Intuitivamente

Comparação assintótica, ou seja, para n ENORME.

comparação	comparação assintótica
$T(n) \leq f(n)$	$T(n)$ é $\mathrm{O}(f(n))$
$T(n) \geq f(n)$	$T(n)$ é $\Omega(f(n))$
$T(n) = f(n)$	$T(n)$ é $\Theta(f(n))$

Nomes de classes Θ

classe	nome
$\Theta(1)$	constante
$\Theta(\log n)$	logarítmica
$\Theta(n)$	linear
$\Theta(n \log n)$	$n \log n$
$\Theta(n^2)$	quadrática
$\Theta(n^3)$	cúbica
$\Theta(n^k)$ com $k \geq 1$	polinomial
$\Theta(2^n)$	exponencial
$\Theta(a^n)$ com $a > 1$	exponencial

Palavras de Cautela

Suponha que $\mathcal{A}$ e $\mathcal{B}$ são algoritmos para um mesmo problema. Suponha que o consumo de tempo de $\mathcal{A}$ é "essencialmente" $100\,n$ e que o consumo de tempo de $\mathcal{B}$ é "essencialmente" $n \log_{10} n$.

$100\,n$ é $\Theta(n)$ e $n \log_{10} n$ é $\Theta(n \lg n)$.
Logo, $\mathcal{A}$ é assintoticamente mais eficiente que $\mathcal{B}$.

$\mathcal{A}$ é mais eficiente que $\mathcal{B}$ para $n \geq 10^{100}$.

10^{100} = um *googol*

> número de átomos no universo observável,

um número ENORME

Palavras de Cautela

Conclusão:

Lembre das constantes e termos de baixa ordem que estão “escondidos” na notação assintótica.

Em geral um algoritmo que consome tempo $\Theta(n \lg n)$, e com fatores constantes razoáveis, é bem eficiente.

Um algoritmo que consome tempo $\Theta(n^2)$ pode, algumas vezes, ser satisfatório.

Um algoritmo que consome tempo $\Theta(2^n)$ é dificilmente aceitável.

Do ponto de vista de AA, eficiente $=$ polinomial.

Análise do caso médio

Consideremos que a entrada do algoritmo é uma permutação de 1 a n escolhida com probabilidade uniforme.

Qual é o número esperado de execuções da linha 4?

ORDENA-POR-INSERÇÃO (A, n)

1 **para** $j \leftarrow 2$ **até** n **faça**
2 $\quad$ $chave \leftarrow A[j]$
3 $\quad$ $i \leftarrow j - 1$
4 $\quad$ **enquanto** $i \geq 1$ **e** $A[i] > chave$ **faça**
5 $\quad\quad$ $A[i+1] \leftarrow A[i]$ $\quad \triangleright$ desloca
6 $\quad\quad$ $i \leftarrow i - 1$
7 $\quad$ $A[i+1] \leftarrow chave$ $\quad \triangleright$ insere

Análise do caso médio

ORDENA-POR-INSERÇÃO (A, n)

1 **para** $j \leftarrow 2$ **até** n **faça**
2 $\quad$ $chave \leftarrow A[j]$
3 $\quad$ $i \leftarrow j - 1$
4 $\quad$ **enquanto** $i \geq 1$ **e** $A[i] > chave$ **faça**
5 $\quad\quad$ $A[i+1] \leftarrow A[i]$ $\quad \triangleright$ desloca
6 $\quad\quad$ $i \leftarrow i - 1$
7 $\quad$ $A[i+1] \leftarrow chave$ $\quad \triangleright$ insere

nº de iterações $\in \{0, \ldots, j-1\}$

Para cada valor de j, a linha 4 é executada de 1 a j vezes.

$t + 1$ — teste de saída

Qual é a probabilidade ~~dela~~ de ela ser executada t vezes?

