

CES-11

Algoritmos e Estruturas de Dados

Carlos Forster
Carlos Alonso
Juliana Bezerra

Objetivos gerais

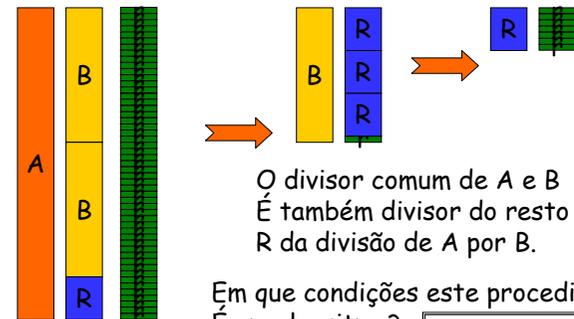
- Compreensão da necessidade de uma boa estruturação das informações processadas no computador
- Capacidade de escolher a estrutura de dados mais adequada para uma determinada aplicação
- Capacidade de programar métodos que criam e manipulam essas estruturas
- Famoso livro de Niklaus Wirth: *Algorithms + Data Structures = Programs*
- "Estar alfabetizado" *versus* "saber redigir"

Algoritmo

- Informalmente, um algoritmo é qualquer procedimento computacional bem definido que recebe um conjunto de valores como *entrada* e produz outro conjunto de valores como *saída*
- Algoritmo correto
 - Para cada instância de entrada, termina e produz uma saída correta
- Algoritmo incorreto
 - Quando não para (permanece em *loop*)
 - Quando para com uma resposta diferente da desejada

Exemplo

- Máximo divisor comum (Euclides)



Segmentos
Comensuráveis

Programa do MDC

```

int mdc(int a, int b)
{
    int r;
    do{
        r=a%b;
        a=b;
        b=r;
    }while (r>0);
    return a;
}

```

a	b	r
42	30	12
30	12	6
12	6	0
6	0	

Demonstração do MDC

- MDC de A e B para inteiros positivos é o MDC de B com o resto R da divisão de A por B.
 - Seja X um divisor comum de A e B, então $A=kX$ e $B=mX$
 - Dividindo A por B, temos $A=nB+R$
 - $kX=nmX+R$
 - Logo, $R=(k-nm)X$
- Volta da demonstração
 - $R=pX$; $B=mX$; $A=nB+R \rightarrow A=nmX+pX=(nm+p)X$
- Mostrar que se X é o MDC(B,R) então é MDC(A,B):
 - Seja X o maior DC de B e R
 - Se X não for o maior DC de A e B, então $Y>X$ o seria, mas
 - Se Y é DC de A e B, então Y é DC de B e R e maior que X
 - Então Y seria o MDC(A,B) - prova por contradição

Exemplos de algoritmos

- Projeto Genoma Humano
 - O DNA humano possui cerca de 100 mil genes e 3 bilhões de pares de bases químicas
 - Necessidade de armazenar os dados e analisá-los
 - Inserção, eliminação e atualização desses dados
 - Ordenação, classificação e pesquisa de informações
 - Correlação entre esses dados
 - Cálculos científicos envolvendo grandes matrizes multidimensionais
- Internet
 - Emprega algoritmos para gerenciar e manipular um enorme volume de dados
 - Localização de rotas por onde os dados trafegam
 - Mecanismos de pesquisa de páginas

Exemplos

Email(<http://royal.pingdom.com/2010/01/22/internet-2009-in-numbers/>)

90 trillion - The number of emails sent on the Internet in 2009
247 billion - Average number of email messages per day.
1.4 billion - The number of email users worldwide.
100 million - New email users since the year before.
81% - The percentage of emails that were spam.
92% - Peak spam levels late in the year.
24% - Increase in spam since last year.
200 billion - The number of spam emails per day (assuming 81% are spam).

