIO DE CORREÇÃO.
8. É permitido que as soluções sejam desenvolvidas por mais de um aluno, porém os relatórios devem ser escritos individualmente.

Enquanto a prova P1 possui uma forma tradicional de prova escrita realizada durante uma aula, o trabalho prático T3 normalmente consiste na implementação de uma máquina de Turing usando um software de apoio de domínio público para exercitar a complexidade de conceitos imbuídos na definição de tais máquinas. Normalmente, um software estará disponível na página da disciplina para que seja permitido aos alunos realizarem o trabalho.

BIBLIOGRAFIA:

- **BÁSICA:**

1. Cormen, Thomas H. **Introduction to algorithms**. MIT Press, 2001. Registro 005.1 I61i.
2. Aho, Alfred V. **Foundations of computer science**. Computer Science Press, 1998. Registro 004 A286fb.
3. Cohen, D. I. A. **Introduction to computer theory**. Wiley & Sons, Inc. Revised Printing, 1991. Registro 004 C678ia.

<p>Emitido em:</p> <p>Página 4 de 5</p>	<p>Carimbo e Assinatura da Unidade:</p>
--	---

- **COMPLEMENTAR:**

1. Gersting, Judith L. **Fundamentos matemáticos para ciência da computação**. LTC, 2001. Registro 004.0151 G383mb.
2. Levitin, Anany **Introduction to the design and analysis of algorithms**. Addison-Wesley, 2003. Registro 005.1 L666i.
3. Sedgewick, Robert **Algorithms in C**. Addison-Wesley, 2002. Registro 005.131 S448a.
4. Skiena, Steven S. **The algorithm design manual**. Springer-Verlag, 1998. Registro 005.1 S628a.

Emitido em: Página 5 de 5	Carimbo e Assinatura da Unidade:
----------------------------------	----------------------------------