

ORDENAÇÃO: *QUICKSORT*

PUC MINAS

ENGENHARIA DE SOFTWARE

ALGORITMO

- Algoritmo **recursivo** criado por Tony Hoare e publicado em 1962.
- Ideia básica:
 - dividir o problema da ordenação;
 - em ordenações de subconjuntos menores.
- Premissa do *quicksort*:
 - a partir de um elemento pivô;
 - formar um conjunto de maiores e outro de menores que o pivô.

ALGORITMO

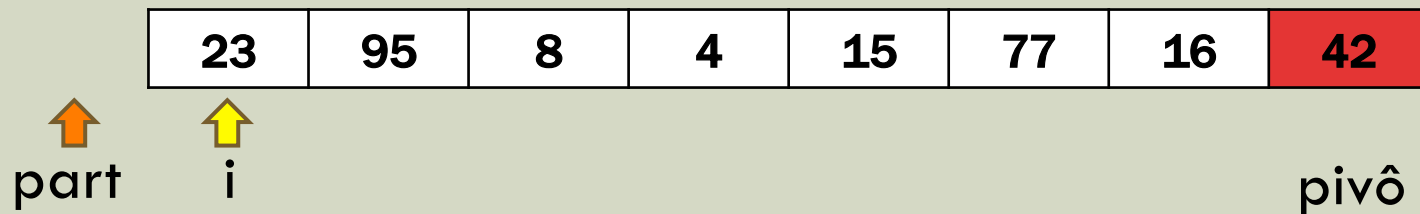
- Escolha o início e o fim da partição;
- Escolha um valor pivô;
- Examine todos os valores da partição separando:
 - maiores que o pivô à direita e menores à esquerda.
- Repita o procedimento para as partições resultantes;
 - que sejam maiores que um elemento.

QUICKSORT

23	95	8	4	15	77	16	42
----	----	---	---	----	----	----	----

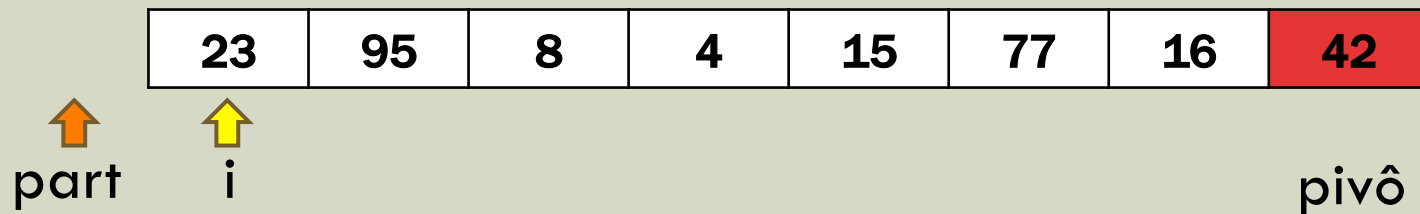
- Escolha do pivô:
 - arbitrária.
- Usaremos como pivô o último elemento do conjunto considerado. (Cormen)

QUICKSORT



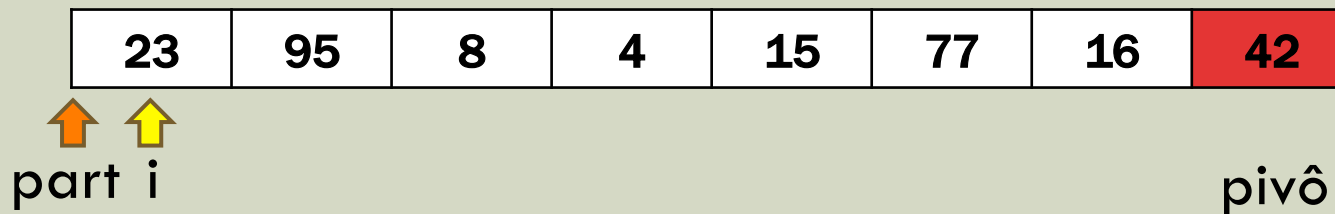
- Item $i < \text{pivô}$?
 - Se sim; avançar *part*, trocar *i* com *part*, avançar *i*.
 - Senão; avançar *i*.
- Ao final, avançar *part* e trocar *pivô* com *part*.

