Análise de Algoritmos

Slides de Paulo Feofiloff

[com erros do coelho e agora também da cris]

i-ésimo menor elemento

CLRS 9

i-ésimo menor

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

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

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

Menor

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

```
MENOR (A, n)
1    menor ← A[1]
2    para k ← 2 até n faça
3         se A[k] < menor
4              então menor ← A[k]
5    devolva menor
```

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

Segundo menor

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

SEG-MENOR (A, n)

```
menor ← min{A[1], A[2]}      segmenor ← max{A[1], A[2]}
para k ← 3 até n faça
      se A[k] < menor
            então segmenor ← menor
                  menor ← A[k]
      senão se A[k] < segmenor
            então segmenor ← A[k]
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)
1    ORDENE (A, n)
2    devolva A[i]
```

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

Particione

Rearranja $A[p\,..\,d]$ de modo que $p \le q \le d$ e
$A[p\,..\,q{-}1] \le A[q] < A[q{+}1\,..\,d]$

```
PARTICIONE (A, p, d)
1   x ← A[d]            ▷ x é o "pivô"
2   i ← p−1
3   para j ← p até d − 1 faça
4         se A[j] ≤ x
5               então i ← i + 1
6                     A[i] ↔ A[j]
7   A[i+1] ↔ A[d]
8   devolva i + 1
```

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

Particione

Rearranja $A[p\,..\,d]$ de modo que $p \le q \le d$ e
$A[p\,..\,q{-}1] \le A[q] < A[q{+}1\,..\,d]$

```
PARTICIONE (A, p, d)
1   x ← A[d]            ▷ x é o "pivô"
2   i ← p−1
3   para j ← p até d − 1 faça
4         se A[j] ≤ x
5               então i ← i + 1
6                     A[i] ↔ A[j]
7   A[i+1] ↔ A[d]
8   devolva i + 1
```

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

Particione

Rearranja $A[p\,..\,d]$ de modo que $p \le q \le d$ e
$A[p\,..\,q{-}1] \le A[q] < A[q{+}1\,..\,d]$

```
PARTICIONE (A, p, d)
    x ← A[d]                ▷ x é o "pivô"
    i ← p−1
    para j ← p até d − 1 faça
        se A[j] ≤ x
            então i ← i + 1
                  A[i] ↔ A[j]
    A[i+1] ↔ A[d]
    devolva i + 1
```

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

Algoritmo SELECT

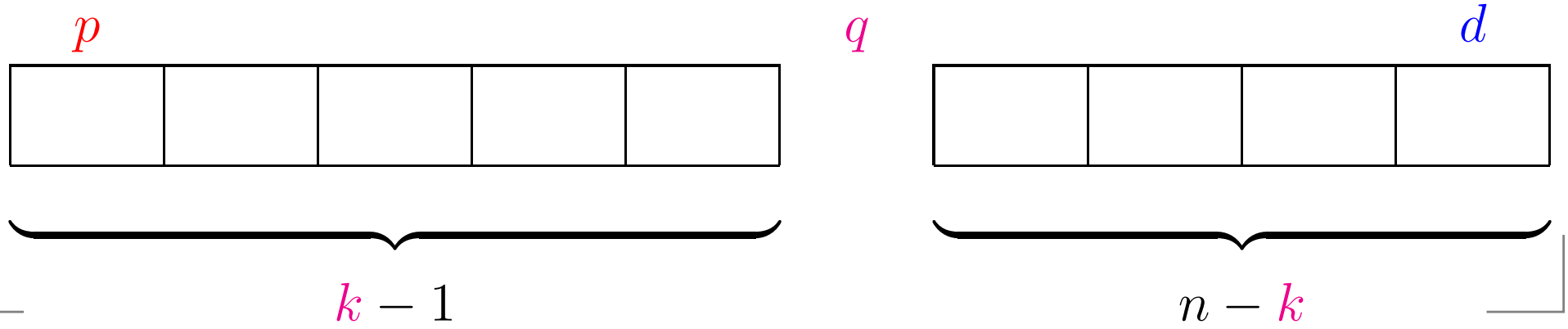
Recebe $A[p\,..\,d]$ e i tal que $1 \leq i \leq d-p+1$
e devolve valor do i-ésimo menor elemento de $A[p\,..\,d]$

SELECT(A, p, d, i)

se $p = d$
    então devolva $A[p]$
$q \leftarrow$ PARTICIONE (p, d)
$k \leftarrow q - p + 1$
se $k = i$
    então devolva $A[q]$
se $k > i$
    então devolva SELECT $(A, p, q-1, i)$
    senão devolva SELECT $(A, q+1, d, i-k)$

Algoritmo SELECT

```
SELECT(A, p, d, i)
    se p = d
        então devolva A[p]
    q ← PARTICIONE (A, p, d)
    k ← q − p + 1
    se k = i
        então devolva A[q]
    se k > i
        então devolva SELECT (A, p, q − 1, i)
        senão devolva SELECT (A, q + 1, d, i − k)
```



Consumo de tempo

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

linha	consumo de todas as execuções da linha
1-2	$= 2\,\Theta(1)$
3	$= \Theta(n)$
4-7	$= 4\,\Theta(1)$
8	$= T(k-1)$
9	$= T(n-k)$

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

Consumo de tempo

$T(n)$ pertence a mesma classe Θ que:

$$T'(1) = 1$$

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

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 consumo de tempo do algoritmo SELECT é $\Theta(n)$.

