Introdução à Disciplina de Projeto e Análise de Algoritmos

Prof. Daniel Kikuti

Universidade Estadual de Maringá

Sumário

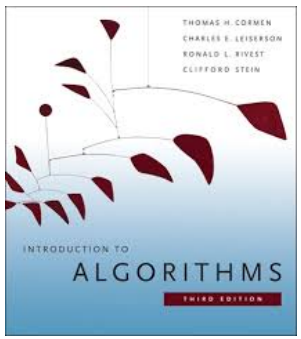
- Informações administrativas.
- Sobre a disciplina.
- Definição do objeto de estudo.
- Como avaliar algoritmos?
- Colocando a mão na massa.

Informações administrativas

- Ementa e critério de avaliação.
- Datas importantes.
- Disciplina no Moodle (moodlep.uem.br).

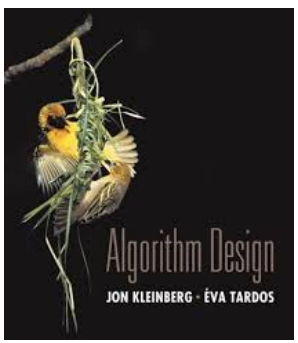
Referências Bibliográficas

Principal referência

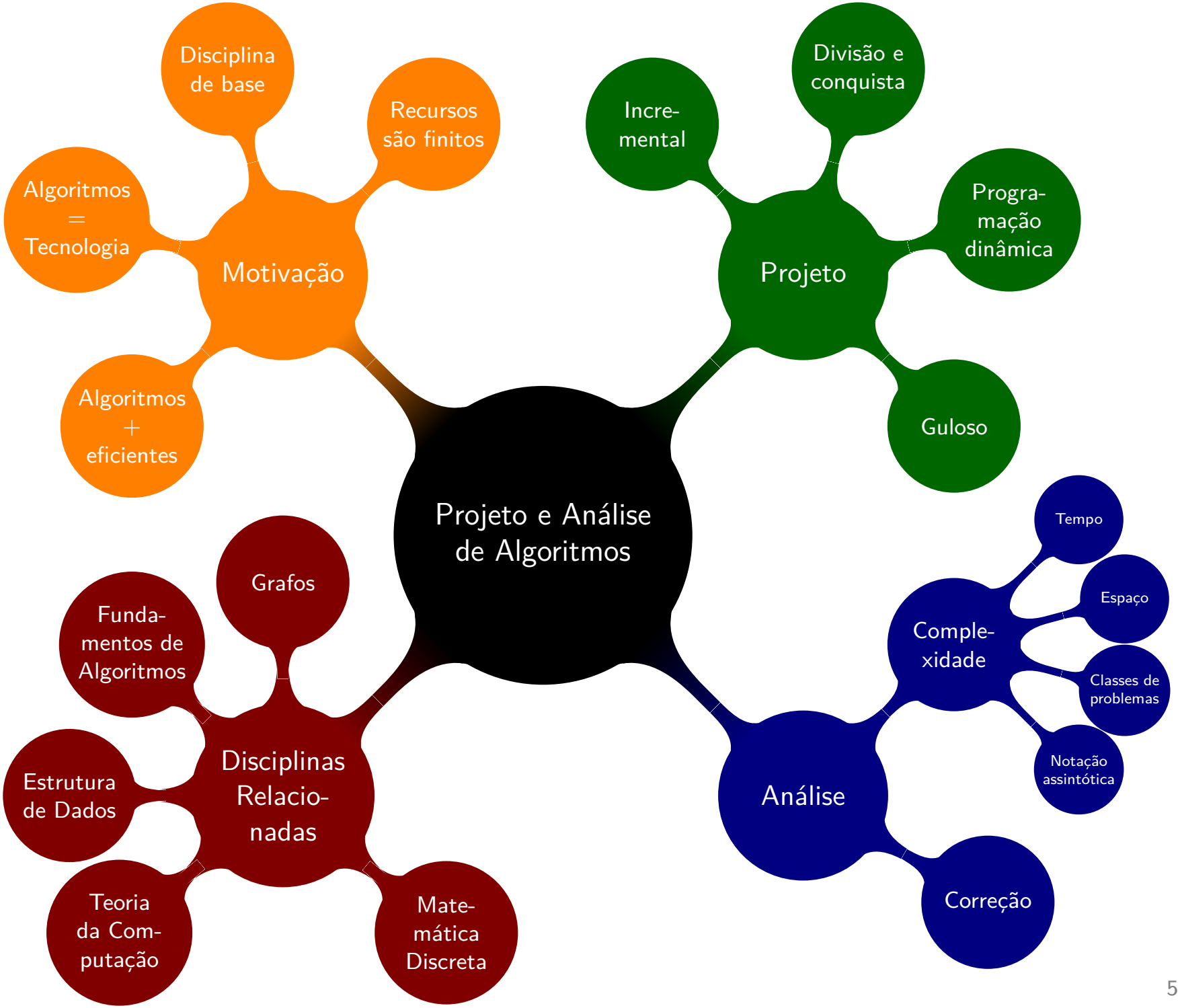


Cormen, T.H., Leiserson, C.E., Rivest, R.L. & Stein, C. *Introduction to algorithms,* 3^{rd} ed. The MIT Press, 2009. [6 exemplares: BCE-Acervo (005.1 I61 3.ed. 2009)]

Complementar



Kleinberg, J. & Tardos, E. *Algorithm Design*. Pearson/Addison-Wesley, 2005. [1 exemplar: BCE-Acervo (005.1 K64a 2006)]


Projeto e Análise de Algoritmos
Motivação
Disciplina de base
Recursos são finitos
Algoritmos = Tecnologia
Algoritmos + eficientes
Projeto
Divisão e conquista
Incre-mental
Progra-mação dinâmica
Guloso
Disciplinas Relacio-nadas
Grafos
Funda-mentos de Algoritmos
Estrutura de Dados
Teoria da Com-putação
Mate-mática Discreta
Análise
Comple-xidade
Tempo
Espaço
Classes de problemas
Notação assintótica
Correção

Definição do objeto de estudo

O que é um algoritmo?

Procedimento computacional bem definido que recebe um valor (ou conjunto de valores) como entrada e produz algum valor (ou conjunto de valores) como saída. Uma sequencia de passos computacionais que transformam uma entrada em uma saída. [Cormen et. al, 2009].