QUICKSORT



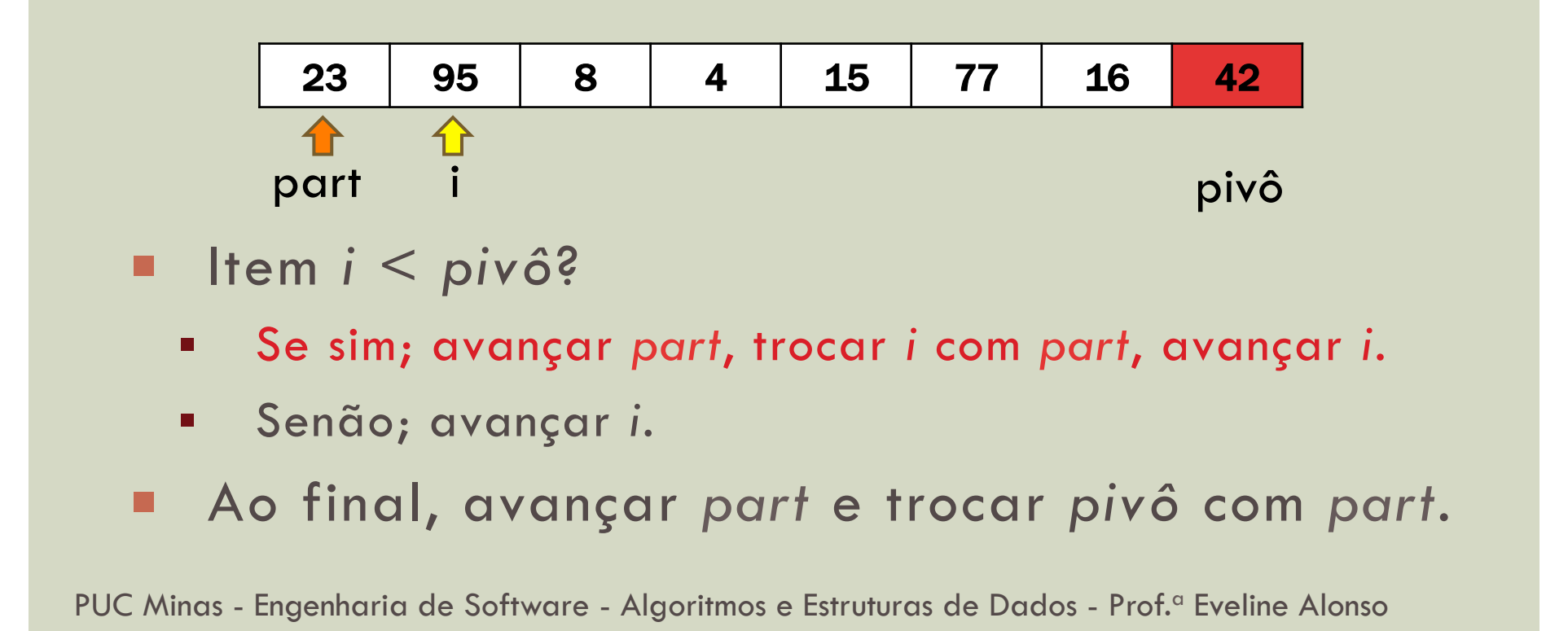
- Item $i < \text{pivô}$?
 - Se sim; avançar *part*, trocar *i* com *part*, avançar *i*.
 - Senão; avançar *i*.
- Ao final, avançar *part* e trocar *pivô* com *part*.

QUICKSORT



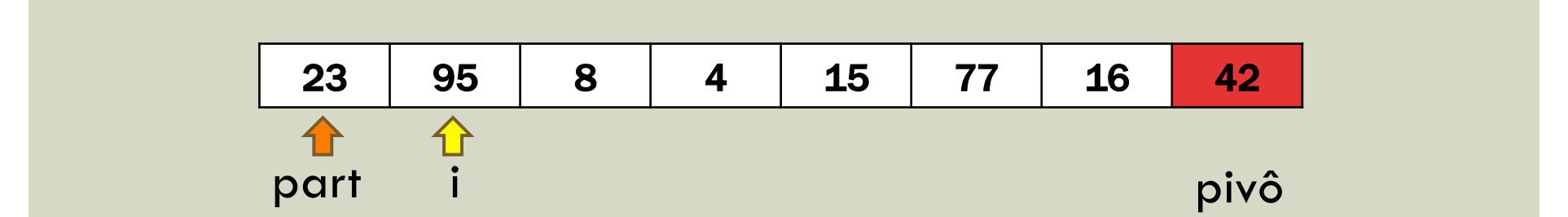
- Item $i < \text{pivô}$?
 - Se sim; avançar *part*, trocar i com *part*, avançar i .
 - Senão; avançar i .
- Ao final, avançar *part* e trocar *pivô* com *part*.

Downloaded from <http://ajph.org/> on November 10, 2014



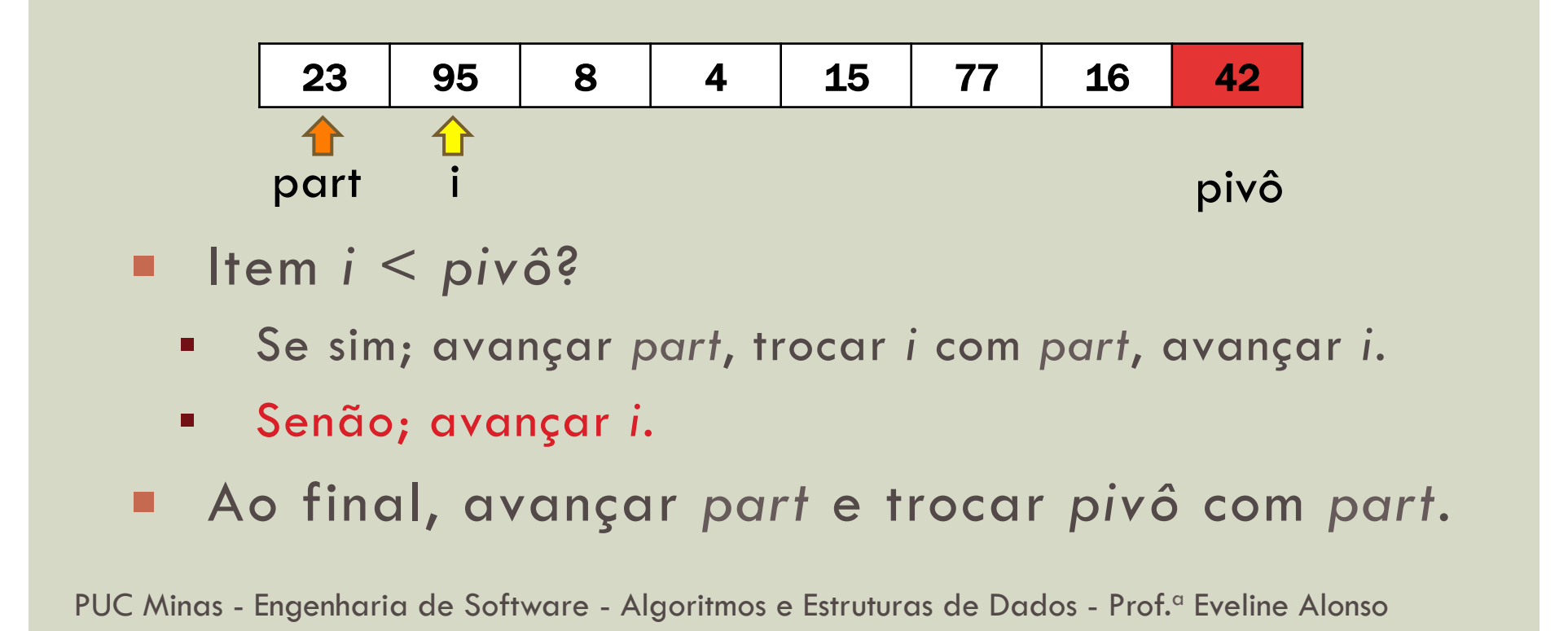
- PUC Minas - Engenharia de Software - Algoritmos e Estruturas de Dados - Prof.^a Eveline Alonso

