Seleção da Mediana

Análise de Algoritmos – Ciência da Computação

Prof. Daniel Saad Nogueira Nunes

Instituto Federal de Brasília,
Câmpus Taguatinga

INSTITUTO FEDERAL
Brasília

Sumário

Introdução

Seleção do k-ésimo

Suponha que você queira selecionar o k-ésimo menor elemento dentro de uma sequência de elementos $(a_0, a_1, \ldots, a_{n-1})$.

Introdução

Seleção do k-ésimo

Suponha que você queira selecionar o k-ésimo menor elemento dentro de uma sequência de elementos $(a_0, a_1, \ldots, a_{n-1})$.

Solução Força-Bruta

Ordenar e selecionar o k-ésimo.

$$\Theta(n \lg n)$$

Relembrando o Quicksort

Partition

Podemos utilizar o Partition para selecionar o k-ésimo menor.

Relembrando o Quicksort

Relembrando o Quicksort

Relembrando o Quicksort

Complexidade do Naive-Select

No pior caso, temos:

$$T(n) = T(n-1) + \Theta(n)$$
$$T(n) \in \Theta(n^2)$$

Relembrando o Quicksort

Complexidade do NAIVE-SELECT

No pior caso, temos:

$$T(n) = T(n-1) + \Theta(n)$$
$$T(n) \in \Theta(n^2)$$

Pior que o força-bruta!

Qual o problema do NAIVE-SELECT?

Relembrando o Quicksort

Complexidade do Naive-Select

No pior caso, temos:

$$T(n) = T(n-1) + \Theta(n)$$
$$T(n) \in \Theta(n^2)$$

O problema é a escolha do pivô!

Seleção do k-ésimo

Seleção do k-ésimo

1. Divida o vetor original V em vetores $V_i = V[i, (i+1)*5-1]$. Ao todo, teremos $\lfloor n/5 \rfloor$ vetores de tamanho 5 e um vetor de tamanho $n \mod 5$.
2. Ordene cada um dos vetores V_i.
3. Ache a mediana de cada um dos vetores V_i.
4. Use o procedimento recursivo para encontrar a mediana das medianas.
5. Use Partiton usando como **pivô** as medianas das medianas. Se o k-ésimo for o próprio pivô, retorne o próprio pivô, caso contrário, devemos usar o procedimento recursivo para encontrar o k-ésimo.
6. Se k é menor que a posição do pivô x, use o procedimento recursivo para encontrar o k-ésimo na partição mais baixa. Caso contrário, use o procedimento recursivo para encontrar o $x-k$-ésimo na partição mais alta.

Seleção do k-ésimo

Seleção do k-ésimo

1. Divida o vetor original V em vetores $V_i = V[i, (i+1)*5-1]$. Ao todo, teremos $\lfloor n/5 \rfloor$ vetores de tamanho 5 e um vetor de tamanho $n \mod 5$.
2. Ordene cada um dos vetores V_i.
3. Ache a mediana de cada um dos vetores V_i.
4. Use o procedimento recursivo para encontrar a mediana das medianas.
5. Use PARTITON usando como **pivô** as medianas das medianas. Se o k-ésimo for o próprio pivô, retorne o próprio pivô, caso contrário, devemos usar o procedimento recursivo para encontrar o k-ésimo.
6. Se k é menor que a posição do pivô x, use o procedimento recursivo para encontrar o k-ésimo na partição mais baixa. Caso contrário, use o procedimento recursivo para encontrar o $x-k$-ésimo na partição mais alta.