Definição do objeto de estudo

O que é um algoritmo?

Procedimento computacional bem definido que recebe um valor (ou conjunto de valores) como entrada e produz algum valor (ou conjunto de valores) como saída. Uma sequencia de passos computacionais que transformam uma entrada em uma saída. [Cormen et. al, 2009].

Exemplos de problemas computacionais

- Conjunto de números e queremos encontrar uma ordenação crescente destes números.
- Mapa rodoviário e queremos encontrar um caminho entre dois pontos.

O que avaliar em algoritmos?

O que avaliar em algoritmos?

Simplicidade

Um algoritmo é **simples** se puder ser facilmente entendido, implementado e mantido.

O que avaliar em algoritmos?

Simplicidade

Um algoritmo é **simples** se puder ser facilmente entendido, implementado e mantido.

Correção (corretude)

Um algoritmo está **correto** se para toda instância de entrada ele termina com a saída correta.
"Testes servem apenas para provar que um algoritmo tem erros, nunca prova que está correto." (Dijkstra)

O que avaliar em algoritmos?

Simplicidade

Um algoritmo é **simples** se puder ser facilmente entendido, implementado e mantido.

Correção (corretude)

Um algoritmo está **correto** se para toda instância de entrada ele termina com a saída correta.
"Testes servem apenas para provar que um algoritmo tem erros, nunca prova que está correto." (Dijkstra)

Eficiência

A eficiência de um algoritmo é uma medida quantitativa dos recursos necessários (tempo, espaço, etc) para seu funcionamento.

Como medir a eficiência?