Downloaded from <http://ajph.org/> on November 10, 2014



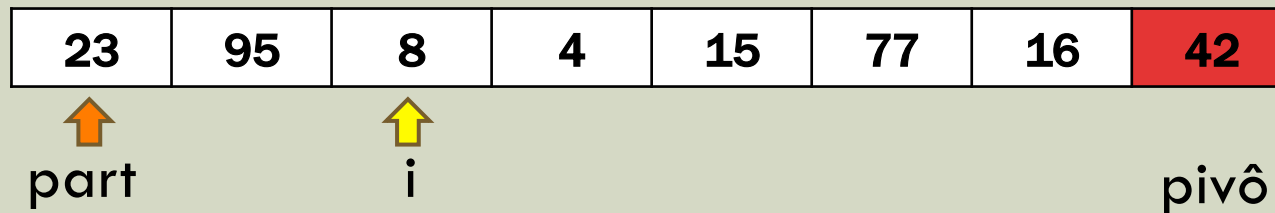
- Item $i < \text{pivô}$?
 - Se sim; avançar $part$, trocar i com $part$, avançar i .
 - Senão; avançar i .
- Ao final, avançar $part$ e trocar pivô com $part$.

Downloaded from <http://ajph.org/> on November 10, 2014



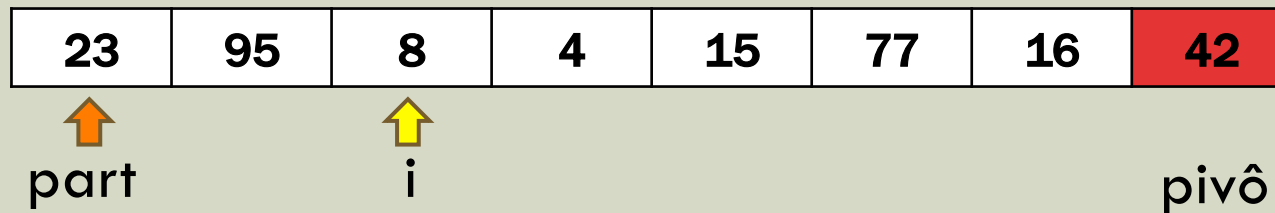
- PUC Minas - Engenharia de Software - Algoritmos e Estruturas de Dados - Prof.^a Eveline Alonso

QUICKSORT



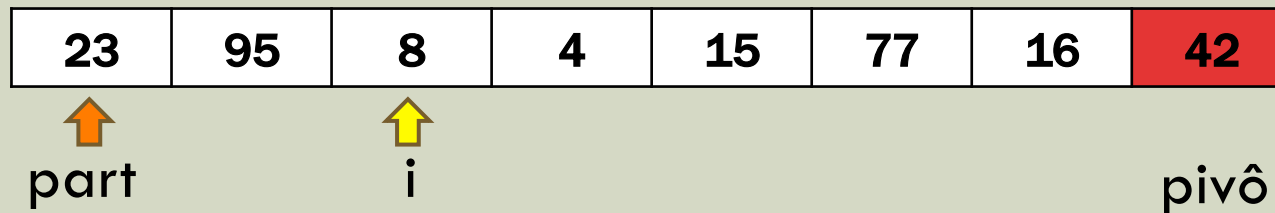
- Item $i < \text{pivô}$?
 - Se sim; avançar $part$, trocar i com $part$, avançar i .
 - **Senão; avançar i .**
- Ao final, avançar $part$ e trocar $pivô$ com $part$.