Seleção do k-ésimo

Seleção do k-ésimo

1. Divida o vetor original V em vetores $V_i = V[i, (i+1)*5-1]$. Ao todo, teremos $\lfloor n/5 \rfloor$ vetores de tamanho 5 e um vetor de tamanho $n \mod 5$.
2. Ordene cada um dos vetores V_i.
3. Ache a mediana de cada um dos vetores V_i.
4. Use o procedimento recursivo para encontrar a mediana das medianas.
5. Use PARTITON usando como **pivô** as medianas das medianas. Se o k-ésimo for o próprio pivô, retorne o próprio pivô, caso contrário, devemos usar o procedimento recursivo para encontrar o k-ésimo.
6. Se k é menor que a posição do pivô x, use o procedimento recursivo para encontrar o k-ésimo na partição mais baixa. Caso contrário, use o procedimento recursivo para encontrar o $x-k$-ésimo na partição mais alta.

Seleção do k-ésimo

Seleção do k-ésimo

1. Divida o vetor original V em vetores $V_i = V[i, (i+1) * 5 - 1]$. Ao todo, teremos $\lfloor n/5 \rfloor$ vetores de tamanho 5 e um vetor de tamanho $n \mod 5$.
2. Ordene cada um dos vetores V_i.
3. Ache a mediana de cada um dos vetores V_i.
4. Use o procedimento recursivo para encontrar a mediana das medianas.
5. Use PARTITON usando como **pivô** as medianas das medianas. Se o k-ésimo for o próprio pivô, retorne o próprio pivô, caso contrário, devemos usar o procedimento recursivo para encontrar o k-ésimo.
6. Se k é menor que a posição do pivô x, use o procedimento recursivo para encontrar o k-ésimo na partição mais baixa. Caso contrário, use o procedimento recursivo para encontrar o $x - k$-ésimo na partição mais alta.

Seleção do k-ésimo

Seleção do k-ésimo

1. Divida o vetor original V em vetores $V_i = V[i, (i+1)*5-1]$. Ao todo, teremos $\lfloor n/5 \rfloor$ vetores de tamanho 5 e um vetor de tamanho $n \mod 5$.
2. Ordene cada um dos vetores V_i.
3. Ache a mediana de cada um dos vetores V_i.
4. Use o procedimento recursivo para encontrar a mediana das medianas.
5. Use PARTITON usando como **pivô** as medianas das medianas. Se o k-ésimo for o próprio pivô, retorne o próprio pivô, caso contrário, devemos usar o procedimento recursivo para encontrar o k-ésimo.
6. Se k é menor que a posição do pivô x, use o procedimento recursivo para encontrar o k-ésimo na partição mais baixa. Caso contrário, use o procedimento recursivo para encontrar o $x-k$-ésimo na partição mais alta.

Seleção do k-ésimo

Seleção do k-ésimo

1. Divida o vetor original V em vetores $V_i = V[i, (i+1)*5-1]$. Ao todo, teremos $\lfloor n/5 \rfloor$ vetores de tamanho 5 e um vetor de tamanho $n \mod 5$.
2. Ordene cada um dos vetores V_i.
3. Ache a mediana de cada um dos vetores V_i.
4. Use o procedimento recursivo para encontrar a mediana das medianas.
5. Use PARTITON usando como **pivô** as medianas das medianas. Se o k-ésimo for o próprio pivô, retorne o próprio pivô, caso contrário, devemos usar o procedimento recursivo para encontrar o k-ésimo.
6. Se k é menor que a posição do pivô x, use o procedimento recursivo para encontrar o k-ésimo na partição mais baixa. Caso contrário, use o procedimento recursivo para encontrar o $x-k$-ésimo na partição mais alta.

