

Aula 09

Algoritmo de Seleção

Prof. Marco Aurélio Stefanos
marco em dct.ufms.br
www.dct.ufms.br/~marco

Baseado em slides do Prof. Paulo Fiofilof

Aula 09 – p. 1

i -ésimo menor elemento

Problema: Encontrar o i -ésimo menor elemento de $A[1, \dots, n]$

Suponha $A[1..n]$ sem elementos repetidos.

Exemplo: 33 é o 4o. menor elemento de:

1									10
22	99	32	88	34	33	11	97	55	66

A

1			4						10
11	22	32	33	34	55	66	88	97	99

ordenado

Aula 09 – p. 2

Mediana

Mediana é o $\lfloor \frac{n+1}{2} \rfloor$ -ésimo menor ou o $\lceil \frac{n+1}{2} \rceil$ -ésimo menor elemento

Exemplo: a mediana é 34 ou 55:

1									10
22	99	32	88	34	33	11	97	55	66

A

1				5	6				10
11	22	32	33	34	55	66	88	97	99

ordenado

Aula 09 – p. 3

Menor

Recebe um vetor $A[1..n]$ e devolve o valor do **menor** elemento.

MENOR (A, n)

```
1: menor ← A[1]
2: for k ← 2 TO n do
3:   if A[k] < menor then
4:     menor ← A[k]
5: devolva menor
```

O consumo de tempo do algoritmo **MENOR** é $\Theta(n)$.

Aula 09 – p. 4

Segundo menor

Recebe um vetor $A[1..n]$ e devolve o valor do **segundo menor** elemento, supondo $n \geq 2$.

SEG-MENOR (A, n)

```
1: menor ← min{A[1], A[2]}
2: segmenor ← max{A[1], A[2]}
3: for k ← 3 TO n do
4:   if A[k] < menor then
5:     segmenor ← menor
6:     menor ← A[k]
7:   else
8:     if A[k] < segmenor then
9:       segmenor ← A[k]
10: devolva segmenor
```

O consumo de tempo do algoritmo **SEG-MENOR** é $\Theta(n)$.

i-ésimo menor

Recebe $A[1..n]$ e i tal que $1 \leq i \leq n$
e devolve valor do i -ésimo menor elemento de $A[1..n]$

SELECT-ORD (A, n, i)

ORDENE (A, n)

devolva $A[i]$

O consumo de tempo do algoritmo **SELECT-ORD** é $\Theta(n \lg n)$.

Particione

Rearranja $A[p..r]$ de modo que $p \leq q \leq r$ e

$A[p..q-1] \leq A[q] < A[q+1..r]$

PARTICIONE (A, p, r)

$x \leftarrow A[r]$ $\triangleright x$ é o “pivô”

$i \leftarrow p-1$

for $j \leftarrow p$ TO $r-1$ do

if $A[j] \leq x$ then

$i \leftarrow i+1$

$A[i] \leftrightarrow A[j]$

$A[i+1] \leftrightarrow A[r]$

devolva $i+1$

	p								r	
A	99	33	55	77	11	22	88	66	33	44

Particione

Rearranja $A[p..r]$ de modo que $p \leq q \leq r$ e

$A[p..q-1] \leq A[q] < A[q+1..r]$

PARTICIONE (A, p, r)

$x \leftarrow A[r]$ $\triangleright x$ é o “pivô”

$i \leftarrow p-1$

for $j \leftarrow p$ TO $r-1$ do

if $A[j] \leq x$ then

$i \leftarrow i+1$

$A[i] \leftrightarrow A[j]$

$A[i+1] \leftrightarrow A[r]$

devolva $i+1$

	p			q					r	
A	11	22	33	33	44	55	88	66	77	99

Particione

Rearranja $A[p..r]$ de modo que $p \leq q \leq r$ e

$A[p..q-1] \leq A[q] < A[q+1..r]$

PARTICIONE (A, p, r)

$x \leftarrow A[r] \quad \triangleright x$ é o “pivô”

$i \leftarrow p-1$

for $j \leftarrow p$ **TO** $r-1$ **do**

if $A[j] \leq x$ **then**

$i \leftarrow i+1$

$A[i] \leftrightarrow A[j]$

$A[i+1] \leftrightarrow A[r]$

 devolva $i+1$

O algoritmo **PARTICIONE** consome tempo $\Theta(n)$.