Método experimental

- Implementar os algoritmos.
- Executar um grande número de vezes.
- Analisar os resultados.
- Depende de fatores como: linguagem de programação usada e recursos de hardware.

Como medir a eficiência?

Método experimental

- Implementar os algoritmos.
- Executar um grande número de vezes.
- Analisar os resultados.
- Depende de fatores como: linguagem de programação usada e recursos de hardware.

Método analítico

- Construir um modelo matemático do algoritmo e comparar com outros algoritmos por meio de seus modelos.
- Critério **uniforme** e **independente de tecnologia** para comparar algoritmos.

Modelo formal de computação

O modelo formal de computação estabelece:

- recursos disponíveis;
- instruções básicas;
- custo (tempo) das instruções.

Análise de complexidade

Este modelo formal permite uma análise matemática do tempo (ou espaço) que um algoritmo gasta em função do tamanho da entrada.

Modelos de computação

Máquinas de Turing

- Modelo clássico de computação.
- Muito baixo nível.
- Difícil de descrever algoritmos (muito formal).

Modelo RAM (*Random-Access-Machine)*

- As operações são executadas sequencialmente.
- A execução de toda e qualquer operação toma uma unidade de tempo (tempo constante).
- a memória é infinita – não considera a hierarquia de memória.

Equivalência entre modelos

Máquina de Turing e Máquina RAM são computacionalmente equivalentes.

Exemplo

Algoritmo que encontra o máximo de um vetor de inteiros

MAXIMO$(A[], n)$

```
1 max ← A[1]
2 for i ← 2 to n do
3 |   if A[i] > max then
4 |   |   max ← A[i]
5 return max
```

No modelo RAM

- Cada linha está associada a um custo.
- Precisamos computar o custo total do algoritmo (multiplicar o custo de cada linha pelo número de vezes que cada linha é executada e somar todos estes custos).

Colocando a mão na massa

Quantas vezes os algoritmos irão imprimir a mensagem “Hi”?
Saída do algoritmo em função da entrada.

Algoritmo iterativo

```
PROG1(n)
1 for i ← 1 to n do
2 |   PRINT “Hi”
```

Algoritmo recursivo

```
PROG2(n)
1 if n = 1 then
2 |   PRINT “Hi”
3 else
4 |   PRINT “Hi”
5 |   PROG2(n − 1)
```

Colocando a mão na massa

Quantas vezes os algoritmos irão imprimir a mensagem “Hi”?
Saída do algoritmo em função da entrada.

Algoritmo iterativo

PROG1(n)

```
1 for i ← 1 to n do
2 |   PRINT “Hi”
```

Resposta: n vezes

Algoritmo recursivo

PROG2(n)

```
1 if n = 1 then
2 |   PRINT “Hi”
3 else
4 |   PRINT “Hi”
5 |   PROG2(n − 1)
```

Resposta: n vezes

Colocando a mão na massa

Quantas vezes os algoritmos irão imprimir a mensagem “Hi”?
Saída do algoritmo em função da entrada.

PROG3(n)

```
1 for i ← 1 to n do
2 |   for j ← 1 to n do
3 |   |   PRINT “Hi”
```

PROG4(n)

```
1 for i ← 1 to i² ≤ n do
2 |   PRINT “Hi”
```

Colocando a mão na massa

Quantas vezes os algoritmos irão imprimir a mensagem “Hi”?
Saída do algoritmo em função da entrada.

PROG3(n)

```
1 for i ← 1 to n do
2 |   for j ← 1 to n do
3 |   |   PRINT “Hi”
```

Resposta: n^2 vezes

PROG4(n)

```
1 for i ← 1 to i² ≤ n do
2 |   PRINT “Hi”
```

Resposta: $\sqrt{n}$ vezes

Colocando a mão na massa

Ordenação de um vetor de n elementos

- Considere dois computadores cujas capacidades de processamento são: $A = 10^9$ e $B = 10^7$ instruções por segundo (A é 100 vezes mais rápido que B).
- Algoritmo 1 (implementado na máquina A) executa $f(n) = 2n^2$ instruções.
- Algoritmo 2 (implementado na máquina B) executa $f(n) = 50n \lg n$ instruções.