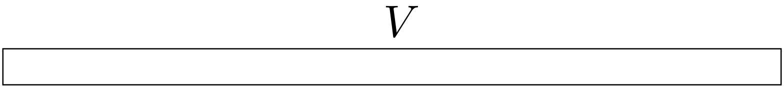
Seleção do k-ésimo

Seleção do k-ésimo

1. Divida o vetor original V em vetores $V_i = V[i, (i+1) * 5 - 1]$. Ao todo, teremos $\lfloor n/5 \rfloor$ vetores de tamanho 5 e um vetor de tamanho $n \mod 5$.
2. Ordene cada um dos vetores V_i.
3. Ache a mediana de cada um dos vetores V_i.
4. Use o procedimento recursivo para encontrar a mediana das medianas.
5. Use PARTITON usando como **pivô** as medianas das medianas. Se o k-ésimo for o próprio pivô, retorne o próprio pivô, caso contrário, devemos usar o procedimento recursivo para encontrar o k-ésimo.
6. Se k é menor que a posição do pivô x, use o procedimento recursivo para encontrar o k-ésimo na partição mais baixa. Caso contrário, use o procedimento recursivo para encontrar o $x - k$-ésimo na partição mais alta.

Seleção do k-ésimo

Seleção do k-ésimo

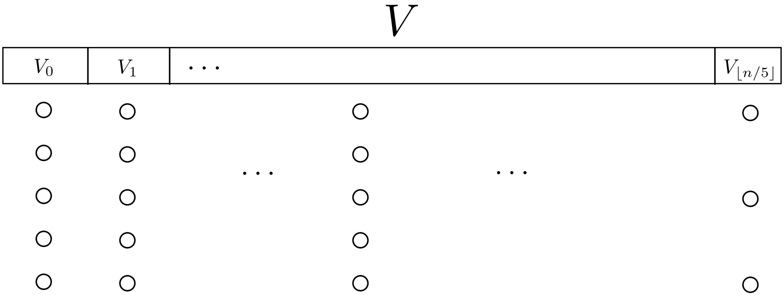


Seleção do k-ésimo

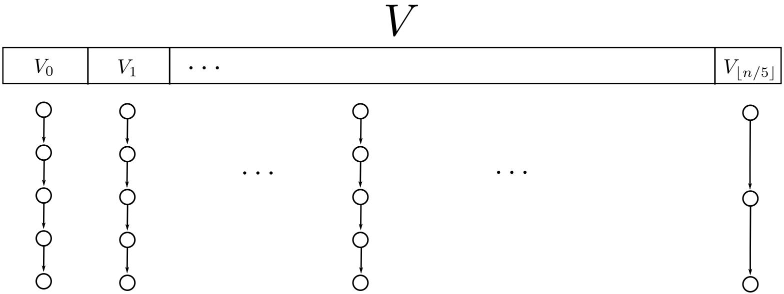
$$V$$

V_0	V_1	$\cdots$	$V_{\lfloor n/5 \rfloor}$

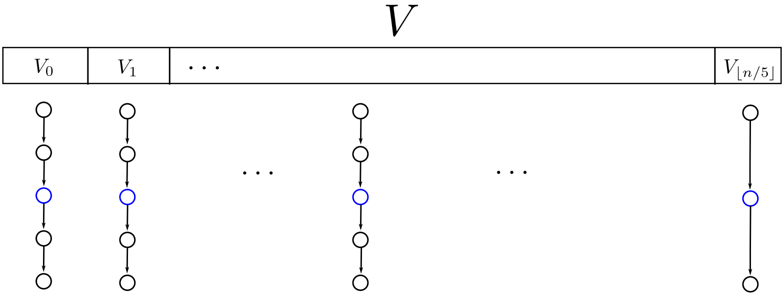
Seleção do k-ésimo



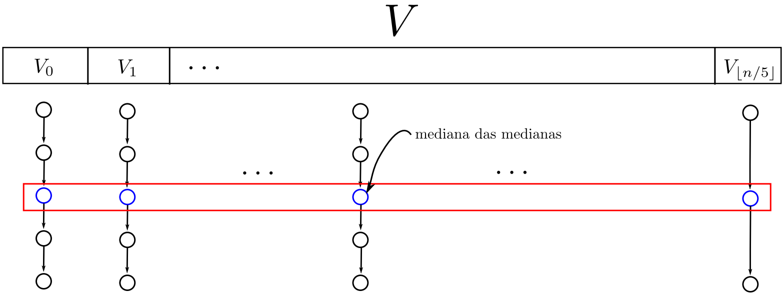
Seleção do k-ésimo



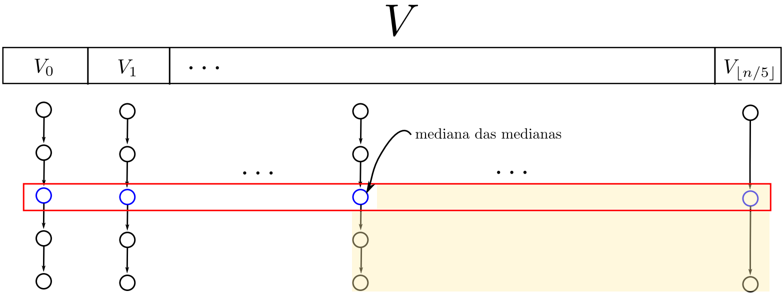
Seleção do k-ésimo



Seleção do k-ésimo



Seleção do k-ésimo



Análise

Análise do Algoritmo SELECT

1. Divida o vetor original V em vetores $V_i = V[i, (i+1)*5-1]$. Ao todo, teremos $\lfloor n/5 \rfloor$ vetores de tamanho 5 e um vetor de tamanho $n \mod 5$. $\Theta(n)$
2. Ordene cada um dos vetores V_i. Tempo $\Theta(1)$ para cada um, tempo $\Theta(n)$ para todos.