Aula 09 – p. 9

Algoritmo SELECT

Select (A, p, r, i)

if $p = r$ **then**

 devolva $A[p]$

$q \leftarrow \text{PARTICIONE}(A, p, r)$

$k \leftarrow q - p + 1$

if $k = i$ **then**

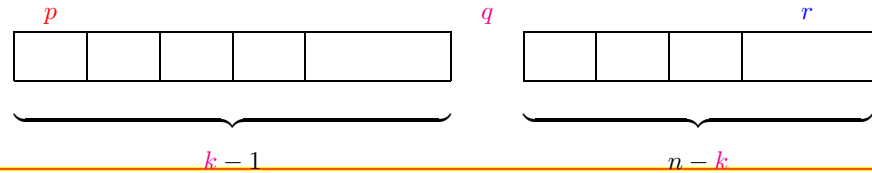
 devolva $A[q]$

if $k > i$ **then**

 devolva Select ($A, p, q-1, i$)

else

 devolva Select ($A, q+1, r, i-k$)



Aula 09 – p. 11

Algoritmo SELECT

Recebe $A[p..r]$ e i tal que $1 \leq i \leq r-p+1$

e devolve valor do i -ésimo menor elemento de $A[p..r]$

Select (A, p, r, i)

1: **if** $p = r$ **then**

2: devolva $A[p]$

3: $q \leftarrow \text{PARTICIONE}(p, r)$

4: $k \leftarrow q - p + 1$

5: **if** $k = i$ **then**

6: devolva $A[q]$

7: **if** $k > i$ **then**

8: devolva Select ($A, p, q-1, i$)

9: **else**

10: devolva Select ($A, q+1, r, i-k$)

Aula 09 – p. 10

Consumo de tempo

$T(n)$ = consumo de tempo máximo quando $n = r - p + 1$

linha	consumo de todas as execuções da linha
1-2	$= 2 \Theta(1)$
3	$= \Theta(n)$
4-7	$= 4 \Theta(1)$
8	$= T(k-1)$
10	$= T(n-k)$

$$\begin{aligned} T(n) &= \Theta(n+6) + \max\{T(k-1), T(n-k)\} \\ &= \Theta(n) + \max\{T(k-1), T(n-k)\} \end{aligned}$$

Aula 09 – p. 12

Consumo de tempo

$T(n)$ pertence a mesma classe Θ que:

$$T'(1) = 1$$

$$T'(n) = T'(n-1) + n \text{ para } n = 2, 3, \dots$$

Solução assintótica: $T'(n)$ é $\Theta(n^2)$

Solução exata:

$$T'(n) = \frac{n^2}{2} + \frac{n}{2}.$$

Algumas conclusões

No melhor caso o tempo do algoritmo Select é $\Theta(n)$.

No pior caso o tempo do algoritmo Select é $\Theta(n^2)$.

Consumo médio?

$$E[T(n)] = ???$$

Exemplos

Número médio de comparações sobre todas as permutações de $A[p..r]$ (supondo que nas linhas 8 e 10 o algoritmo sempre escolhe o lado maior):

$A[p..r]$	comps	$A[p..r]$	comps
1,2	1+0	1,2,3	2+1
2,1	1+0	2,1,3	2+1
média	2/2	1,3,2	2+0
		3,1,2	2+0
		2,3,1	2+1
		3,2,1	2+1
		média	16/6