QUICKSORT



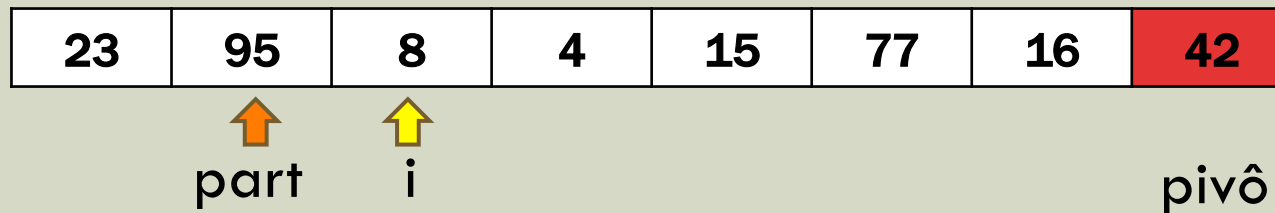
- Item $i < \text{pivô}$?
 - Se sim; avançar $part$, trocar i com $part$, avançar i .
 - Senão; avançar i .
- Ao final, avançar $part$ e trocar $pivô$ com $part$.

QUICKSORT



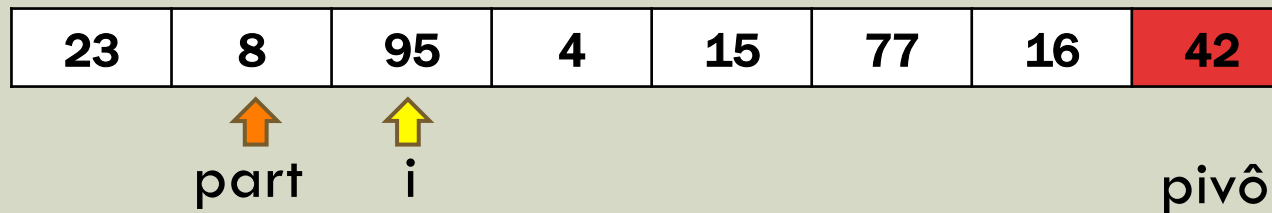
- Item $i < \text{pivô}$?
 - Se sim; avançar $part$, trocar i com $part$, avançar i .
 - Senão; avançar i .
- Ao final, avançar $part$ e trocar $pivô$ com $part$.

QUICKSORT



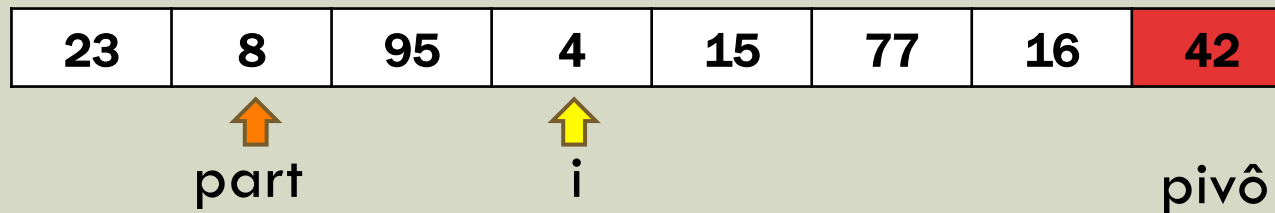
- Item $i < \text{pivô}$?
 - Se sim; avançar *part*, trocar *i* com *part*, avançar *i*.
 - Senão; avançar *i*.
- Ao final, avançar *part* e trocar *pivô* com *part*.

QUICKSORT



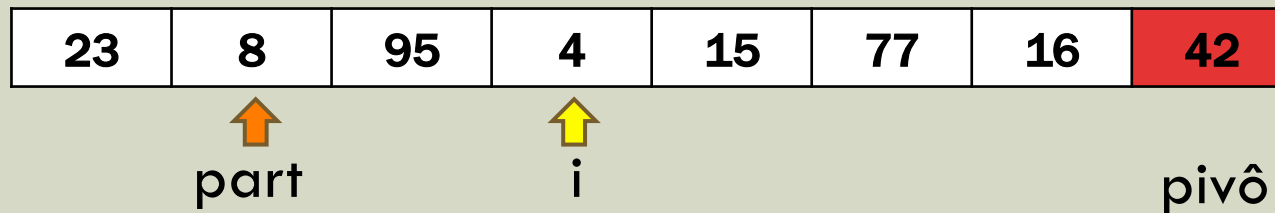
- Item $i < \text{pivô}$?
 - Se sim; avançar *part*, trocar *i* com *part*, avançar *i*.
 - Senão; avançar *i*.
- Ao final, avançar *part* e trocar *pivô* com *part*.

QUICKSORT



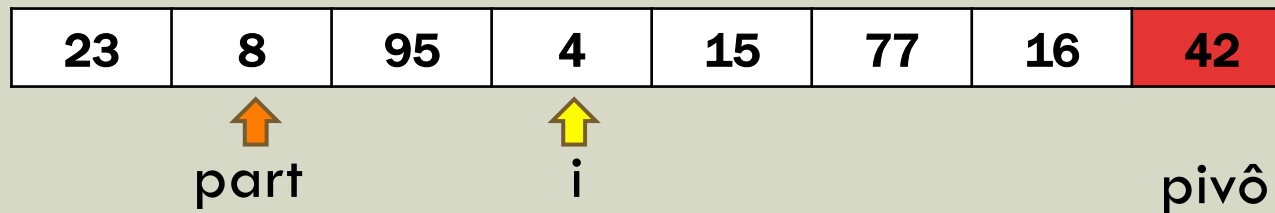
- Item $i < \text{pivô}$?
 - Se sim; avançar *part*, trocar *i* com *part*, avançar *i*.
 - Senão; avançar *i*.
- Ao final, avançar *part* e trocar *pivô* com *part*.