3. Ache a mediana de cada um dos vetores V_i. Achar a mediana de cada V_i leva tempo $\Theta(1)$. Logo, de todos, leva tempo $\Theta(n)$.
4. Use o procedimento recursivo para encontrar a mediana das medianas. $T(n/5)$.
5. Use PARITITON usando como **pivô** as medianas das medianas. Se o k-ésimo for o próprio pivô, retorne o próprio pivô, caso contrário, devemos usar o procedimento recursivo para encontrar o k-ésimo. $\Theta(1)$
6. Se k é menor que a posição do pivô x, use o procedimento recursivo para encontrar o k-ésimo na partição mais baixa. Caso contrário, use o procedimento recursivo para encontrar o $x-k$-ésimo na partição mais alta. $T(7n/10+6)$.

Análise

Análise do Algoritmo SELECT

1. Divida o vetor original V em vetores $V_i = V[i, (i+1) * 5 - 1]$. Ao todo, teremos $\lfloor n/5 \rfloor$ vetores de tamanho 5 e um vetor de tamanho $n \mod 5$. $\Theta(n)$
2. Ordene cada um dos vetores V_i. Tempo $\Theta(1)$ para cada um, tempo $\Theta(n)$ para todos.
3. Ache a mediana de cada um dos vetores V_i. Achar a mediana de cada V_i leva tempo $\Theta(1)$. Logo, de todos, leva tempo $\Theta(n)$.
4. Use o procedimento recursivo para encontrar a mediana das medianas. $T(n/5)$.
5. Use PARITITON usando como **pivô** as medianas das medianas. Se o k-ésimo for o próprio pivô, retorne o próprio pivô, caso contrário, devemos usar o procedimento recursivo para encontrar o k-ésimo. $\Theta(1)$
6. Se k é menor que a posição do pivô x, use o procedimento recursivo para encontrar o k-ésimo na partição mais baixa. Caso contrário, use o procedimento recursivo para encontrar o $x - k$-ésimo na partição mais alta. $T(7n/10 + 6)$.