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

Consumo médio?

$\mathrm{E}[T(n)] =$???

Exemplos

Número médio de comparações sobre todas as permutações de $A[p\,..\,d]$ (supondo que nas linhas 8 e 9 o algoritmo sempre escolhe o lado maior):

$A[p\,..\,d]$	comps
1,2	1+0
2,1	1+0
média	2/2

$A[p\,..\,d]$	comps
1,2,3	2+1
2,1,3	2+1
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

$A[p\,..\,d]$	comps
1,2,3,4	3+3
2,1,3,4	3+3
1,3,2,4	3+2
3,1,2,4	3+2
2,3,1,4	3+3
3,2,1,4	3+3
1,2,4,3	3+1
2,1,4,3	3+1
1,4,2,3	3+1
4,1,2,3	3+1
2,4,1,3	3+1
4,2,1,3	3+1

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

Ainda exemplos

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

n	$\mathrm{E}[T(n)]$	SIM
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

O consumo de tempo assintótico é proporcional a

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

linha	consumo de todas as execuções da linha
1-2	$= 0$
3	$= n - 1$
4-7	$= 0$
8	$= C(k-1)$
9	$= C(n-k)$
total	$\leq \max\{C(k-1), C(n-k)\} + n - 1$

Número de comparações

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

$$C'(1) = 0$$

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

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

Solução exata:

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

Particione aleatorizado

Rearranja $A[p\,..\,d]$ de modo que $p \leq q \leq d$ e
$A[p\,..\,q{-}1] \leq A[q] < A[q{+}1\,..\,d]$

```
PARTICIONE-ALEA(A, p, d)
1    i ← RANDOM (p, d)
2    A[i] ↔ A[d]
3    devolva PARTICIONE (A, p, d)
```

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

SELECT-ALEATORIZADO (= randomized select)

Recebe $A[p\,..\,d]$ e i tal que $1 \le i \le d-p+1$
e devolve valor do i-ésimo menor elemento de $A[p\,..\,d]$

```
SELECT-ALEA(A, p, d, i)
    se p = d
        então devolva A[p]
    q ← PARTICIONE-ALEA (A, p, d)
    k ← q − p + 1
    se k = i
        então devolva A[q]
    se k > i
        então devolva SELECT-ALEA (A, p, q − 1, i)
        senão devolva SELECT-ALEA (A, q + 1, d, i − k)
```

Consumo de tempo

O consumo de tempo é proporcional a

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

linha	consumo de todas as execuções da linha
1-2	$= 0$
3	$= n - 1$
4-7	$= 0$
8	$= T(k-1)$
9	$= T(n-k)$
total	$\leq \max\{T(k-1), T(n-k)\} + n - 1$

$T(n)$ é uma variável aleatória.

Consumo de tempo

$$T(1) = 0$$

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

onde

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

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

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

Qual a probabilidade de X_h valer 1?

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

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

Qual a probabilidade de X_h valer 1?

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

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

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

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

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

Consumo de tempo esperado

$$\mathrm{E}[T(1)] = 0$$

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

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

Solução: $\mathrm{E}[T(n)] = \mathrm{O}(n)$.

Consumo de tempo esperado

$\mathrm{E}[T(n)]$ pertence a mesma classe O que:

$$S(1) = 0$$

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

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 $S(n) < 4\,n$ para $n = 1, 2, 3, 4, \ldots$

Recorrência

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

$$
\begin{aligned}
S(n) &\leq \frac{2}{n} \sum_{h=a}^{n-1} S(h) + n - 1 \\
&\overset{\text{hi}}{<} \frac{2}{n} \sum_{h=a}^{n-1} 4\,h + n - 1 \\
&= \frac{8}{n} \left(\sum_{h=1}^{n-1} h - \sum_{h=1}^{a-1} h \right) + n - 1 \\
&\leq \frac{4}{n} \left(n^2 - n - \frac{(n-1)(n-3)}{4} \right) + n - 1 \\
&= \frac{4}{n} \left(\frac{3n^2}{4} - \frac{3}{4} \right) + n - 1 \\
&= 3n - \frac{3}{n} + n - 1 = 4n - \frac{3}{n} - 1 < 4n.
\end{aligned}
$$

Conclusão

O consumo de tempo esperado do algoritmo SELECT-ALEA é $\mathrm{O}(n)$.

Exercícios

Exercício 1 [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 2
Prove que o algoritmo Select Aleatorizado (= Randomized Select) funciona corretamente.

Exercício 3 [CLRS 9.2-3]
Escreva uma versão iterativa do algoritmo Select Aleatorizado (= Randomized Select).

Exercício 4
Suponha que concluímos que $\mathrm{E}[T(n)] = O(n)$ de alguma maneira. Deduza, como fizemos no início da aula de hoje, um valor de $c > 0$ para o qual vale que $\mathrm{E}[T(n)] \leq cn$ para $n \geq 1$.