

Paradigmas de projeto de algoritmos

■ Apresentação

- Introdução
- Paradigmas
 - Recursividade
 - Algoritmos tentativa e erro
 - Divisão e conquista
 - Programação dinâmica
 - Algoritmos gulosos
 - Algoritmos aproximados

Paradigmas de projeto de algoritmos

■ Introdução

- O projeto de algoritmos requer **abordagens adequadas**
 - A forma como um algoritmo aborda o problema pode levar a um **desempenho ineficiente**
 - Em certo casos, o **algoritmo pode não conseguir resolver o problema** em tempo viável

Paradigmas de projeto de algoritmos

■ Introdução

- Infelizmente, **não** existe um paradigma que seja o melhor dentre todos!
 - Um problema pode ser resolvido de maneira mais eficiente adotando-se um paradigma do que outro
 - Um paradigma pode levar a um algoritmo $O(2^n)$ e outro paradigma a um algoritmo $O(n^3)$, para a resolução de um mesmo problema

Paradigmas de projeto de algoritmos

■ Introdução

- Por exemplo
 - Como ordenar um vetor de inteiros?
 - Como realizar o produto entre n matrizes de modo que o número de operações seja o menor possível?
 - Problema da mochila
 - Considere n itens a serem levados para uma viagem, dentro de uma mochila de capacidade L que não pode comportar todos os itens. Cada item tem um peso w_i e uma utilidade c_i . Quais itens escolher, de modo que a utilidade total dos itens levados seja a maior possível?

Paradigmas de projeto de algoritmos

■ Apresentação

- **Recursividade**
- Algoritmos tentativa e erro
- Divisão e conquista
- Programação dinâmica
- Algoritmos gulosos
- Algoritmos aproximados

Paradigmas de projeto de algoritmos

■ Recursividade

- **Paradigma** ou **técnica de programação?**
- Alguns critérios devem ser observados
 - Definição recursiva do problema
 - Condição de parada
 - Garantia de terminar
- Soluções **mais elegantes** e **claras**

Paradigmas de projeto de algoritmos

■ Apresentação

- Recursividade
- **Algoritmos tentativa e erro**
- Divisão e conquista
- Programação dinâmica
- Algoritmos gulosos
- Algoritmos aproximados

Paradigmas de projeto de algoritmos

■ Algoritmos tentativa e erro

- Um algoritmo tentativa e erro é aquele que testa **exaustivamente** todas as soluções possíveis de um problema, de modo a obter a desejada
- As **soluções são testadas indiscriminadamente**
 - Não utiliza critérios para eliminar outras soluções que não poderão ser melhores que a obtida no estágio considerado

Paradigmas de projeto de algoritmos

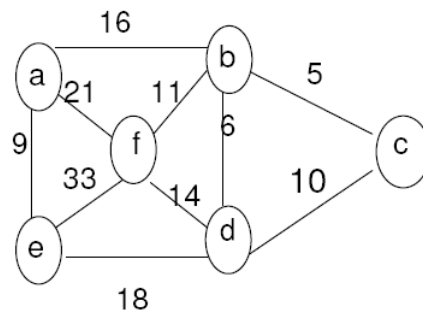
■ Algoritmos tentativa e erro

- As soluções são **enumeradas** de modo semelhante ao **percurso em uma árvore** que possua todas as soluções
 - Muitas vezes a “árvore” de soluções cresce **exponencialmente!**

Paradigmas de projeto de algoritmos

■ Algoritmos tentativa e erro

- **Exercício:**
 - Qual o menor caminho da cidade **a** até a **c**?



Paradigmas de projeto de algoritmos

■ Algoritmos tentativa e erro

- **Exercício:**
 - **TODO**s os caminhos são enumerados:
 - $a \rightarrow b \rightarrow c$: 21
 - $a \rightarrow b \rightarrow d \rightarrow c$: 32
 - $a \rightarrow b \rightarrow f \rightarrow d \rightarrow c$: 51
 - ...