QUICKSORT



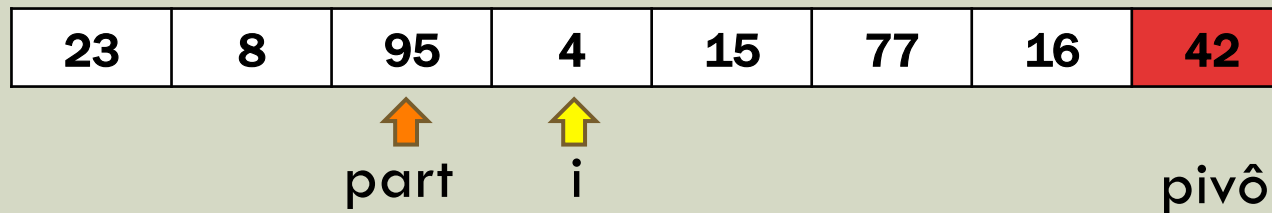
- Item $i < \text{pivô}$?
 - Se sim; avançar $part$, trocar i com $part$, avançar i .
 - Senão; avançar i .
- Ao final, avançar $part$ e trocar $pivô$ com $part$.

QUICKSORT



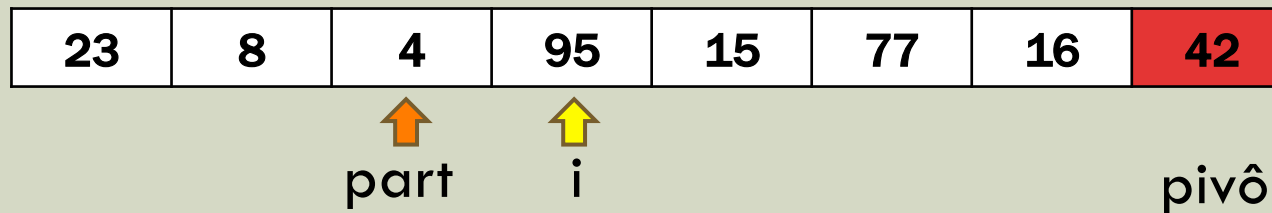
- Item $i < \text{pivô}$?
 - Se sim; avançar *part*, trocar *i* com *part*, avançar *i*.
 - Senão; avançar *i*.
- Ao final, avançar *part* e trocar *pivô* com *part*.

QUICKSORT



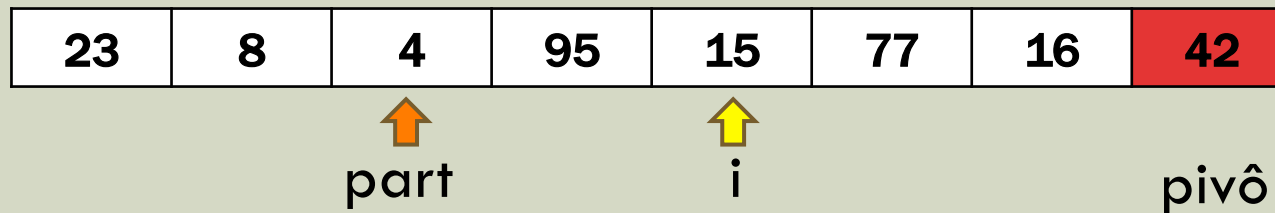
- Item $i < \text{pivô}$?
 - Se sim; avançar *part*, trocar *i* com *part*, avançar *i*.
 - Senão; avançar *i*.
- Ao final, avançar *part* e trocar *pivô* com *part*.

QUICKSORT



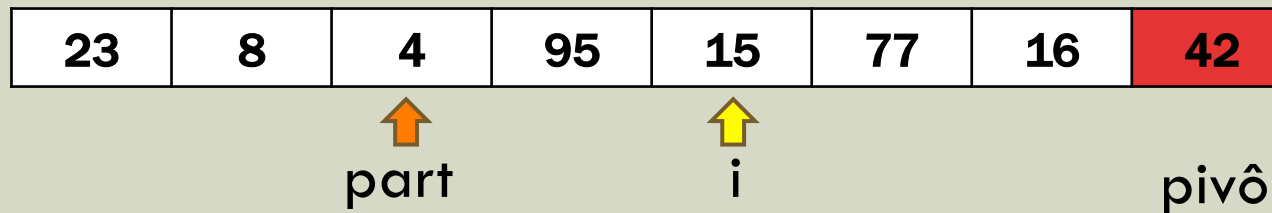
- Item $i < \text{pivô}$?
 - Se sim; avançar *part*, trocar *i* com *part*, avançar *i*.
 - Senão; avançar *i*.
- Ao final, avançar *part* e trocar *pivô* com *part*.

QUICKSORT



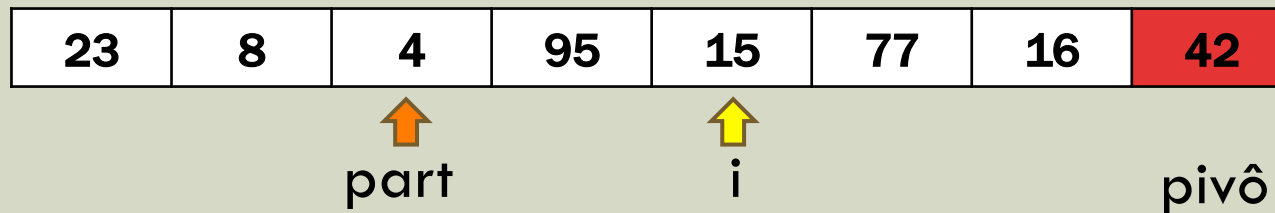
- Item $i < \text{pivô}$?
 - Se sim; avançar *part*, trocar *i* com *part*, avançar *i*.
 - Senão; avançar *i*.
- Ao final, avançar *part* e trocar *pivô* com *part*.