Exemplos

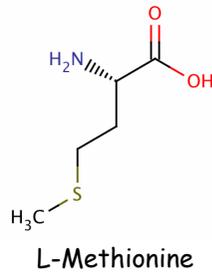
>Methionine synthase

Protein Sequence

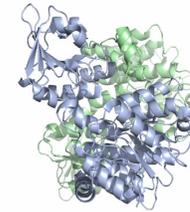
```
MSPALQDLSQPEGLKKTLRDEINAILQKRIMVLDDGGMGTMIQREKLINEEHFRGQEPKDHARPLKGNNDILSTQDPVIVYQIRKEYLLAGADIETNTPSSTSIAQADYGLHLAYRMNMC
SAGVARKAAEVTLQGTGKRFVAGALGPTNKLTVSPSVERPDRYNTFDELVEARYQEA
KGLLDGGVDILLIETIPD ....
```

Gene Sequence

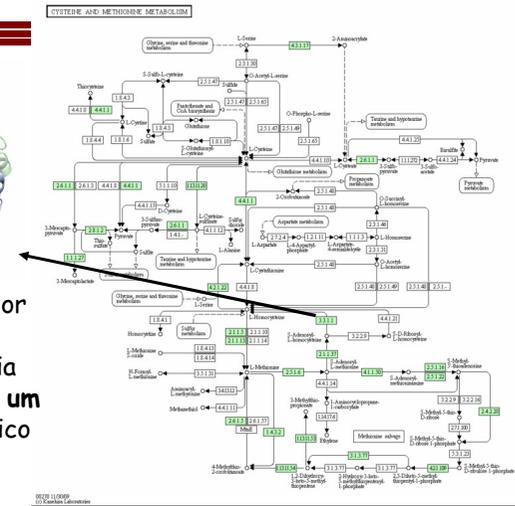
```
ATGTCACCCGGCTCCAGACCTGTGCAACCCGAGGCTGGAAGAAAACCCGCGGGAT
GAGATCAATGCCATTCGCAAGAGGATTATGGTCTGGATGGAGGATGGGGACCATG
ATCCAGCGGGAGAAGCTAAACGAAGAACAATCCGAGGTCAGGAATTTAAAGATCATGCC
AGCCCGCTGAAGGCAACAATGACATTTAAGTATAACTCAGCCTGATGTCATTTACCAA
ATCCATAAGGAATACTTGCCTGGCTGGGGCAGATATCATTGAACAAATACTTTAGCAGC
ACTAGTATGCCCAAGCTGACTATGGCCTTGAACACTTGGCCTACCGGATGAACATGTGC
TCTGCAGGAGTGGCCAGAAAGCTGCCGAGGAGGTAACCTCCAGACAGGAATTAAGAGG
TTTGTGGCAGGGGCTCTGGTCCGACTAATAAGCACTCTCTGTG ...
```



Exemplos



Desenho gerado por computador da estrutura terciária de uma enzima de um processo metabólico



Exemplos de algoritmos

- Comércio eletrônico
 - Necessidade de manter informações privativas (número de cartões, senhas, etc.)
 - Criptografia de chave pública e assinaturas digitais
 - Compactação e descompactação
- Indústria e comércio
 - Alocação adequada de recursos escassos
 - Onde localizar os poços de petróleo para maximizar o lucro?
 - Onde investir em publicidade? Público? Região?
 - Como designar tripulações de voo de um modo menos dispendioso?
 - Qual melhor investimento: ações, opções, títulos públicos?

Desenvolvimento de um algoritmo

- Características dos algoritmos
 - Considere um problema prático
 - Existem muitas resoluções candidatas, mas provavelmente não serão bem aquilo que desejamos...
- Qual o melhor algoritmo?
 - Basta que forneça a resposta correta?
 - Também é importante a sua *eficiência*
 - Tempo de computação
 - Memória utilizada

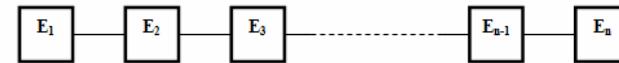
Convém conhecer os algoritmos existentes para então desenvolver o seu

Estruturas de dados

- O consumo de tempo e de memória torna-se extremamente crítico quando o universo de informações é muito grande...
- A eficiente utilização dos recursos computacionais e a redução do tempo de resposta dependem de dois fatores:
 - Boa estruturação das informações
 - Bons algoritmos que manipulem essas estruturas
- Principais modelos para visualizar, interpretar e armazenar dados:
 - Listas lineares
 - Árvores
 - Grafos

Listas lineares

- Seus elementos formam uma sequência linear:



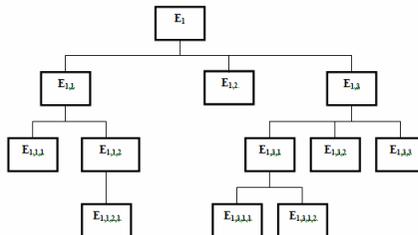
- Cada elemento possui um antecessor e um sucessor (exceto o primeiro e o último)
- Tabelas, em geral, se enquadram neste modelo:

Nome	Endereço	Outros

- Listas telefônicas
- Folhas de pagamento
- Livros de uma biblioteca
- Tabelas de um banco de dados relacional

Árvores

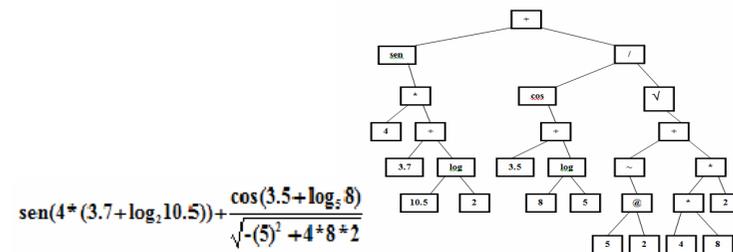
- Há uma hierarquia entre os elementos:



- Cada elemento:
 - Tem um único pai (exceto a raiz)
 - Pode ter vários filhos
 - Não pode ser pai de nenhum ancestral

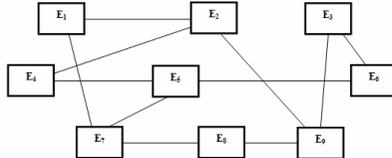
Exemplos de árvores

- Organograma de empresas
- Organização de livros e cursos
- Jogos eliminatórios de um campeonato
- Expressões aritméticas



Grafos

- Há uma interligação geral entre os elementos, sem formar seqüências ou hierarquias:



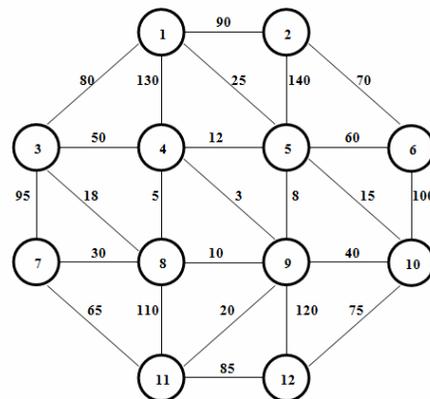
- Exemplos:
 - Tarefas de um projeto
 - Sistema rodoviário
 - Redes de interconexão
 - Fornecimento de produtos entre fábricas
 - Máquinas de estados finitos
 - etc.

Importância da disciplina

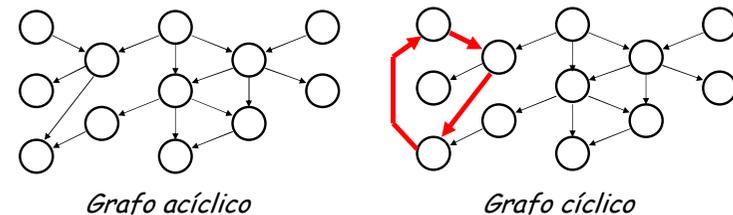
- Na Ciência da Computação:
 - Processo de compilação
 - Gerenciamento de programas e de memória
 - Construção de bancos de dados
 - etc.
- Mesmo sem ser um especialista na área, um engenheiro nos dias de hoje deve ser capaz de elaborar algoritmos não triviais.
- Alguns exemplos de problemas em grafos:
 - Caminhos mais curtos
 - Validação de grafos acíclicos
 - Componentes fortemente conexos
 - Pontos de articulação
 - Árvore geradora de custo mínimo

Caminhos mais curtos

- Distâncias a partir de um dado vértice
- Aplicações:
 - Transporte terrestre ou aéreo
 - Robótica
 - Projetos de VLSI



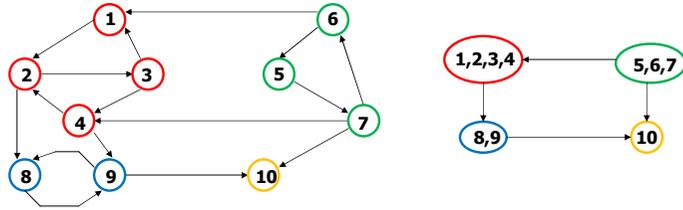
Validação de grafos acíclicos



- Aplicações:
 - Gerenciamento de projetos (redes PERT-CPM: *Program Evaluation and Review Technique, Critical Path Method*)
 - Simulação de circuitos combinacionais