Paradigmas de projeto de algoritmos

■ Apresentação

- Recursividade
- Algoritmos tentativa e erro
- **Divisão e conquista**
- Programação dinâmica
- Algoritmos gulosos
- Algoritmos aproximados

Paradigmas de projeto de algoritmos

■ Divisão e conquista

- **Passos básicos**
 1. **Dividir o problema** a ser resolvido em subproblemas menores e independentes
 2. Encontrar **soluções para as partes**
 3. **Combinar as soluções** obtidas em uma solução global
- Os algoritmos podem utilizar **recursão** para dividir e combinar

Paradigmas de projeto de algoritmos

■ Divisão e conquista

- Processo (geralmente recursivo)
 - dada uma entrada, se ela é suficientemente **simples**, obtemos **diretamente** uma saída correspondente
 - caso contrário, ela é **decomposta** em entradas mais simples, para as quais aplicamos o mesmo processo, obtendo saídas correspondentes que são então combinadas em uma saída para a entrada original

Paradigmas de projeto de algoritmos

■ Divisão e conquista

- Exemplos de algoritmos
 - ?

Paradigmas de projeto de algoritmos

■ Divisão e conquista

- Exemplos de algoritmos
 - Busca binária
 - *Mergesort*, *Quicksort*
 - Maior elemento de uma seqüência
 - Fibonacci recursivo

Paradigmas de projeto de algoritmos

■ Apresentação

- Recursividade
- Algoritmos tentativa e erro
- Divisão e conquista
- **Programação dinâmica**
- Algoritmos gulosos
- Algoritmos aproximados



Lucía

Paradigmas de projeto de algoritmos

■ Apresentação

- Recursividade
- Algoritmos tentativa e erro
- Divisão e conquista
- Programação dinâmica
- **Algoritmos gulosos**
- Algoritmos aproximados

Paradigmas de projeto de algoritmos

■ Algoritmos gulosos

- São tipicamente usados para resolver **problemas de otimização**
- Por exemplo, o algoritmo para encontrar o **caminho mais curto entre duas cidades**
 - Um algoritmo guloso escolhe a estrada que parece **mais promissora no instante atual** e **nunca muda essa decisão**, independentemente do que possa acontecer depois

Paradigmas de projeto de algoritmos

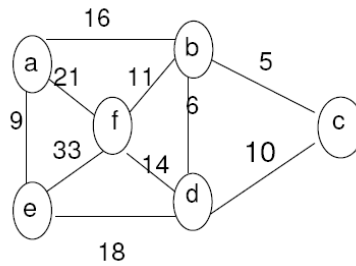
■ Algoritmos gulosos

- A cada **iteração**
 - seleciona um elemento conforme uma função gulosa
 - marca-o para não considerá-lo novamente nos próximos estágios
 - atualiza a entrada
 - examina o elemento selecionado quanto a sua viabilidade
 - decide a sua participação ou não na solução

Paradigmas de projeto de algoritmos

■ Algoritmos gulosos

- **Exercício:** Calcule o menor caminho da cidade **a** até a **c**, utilizando um algoritmo guloso.



Paradigmas de projeto de algoritmos

■ Apresentação

- Recursividade
- Algoritmos tentativa e erro
- Divisão e conquista
- Programação dinâmica
- Algoritmos gulosos
- **Algoritmos aproximados**

Paradigmas de projeto de algoritmos

■ Algoritmos aproximados

- Gera **soluções aproximadas**, que podem não ser ótimas, mas são próximas delas
- Faz-se necessária uma **medida de qualidade**
- Exemplos?

Paradigmas de projeto de algoritmos

■ Algoritmos aproximados

- Gera **soluções aproximadas**, que podem não ser ótimas, mas são próximas delas
- Faz-se necessária uma **medida de qualidade**
- Exemplos?
 - Caixeiro viajante (menor caminho entre cidades)
 - Tradução automática

Paradigmas de projeto de algoritmos

■ Considerações finais

- Outros paradigmas?

Paradigmas de projeto de algoritmos

■ Considerações finais

- Outros paradigmas?
 - Indução matemática
 - Balanceamento (ou técnica?)
 - Algoritmos paralelos