QUICKSORT



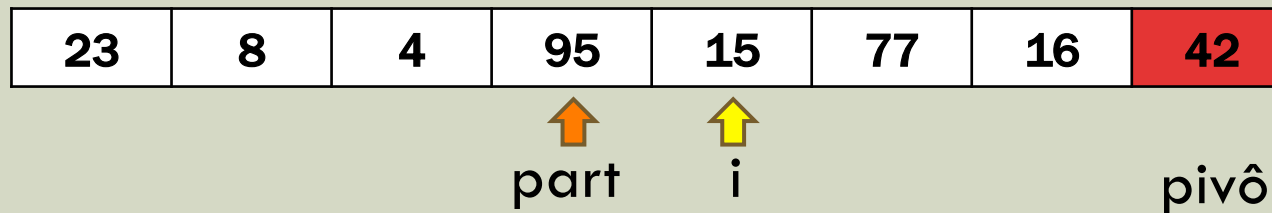
- Item $i < \text{pivô}$?
 - Se sim; avançar $part$, trocar i com $part$, avançar i .
 - Senão; avançar i .
- Ao final, avançar $part$ e trocar $pivô$ com $part$.

QUICKSORT



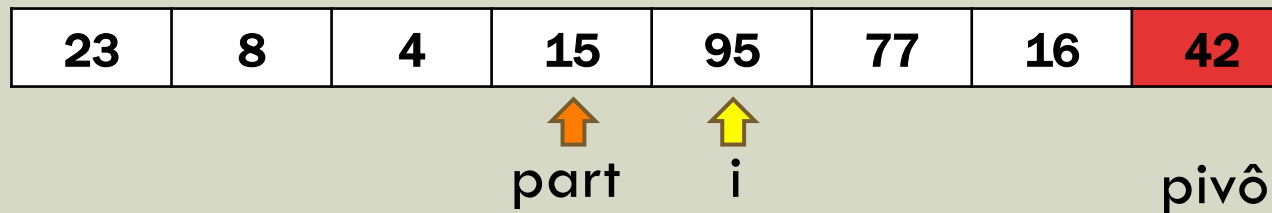
- Item $i < \text{pivô}$?
 - Se sim; avançar *part*, trocar *i* com *part*, avançar *i*.
 - Senão; avançar *i*.
- Ao final, avançar *part* e trocar *pivô* com *part*.

QUICKSORT



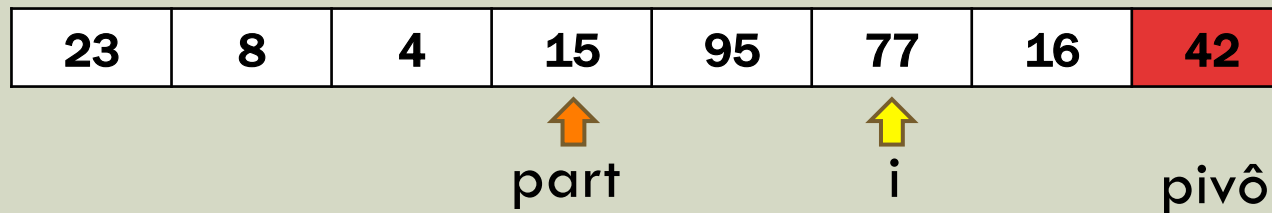
- Item $i < \text{pivô}$?
 - Se sim; avançar *part*, trocar *i* com *part*, avançar *i*.
 - Senão; avançar *i*.
- Ao final, avançar *part* e trocar *pivô* com *part*.

QUICKSORT



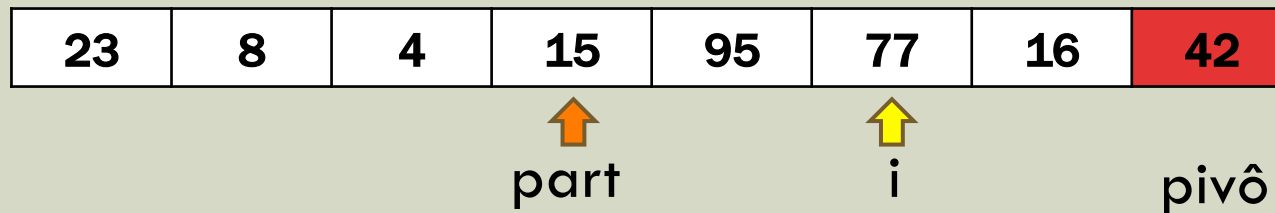
- Item $i < \text{pivô}$?
 - Se sim; avançar *part*, trocar *i* com *part*, avançar *i*.
 - Senão; avançar *i*.
- Ao final, avançar *part* e trocar *pivô* com *part*.

QUICKSORT



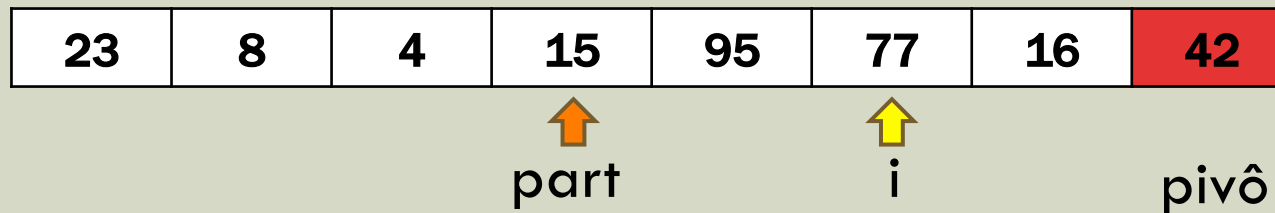
- Item $i < \text{pivô}$?
 - Se sim; avançar *part*, trocar *i* com *part*, avançar *i*.
 - Senão; avançar *i*.
- Ao final, avançar *part* e trocar *pivô* com *part*.