Mais exemplos

$A[p..r]$	comps	$A[p..r]$	comps
1,2,3,4	3+3	1,3,4,2	3+1
2,1,3,4	3+3	3,1,4,2	3+1
1,3,2,4	3+2	1,4,3,2	3+1
3,1,2,4	3+2	4,1,3,2	3+1
2,3,1,4	3+3	3,4,1,2	3+1
3,2,1,4	3+3	4,3,1,2	3+1
1,2,4,3	3+1	2,3,4,1	3+3
2,1,4,3	3+1	3,2,4,1	3+3
1,4,2,3	3+1	2,4,3,1	3+2
4,1,2,3	3+1	4,2,3,1	3+2
2,4,1,3	3+1	3,4,2,1	3+3
4,2,1,3	3+1	4,3,2,1	3+3
		média	116/24

Ainda exemplos

No caso $r - p + 1 = 5$, a média é $864/120$.

n	$E[T(n)]$	$\cong$
1	0	0
2	2/2	1
3	16/6	2.7
4	116/24	4.8
5	864/120	7.2

Número de comparações

No pior caso $C(n)$ pertence a mesma classe Θ que:

$$C'(1) = 0$$

$$C'(n) = C'(n-1) + n - 1 \text{ para } n = 3, 4, \dots$$

Solução assintótica: $C'(n)$ é $\Theta(n^2)$

Solução exata:

$$C'(n) = \frac{n^2}{2} - \frac{n}{2}.$$

Número de comparações

O consumo de tempo assintótico é proporcional a

$C(n)$ = número de comparações entre elementos de A
quando $n = r - p + 1$

linha	consumo de todas as execuções da linha
1-2	= 0
3	= $n - 1$
4-7	= 0
8	= $C(k - 1)$
10	= $C(n - k)$

$$\text{total} \leq \max\{C(k-1), C(n-k)\} + n - 1$$

Particione Probabilístico

Rearranja $A[p..r]$ de modo que $p \leq q \leq r$ e

$$A[p..q-1] \leq A[q] < A[q+1..r]$$

PARTICIONE-PROB(A, p, r)

$i \leftarrow \text{RANDOM}(p, r)$

$A[i] \leftrightarrow A[r]$

devolva **PARTICIONE**(A, p, r)

O algoritmo **PARTICIONE-PROB** consome tempo $\Theta(n)$.

SELECT-PROBABILÍSTICO

Recebe $A[p..r]$ e i tal que $1 \leq i \leq r-p+1$
e devolve valor do i -ésimo menor elemento de $A[p..r]$
SelectProb(A, p, r, i)

```
if  $p = r$  then
  devolva  $A[p]$ 
 $q \leftarrow$  PARTICIONE-PROB( $A, p, r$ )
 $k \leftarrow q - p + 1$ 
if  $k = i$  then
  devolva  $A[q]$ 
if  $k > i$  then
  devolva SelectProb( $A, p, q - 1, i$ )
else
  devolva SelectProb( $A, q + 1, r, i - k$ )
```

Aula 09 – p. 21

Consumo de tempo

O consumo de tempo é proporcional a

$T(n)$ = número de comparações entre elementos de A
quando $n = r - p + 1$

linha	consumo de todas as execuções da linha
-------	--

1-2	= 0
3	= $n - 1$
4-7	= 0
8	= $T(k - 1)$
10	= $T(n - k)$

total $\leq \max\{T(k - 1), T(n - k)\} + n - 1$

$T(n)$ é uma **variável aleatória**.

Aula 09 – p. 22

Consumo de tempo

$$T(1) = 0$$

$$T(n) \leq \sum_{h=1}^{n-1} X_h T(h) + n - 1 \text{ para } n = 2, 3, \dots$$

onde

$$X_h = \begin{cases} 1 & \text{se } \max\{k - 1, n - k\} = h \\ 0 & \text{caso contrário} \end{cases}$$

Aula 09 – p. 23

$$\Pr\{X_h = 1\} = E[X_h]$$

$$X_h = \begin{cases} 1 & \text{se } \max\{k - 1, n - k\} = h \\ 0 & \text{caso contrário} \end{cases}$$

Qual a probabilidade de X_h valer 1?

Aula 09 – p. 24

$$\Pr\{X_h = 1\} = E[X_h]$$

$$X_h = \begin{cases} 1 & \text{se } \max\{k-1, n-k\} = h \\ 0 & \text{caso contrário} \end{cases}$$

Qual a probabilidade de X_h valer 1?