Quanto tempo cada um dos algoritmos gastará para ordenar um milhão de elementos? Qual algoritmo é mais rápido?

Colocando a mão na massa

Ordenação de um vetor de n elementos

- Considere dois computadores cujas capacidades de processamento são: $A = 10^9$ e $B = 10^7$ instruções por segundo (A é 100 vezes mais rápido que B).
- Algoritmo 1 (implementado na máquina A) executa $f(n) = 2n^2$ instruções.
- Algoritmo 2 (implementado na máquina B) executa $f(n) = 50n \lg n$ instruções.

Quanto tempo cada um dos algoritmos gastará para ordenar um milhão de elementos? Qual algoritmo é mais rápido?

Resposta

O Algoritmo 1 na máquina A gasta 2000 segundos. O Algoritmo 2 na máquina B gasta 100 segundos.

Tarefa

Leitura fácil

Leia o Capítulo 1 do Cormen (10 páginas). Observe como os autores encaram as questões:

- O que são algoritmos?
- Por que o estudo de algoritmos vale a pena?
- Qual o papel dos algoritmos em relação a outras tecnologias usadas em computadores?

Preencher tabela

Resolvam o problema 1-1 (Comparação de tempos de execução). [Não precisa calcular para mês, ano e século].

Alguns exercícios

Quantas vezes os algoritmos irão imprimir a mensagem "Hi"?
Saída do algoritmo em função da entrada (para $n > 2$).

PROG5(n)

```
1 while n > 1 do
2 |   n ← n/2
3 |   PRINT "Hi"
```

PROG6(n)

```
1 for i ← 1; i ≤ n; i ← i * 2 do
2 |   PRINT "Hi"
```

Alguns exercícios

Quantas vezes os algoritmos irão imprimir a mensagem "Hi"?
Saída do algoritmo em função da entrada (para $n > 2$).

PROG5(n)

```
1 while n > 1 do
2 |   n ← n/2
3 |   PRINT "Hi"
```

Resposta: $\lfloor \lg n \rfloor$ vezes

PROG6(n)

```
1 for i ← 1; i ≤ n; i ← i * 2 do
2 |   PRINT "Hi"
```

Resposta: $\lfloor \lg n \rfloor + 1$ vezes

Mais exercícios

Quantas vezes o algoritmo irá imprimir a mensagem "Hi"? Saída do algoritmo em função da entrada.

```
PROG7(n)
1 for i ← n/2 to n do
2 |   for j ← 1 to n/2 do
3 |   |   for k ← 1; k ≤ n; k ← k * 2 do
4 |   |   |   PRINT "Hi"
```

Mais exercícios

Quantas vezes o algoritmo irá imprimir a mensagem "Hi"? Saída do algoritmo em função da entrada.

```
PROG7(n)
1 for i ← n/2 to n do
2 |   for j ← 1 to n/2 do
3 |   |   for k ← 1; k ≤ n; k ← k * 2 do
4 |   |   |   PRINT "Hi"
```

Resposta: $\frac{n^2}{4} \lg n$ vezes

Muito mais exercícios.

Quantas vezes o algoritmo irá imprimir a mensagem “Hi”? Saída do algoritmo em função da entrada.

PROG8(n)

```
1 for i ← 1 to n do
2 |   for j ← 1 to i^2 do
3 |   |   for k ← 1 to n/2 do
4 |   |   |   PRINT “Hi”
```

Muito mais exercícios.

Quantas vezes o algoritmo irá imprimir a mensagem "Hi"? Saída do algoritmo em função da entrada.

PROG8(n)

```
1 for i ← 1 to n do
2 |   for j ← 1 to i^2 do
3 |   |   for k ← 1 to n/2 do
4 |   |   |   PRINT "Hi"
```

Resposta: $\frac{n}{2}\left(\frac{n(n+1)(2n+1)}{6}\right)$ vezes