QUICKSORT



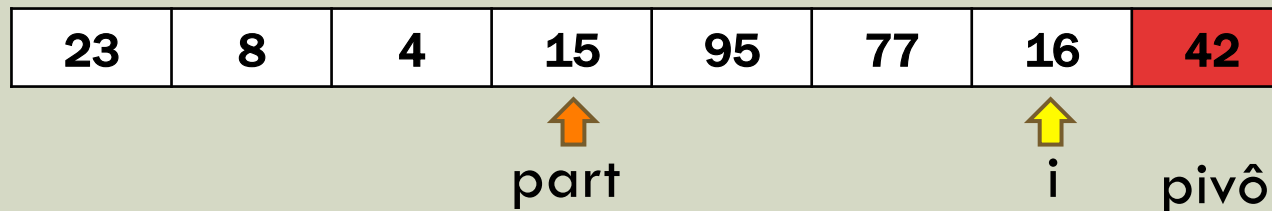
- Item $i < \text{pivô}$?
 - Se sim; avançar $part$, trocar i com $part$, avançar i .
 - Senão; avançar i .
- Ao final, avançar $part$ e trocar pivô com $part$.

QUICKSORT



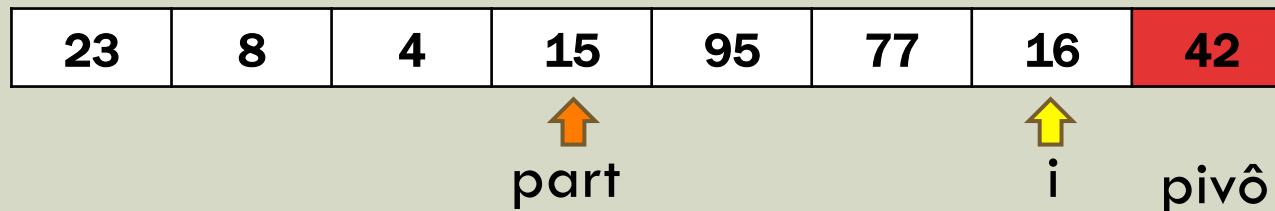
- Item $i < \text{pivô}$?
 - Se sim; avançar $part$, trocar i com $part$, avançar i .
 - **Senão; avançar i .**
- Ao final, avançar $part$ e trocar pivô com $part$.

QUICKSORT



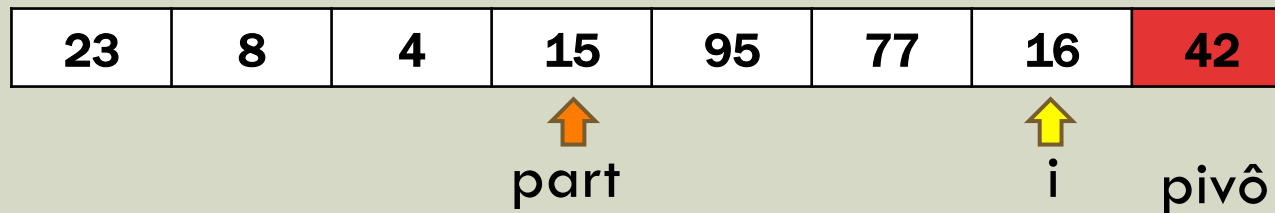
- Item $i < \text{pivô}$?
 - Se sim; avançar $part$, trocar i com $part$, avançar i .
 - **Senão; avançar i .**
- Ao final, avançar $part$ e trocar $pivô$ com $part$.

QUICKSORT



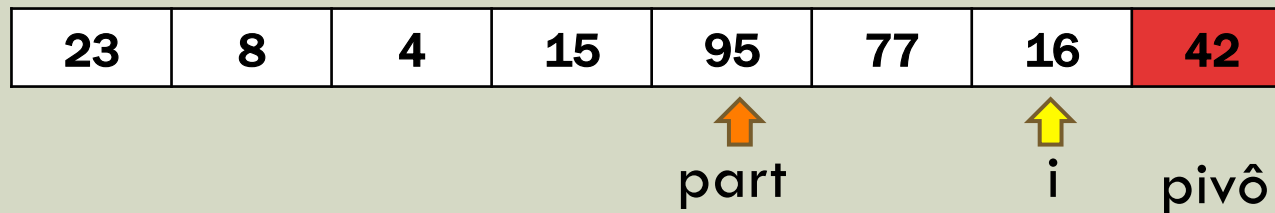
- Item $i < \text{pivô}$?
 - Se sim; avançar $part$, trocar i com $part$, avançar i .
 - Senão; avançar i .
- Ao final, avançar $part$ e trocar $pivô$ com $part$.