Para $h = 1, \dots, \lfloor n/2 \rfloor - 1$, $\Pr\{X_h = 1\} = 0 = E[X_h]$.

Para $h = \lceil n/2 \rceil, \dots, n$,

$$\Pr\{X_h=1\} = \frac{1}{n} + \frac{1}{n} = \frac{2}{n} = E[X_h]$$

Se n é ímpar e $h = \lfloor n/2 \rfloor$, então

$$\Pr\{X_h = 1\} = \frac{1}{n} = E[X_h]$$

Aula 09 – p. 24

Consumo de tempo esperado

$$E[T(1)] = 0$$

$$\begin{aligned} E[T(n)] &\leq \sum_{h=1}^{n-1} E[X_h T(h)] + n - 1 \\ &\leq \sum_{h=1}^{n-1} E[X_h] E[T(h)] + n - 1 \quad (\text{CLRS 9.2-2}) \\ &\leq \frac{2}{n} \sum_{h=a}^{n-1} E[T(h)] + n - 1 \quad \text{para } n = 2, 3, \dots \end{aligned}$$

onde $a = \lfloor n/2 \rfloor$.

Solução: $E[T(n)] = O(n)$.

Aula 09 – p. 25

Consumo de tempo esperado

$E[T(n)]$ pertence a mesma classe O que:

$$S(1) = 0$$

$$S(n) \leq \frac{2}{n} \sum_{h=a}^{n-1} S(h) + n - 1 \quad \text{para } n = 2, 3, \dots$$

onde $a = \lfloor n/2 \rfloor$.

n	1	2	3	4	5	6	7	8	9	10
$S(n)$	0.0	1.0	2.7	4.8	7.4	10.0	13.1	15.8	19.4	22.1
$4n$	4	8	12	16	20	24	28	32	36	40

Vamos verificar que $S(n) < 4n$ para $n = 1, 2, 3, 4, \dots$

Aula 09 – p. 26

Recorrência

Prova: Se $n = 1$, então $S(n) = 0 < 4 = 4 \cdot 1 = 4n$. Se $n \geq 2$,

$$\begin{aligned} S(n) &\leq \frac{2}{n} \sum_{h=a}^{n-1} S(h) + n - 1 \\ &\stackrel{\text{hi}}{<} \frac{2}{n} \sum_{h=a}^{n-1} 4h + n - 1 \\ &= \frac{8}{n} \left(\sum_{h=1}^{n-1} h - \sum_{h=1}^{a-1} h \right) + n - 1 \\ &= \frac{4}{n} ((n-1)n - (\lfloor n/2 \rfloor - 1)\lfloor n/2 \rfloor) + n - 1 \\ &\leq \frac{4}{n} ((n-1)n - (n/2 - 1)n/2) + n - 1 \end{aligned}$$

Aula 09 – p. 27

Recorrência

$$\begin{aligned} S(n) &\leq \frac{4}{n} \left(n^2 - n - \frac{n^2}{4} + n/2 \right) + n - 1 \\ &= \frac{4}{n} \left(\frac{3n^2}{4} - \frac{n}{2} \right) + n - 1 \\ &= \frac{4}{n} \left(\frac{n}{2} \left(\frac{3n}{2} - 1 \right) \right) + n - 1 \\ &= 3n - 2 + n - 1 = 4n - 3 < 4n. \end{aligned}$$

Conclusão

O consumo de tempo esperado do algoritmo **SelectProb** é $O(n)$.

Aula 09 – p. 28

Exercícios

Exercício [CLRS 9.1-1] [muito bom!]

Mostre que o segundo menor elemento de um vetor $A[1..n]$ pode ser encontrado com não mais que $n + \lceil \lg n \rceil - 2$ comparações.

Exercício

Prove que o algoritmo Select Probabilístico (= Randomized Select) funciona corretamente.

Exercício [CLRS 9.2-3]

Escreva uma versão iterativa do algoritmo Select Probabilístico (= Randomized Select).

Aula 09 – p. 29