$= \text{prob}(A[1..j]$ tem exatamente t elementos maiores que $A[j])$

permutação aleatória uniforme de 1 até n

prob(A[1..j] tem exatamente t elementos maiores que A[j])

Para construir qualquer (e toda!) permutação A[1..n] de 1 a n tq A[1..j] tem exatamente t elementos maiores que A[j], podemos:

$\in \{0, \dots, j-1\}$

1. escolher uniformemente um conjunto $S \subseteq \{1, \dots, n\}$ de j elementos para preencher as células de A[1..j]

estas células serão preenchidas com elementos de S prob($A[1..j]$ tem exatamente t elementos maiores que $A[j]$)

Para construir qualquer (e toda!) permutação $A[1..n]$ de 1 a n tq $A[1..j]$ tem exatamente t elementos maiores que $A[j]$,

$\in \{0, \ldots, j-1\}$

podemos:

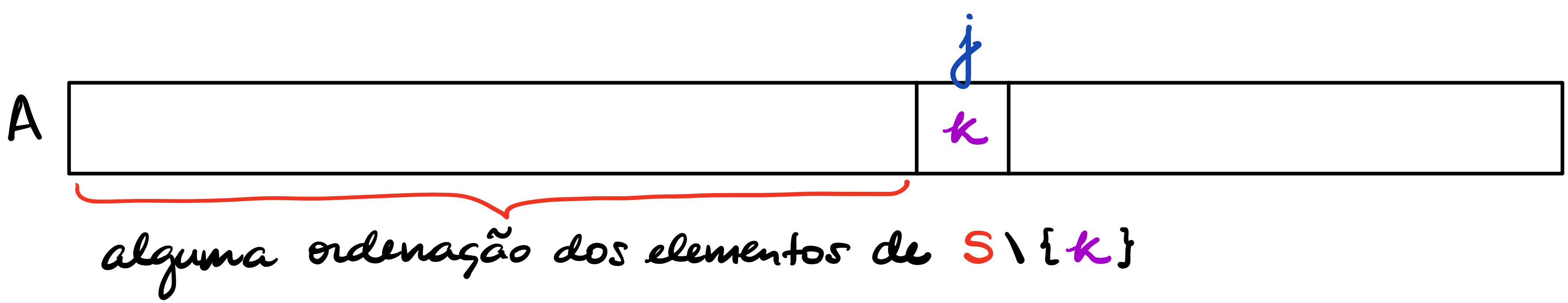
1. escolher uniformemente um conjunto $S \subseteq \{1, \ldots, n\}$ de j elementos para preencher as células de $A[1..j]$
2. escolher o único $k \in S$ t.q. S tem exatamente t elementos maiores que k, e tomar $A[j] \leftarrow k$

prob(A[1..j] tem exatamente t elementos maiores que A[j])

Para construir qualquer (e toda!) permutação A[1..n] de 1 a n tq A[1..j] tem exatamente t elementos maiores que A[j], podemos:

$\leftarrow \in \{0, \dots, j-1\}$

1. escolher uniformemente um conjunto $S \subseteq \{1, \dots, n\}$ de j elementos para preencher as células de A[1..j]
2. escolher o único $k \in S$ t.q. S tem exatamente t elementos maiores que k, e tomar $A[j] \leftarrow k$
3. preencher A[1..j−1] com os elementos de $S \setminus \{k\}$

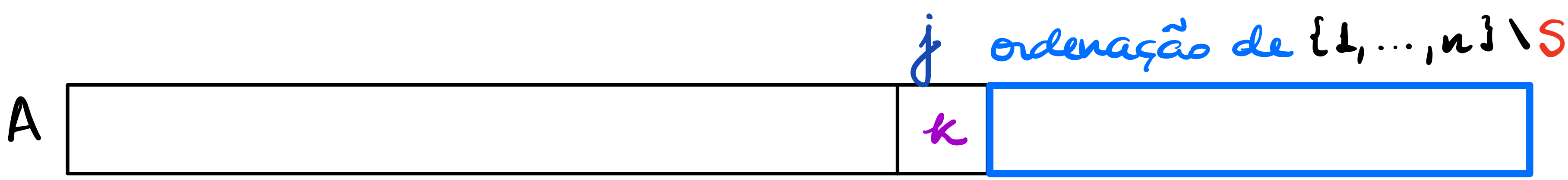


prob(A[1..j] tem exatamente t elementos maiores que A[j])

Para construir qualquer (e toda!) permutação A[1..n] de 1 a n tq A[1..j] tem exatamente t elementos maiores que A[j], podemos:

$t \in \{0, \dots, j-1\}$

1. escolher uniformemente um conjunto $S \subseteq \{1, \dots, n\}$ de j elementos para preencher as células de A[1..j]
2. escolher o único $k \in S$ t.q. S tem exatamente t elementos maiores que k, e tomar $A[j] \leftarrow k$
3. preencher A[1..j−1] com os elementos de $S \setminus \{k\}$
4. preencher A[j+1..n] com os elementos de $\{1, \dots, n\} \setminus S$



prob(A[1..j] tem exatamente t elementos maiores que A[j])

1. escolher uniformemente um conjunto $S \subseteq \{1,\dots,n\}$ de j elementos para preencher as células de A[1..j]
2. escolher o único $k \in S$ t.q. S tem exatamente t elementos maiores que k, e tomar $A[j] \leftarrow k$
3. preencher A[1..j−1] com os elementos de $S \setminus \{k\}$
4. preencher A[j+1..n] com os elementos de $\{1,\dots,n\} \setminus S$

$$\text{probabilidade} = \frac{\binom{n}{j} \cdot 1 \cdot (j-1)!\,(n-j)!}{n!}$$

$$= \frac{\frac{\cancel{n!}}{j!\,\cancel{(n-j)!}} \cdot 1 \cdot (j-1)!\,\cancel{(n-j)!}}{\cancel{n!}} = \frac{1}{j}$$

Análise do caso médio

ORDENA-POR-INSERÇÃO (A, n)

1 **para** $j \leftarrow 2$ **até** n **faça**
2 $\quad chave \leftarrow A[j]$
3 $\quad i \leftarrow j - 1$
4 $\quad$ **enquanto** $i \geq 1$ **e** $A[i] > chave$ **faça**
5 $\quad\quad A[i+1] \leftarrow A[i]$ $\quad \triangleright$ desloca
6 $\quad\quad i \leftarrow i - 1$
7 $\quad A[i+1] \leftarrow chave$ $\quad \triangleright$ insere

Para cada valor de j, a linha 4 é executada de 1 a j vezes.

de ela $\quad\quad\quad\quad$ $\in \{1, \dots, j\}$

Qual é a probabilidade ~~dela~~ ser executada t vezes?

Esta probabilidade é $1/j$.

Análise do caso médio

Para cada valor de j, a linha 4 é executada de 1 a j vezes.

Qual é a probabilidade dela ser executada t vezes?

Esta probabilidade é $1/j$.

Portanto o número esperado de execuções da linha 4 é

$$\sum_{t=1}^{j} t\,\frac{1}{j} \;=\; \frac{1}{j}\sum_{t=1}^{j} t \;=\; \frac{1}{j}\,\frac{(j+1)j}{2} \;=\; \frac{j+1}{2}.$$

E o número esperado de execuções da linha 4 no total é

$$\sum_{j=2}^{n} \frac{j+1}{2} \;=\; \frac{1}{2}\sum_{j=2}^{n}(j+1) \;=\; \frac{(n+4)(n-1)}{4} \;=\; \Theta(n^2).$$

Ordenação

$A[1\mathinner{\ldotp\ldotp} n]$ é crescente se $A[1] \leq \cdots \leq A[n]$.

Problema: Rearranjar um vetor $A[1\mathinner{\ldotp\ldotp} n]$ de modo que ele fique crescente.

ORDENA-POR-INSERÇÃO consome tempo $\Theta(n^2)$ no pior caso e $\Theta(n^2)$ no caso médio.

podemos falar isso? ✓

Podemos falar que o consumo de tempo no melhor caso é

$\Omega(n)$? ✓

$\Theta(n)$? ✓

$\Omega(1)$? ✓

$\Theta(1)$? ✗

Ordenação

$A[1\,..\,n]$ é crescente se $A[1] \leq \cdots \leq A[n]$.

Problema: Rearranjar um vetor $A[1\,..\,n]$ de modo que ele fique crescente.

ORDENA-POR-INSERÇÃO consome tempo $\mathrm{O}(n^2)$ no pior caso e $\Theta(n^2)$ no caso médio.

Conseguimos fazer melhor?

Sim! Usando divisão e conquista!