Componentes fortemente conexos

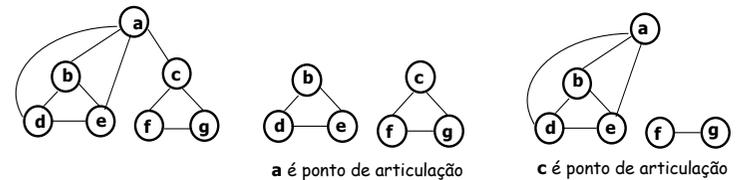
- Subconjuntos maximais de vértices onde há caminhos de qualquer vértice a todos os outros.



- Aplicações:
 - Redução do tamanho de determinados problemas
 - Classes de equivalência em circuitos digitais
 - Privacidade em sistemas de comunicação

Pontos de articulação

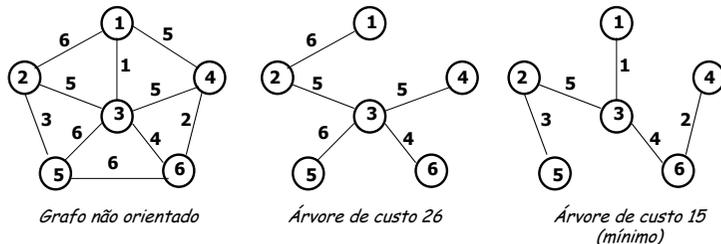
- Um vértice é ponto de articulação se, ao ser removido, desconecta o grafo.



- Aplicações:
 - Redes de interconexão em geral
 - Sistemas de transmissão de energia elétrica
 - Sistemas de distribuição hidráulica

Árvore geradora de custo mínimo

- Às vezes, é necessário encontrar um subgrafo gerador (todos os vértices e um subconjunto das arestas) que seja árvore e que tenha custo mínimo.



- Aplicações:
 - Redes de interconexão em geral

Plano do curso

- Primeiro Bimestre:
 - Breve revisão
 - Noções de complexidade de algoritmos
 - Listas lineares
 - Pilhas, filas e *deques*
 - Árvores
 - Árvores binárias
- Segundo Bimestre:
 - Algoritmos de ordenação
 - Paradigma da Divisão-e-Conquista
 - Grafos
 - Representações
 - Soluções de alguns problemas clássicos
 - Noções de Programação Orientada a Objetos

Avaliações individuais

- Em cada bimestre:
 - 2 provas
 - 3 tarefas de laboratório
- Pesos:
 - Provas: 60%
 - Média dos exercícios: 40%

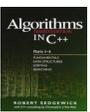
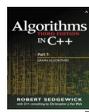
Premissas éticas nas tarefas de Lab

- É permitido:
 - Usar material didático (slides, apostilas, códigos) de outros professores do ITA ou disponível na internet (neste último caso, se for código, sem fornecê-lo a outros colegas)
 - Pensar na solução junto com um colega, antes de programarem
 - Trocar idéias com outro colega, mas sem olhar o código que ele escreveu
 - Ajudar um colega a encontrar erros de codificação, desde que já tenha terminado o próprio exercício
- Não é permitido:
 - Utilizar código pronto encontrado na internet
 - Olhar ou copiar soluções de outro aluno (da mesma turma ou de anteriores)
 - Fazer o exercício (mesmo parcialmente) de um colega com dificuldades
 - Escrever o código junto com outro colega

Bibliografia

- P. Feofiloff
Algoritmos em Linguagem C 
- N. Ziviani
Projeto de Algoritmos 
- A.M. Tanenbaum, Y. Langsam, M.J. Augenstein
Estruturas de Dados usando C 
- M.T. Goodrich, R. Tamassia
Projeto de Algoritmos 
- B.R. Preiss
Estruturas de Dados e Algoritmos 

Bibliografia complementar

- T.H. Cormen, C.E. Leiserson, R.L. Rivest
Introduction to Algorithms 
- A. Drozdek
Estrutura de Dados e Algoritmos em C++ 
- R. Sedgwick
Algorithms in [C, C++, Java]   
- A. V. Aho, J. E. Hopcroft, J. D. Ullman
Data Structures and Algorithms 