Análise

Análise do Algoritmo SELECT

1. Divida o vetor original V em vetores $V_i = V[i, (i+1) * 5 - 1]$. Ao todo, teremos $\lfloor n/5 \rfloor$ vetores de tamanho 5 e um vetor de tamanho $n \mod 5$. $\Theta(n)$
2. Ordene cada um dos vetores V_i. Tempo $\Theta(1)$ para cada um, tempo $\Theta(n)$ para todos.
3. Ache a mediana de cada um dos vetores V_i. Achar a mediana de cada V_i leva tempo $\Theta(1)$. Logo, de todos, leva tempo $\Theta(n)$.
4. Use o procedimento recursivo para encontrar a mediana das medianas. $T(n/5)$.
5. Use PARITITON usando como **pivô** as medianas das medianas. Se o k-ésimo for o próprio pivô, retorne o próprio pivô, caso contrário, devemos usar o procedimento recursivo para encontrar o k-ésimo. $\Theta(1)$
6. Se k é menor que a posição do pivô x, use o procedimento recursivo para encontrar o k-ésimo na partição mais baixa. Caso contrário, use o procedimento recursivo para encontrar o $x - k$-ésimo na partição mais alta. $T(7n/10 + 6)$.

Análise

Análise do Algoritmo SELECT

1. Divida o vetor original V em vetores $V_i = V[i, (i+1)*5-1]$. Ao todo, teremos $\lfloor n/5 \rfloor$ vetores de tamanho 5 e um vetor de tamanho $n \mod 5$. $\Theta(n)$
2. Ordene cada um dos vetores V_i. Tempo $\Theta(1)$ para cada um, tempo $\Theta(n)$ para todos.
3. Ache a mediana de cada um dos vetores V_i. Achar a mediana de cada V_i leva tempo $\Theta(1)$. Logo, de todos, leva tempo $\Theta(n)$.
4. Use o procedimento recursivo para encontrar a mediana das medianas. $T(n/5)$.
5. Use PARITITON usando como **pivô** as medianas das medianas. Se o k-ésimo for o próprio pivô, retorne o próprio pivô, caso contrário, devemos usar o procedimento recursivo para encontrar o k-ésimo. $\Theta(1)$
6. Se k é menor que a posição do pivô x, use o procedimento recursivo para encontrar o k-ésimo na partição mais baixa. Caso contrário, use o procedimento recursivo para encontrar o $x-k$-ésimo na partição mais alta. $T(7n/10+6)$.

Análise

Análise do Algoritmo SELECT

1. Divida o vetor original V em vetores $V_i = V[i, (i+1) * 5 - 1]$. Ao todo, teremos $\lfloor n/5 \rfloor$ vetores de tamanho 5 e um vetor de tamanho $n \mod 5$. $\Theta(n)$
2. Ordene cada um dos vetores V_i. Tempo $\Theta(1)$ para cada um, tempo $\Theta(n)$ para todos.
3. Ache a mediana de cada um dos vetores V_i. Achar a mediana de cada V_i leva tempo $\Theta(1)$. Logo, de todos, leva tempo $\Theta(n)$.
4. Use o procedimento recursivo para encontrar a mediana das medianas. $T(n/5)$.
5. Use PARITITON usando como **pivô** as medianas das medianas. Se o k-ésimo for o próprio pivô, retorne o próprio pivô, caso contrário, devemos usar o procedimento recursivo para encontrar o k-ésimo. $\Theta(1)$
6. Se k é menor que a posição do pivô x, use o procedimento recursivo para encontrar o k-ésimo na partição mais baixa. Caso contrário, use o procedimento recursivo para encontrar o $x-k$-ésimo na partição mais alta. $T(7n/10+6)$.

Análise

Análise do Algoritmo SELECT

1. Divida o vetor original V em vetores $V_i = V[i, (i+1)*5-1]$. Ao todo, teremos $\lfloor n/5 \rfloor$ vetores de tamanho 5 e um vetor de tamanho $n \mod 5$. $\Theta(n)$
2. Ordene cada um dos vetores V_i. Tempo $\Theta(1)$ para cada um, tempo $\Theta(n)$ para todos.
3. Ache a mediana de cada um dos vetores V_i. Achar a mediana de cada V_i leva tempo $\Theta(1)$. Logo, de todos, leva tempo $\Theta(n)$.
4. Use o procedimento recursivo para encontrar a mediana das medianas. $T(n/5)$.
5. Use PARITITON usando como **pivô** as medianas das medianas. Se o k-ésimo for o próprio pivô, retorne o próprio pivô, caso contrário, devemos usar o procedimento recursivo para encontrar o k-ésimo. $\Theta(1)$
6. Se k é menor que a posição do pivô x, use o procedimento recursivo para encontrar o k-ésimo na partição mais baixa. Caso contrário, use o procedimento recursivo para encontrar o $x-k$-ésimo na partição mais alta. $T(7n/10+6)$.

Análise

Análise do Algoritmo Select

1. Divida o vetor original V em vetores $V_i = V[i, (i+1)*5-1]$. Ao todo, teremos $\lfloor n/5 \rfloor$ vetores de tamanho 5 e um vetor de tamanho $n \mod 5$. $\Theta(n)$
2. Ordene cada um dos vetores V_i. Tempo $\Theta(1)$ para cada um, tempo $\Theta(n)$ para todos.
3. Ache a mediana de cada um dos vetores V_i. Achar a mediana de cada V_i leva tempo $\Theta(1)$. Logo, de todos, leva tempo $\Theta(n)$.
4. Use o procedimento recursivo para encontrar a mediana das medianas. $T(n/5)$.
5. Use Parititon usando como **pivô** as medianas das medianas. Se o k-ésimo for o próprio pivô, retorne o próprio pivô, caso contrário, devemos usar o procedimento recursivo para encontrar o k-ésimo. $\Theta(1)$
6. Se k é menor que a posição do pivô x, use o procedimento recursivo para encontrar o k-ésimo na partição mais baixa. Caso contrário, use o procedimento recursivo para encontrar o $x-k$-ésimo na partição mais alta. $T(7n/10+6)$.

Análise

- A chave para análise é perguntar quantos elementos são menores que a mediana das medianas.
- Sabemos que metade das $\lceil \frac{n}{5} \rceil$ medianas são menores que a mediana das medianas.
- Considerando essa metade, um grupo contribui com dois elementos menores que a mediana da mediana e outro contribui com uma quantidade variável (1 a 3).

Análise

$$3\left(\left\lceil\frac{1}{2}\left\lceil\frac{n}{5}\right\rceil\right\rceil-2\right)\geq\frac{3n}{10}-6$$

Análise

Análise do Algoritmo SELECT

$$T(n) = \begin{cases} \Theta(1), & n < 140 \\ T(\lceil n/5 \rceil) + T(7n/10 + 6) + \Theta(n) \end{cases}$$

Análise

Análise do Algoritmo Select

$$T(n) = \begin{cases} \Theta(1), & n < 140 \\ T(\lceil n/5 \rceil) + T(7n/10 + 6) + \Theta(n) \end{cases}$$

$$\begin{aligned} T(n) &\leq c(\lceil n/5 \rceil) + c(7n/10 + 6) + an \\ &\leq cn/5 + c + 7cn/10 + 6c + an \\ &= 9cn/10 + 7c + an \\ &= cn + (-cn/10 + 7c + an) \end{aligned}$$

Resta provar que $-cn/10 + 7c + an \leq 0$.

Questões Interessantes

$$
\begin{aligned}
-cn/10 + 7c + an &\leq 0 \\
-cn + 70c + 10an \leq 0 \\
c(-n + 70) \leq -10an \\
c \leq \frac{-10an}{-n+70} \\
c \geq \frac{10an}{n-70}
\end{aligned}
$$

Se $c > 20a$, estamos prontos, uma vez que $n \geq 140$.

Questões Interessantes

- Como executar o QUICKSORT em tempo $\Theta(n \lg n)$?
- Se em vez de dividir o vetor em partições de tamanho 5, os dividíssemos em partições de tamanho 3, o resultado da análise seria diferente?
- Se em vez de dividir o vetor em partições de tamanho 5, os dividíssemos em partições de tamanho 7, o resultado da análise seria diferente?