QUICKSORT



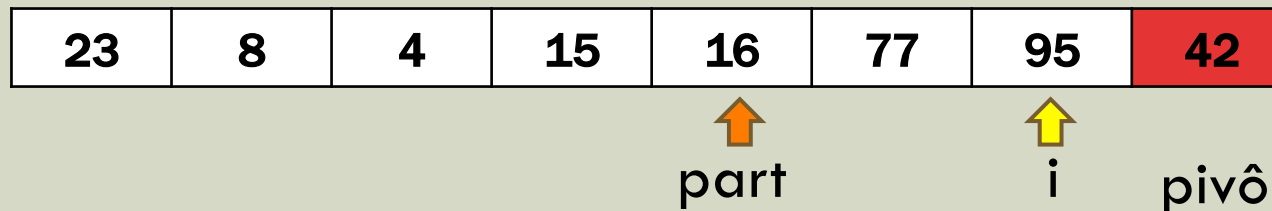
- Item $i < \text{pivô}$?
 - Se sim; avançar *part*, trocar *i* com *part*, avançar *i*.
 - Senão; avançar *i*.
- Ao final, avançar *part* e trocar *pivô* com *part*.

QUICKSORT



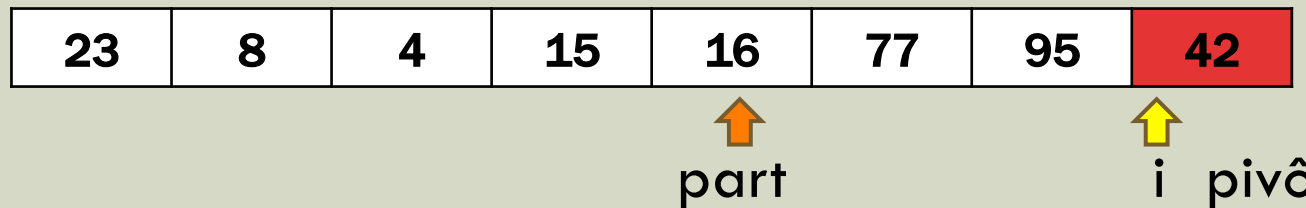
- Item $i < \text{pivô}$?
 - Se sim; avançar *part*, trocar *i* com *part*, avançar *i*.
 - Senão; avançar *i*.
- Ao final, avançar *part* e trocar *pivô* com *part*.

QUICKSORT



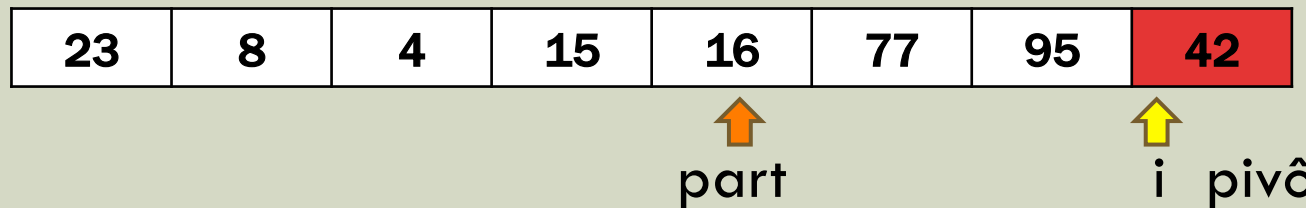
- Item $i < \text{pivô}$?
 - Se sim; avançar *part*, trocar *i* com *part*, avançar *i*.
 - Senão; avançar *i*.
- Ao final, avançar *part* e trocar *pivô* com *part*.

QUICKSORT



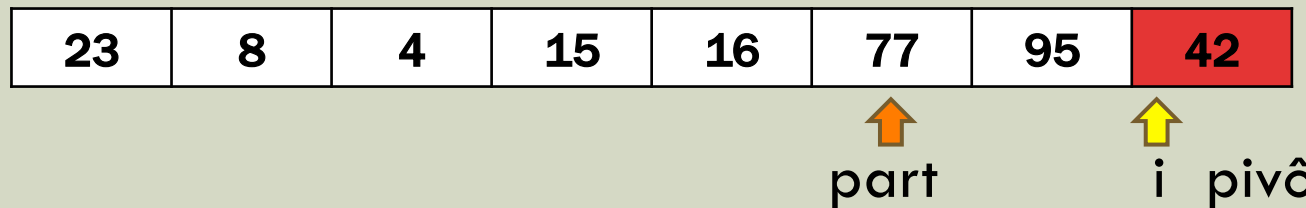
- Item $i < \text{pivô}$?
 - Se sim; avançar *part*, trocar *i* com *part*, avançar *i*.
 - Senão; avançar *i*.
- Ao final, avançar *part* e trocar *pivô* com *part*.

QUICKSORT



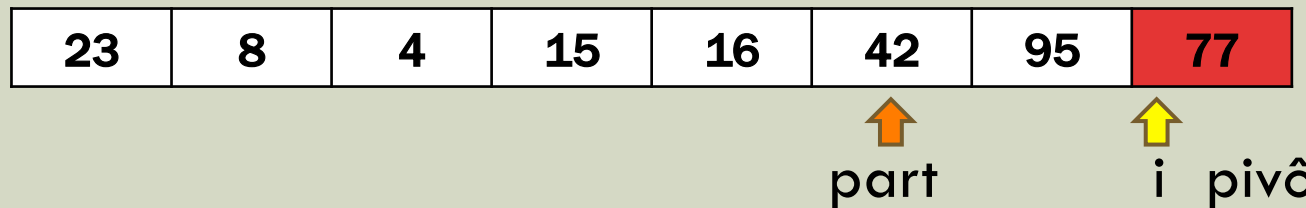
- Item $i < \text{pivô}$?
 - Se sim; avançar $part$, trocar i com $part$, avançar i .
 - Senão; avançar i .
- Ao final, avançar $part$ e trocar $pivô$ com $part$.

QUICKSORT



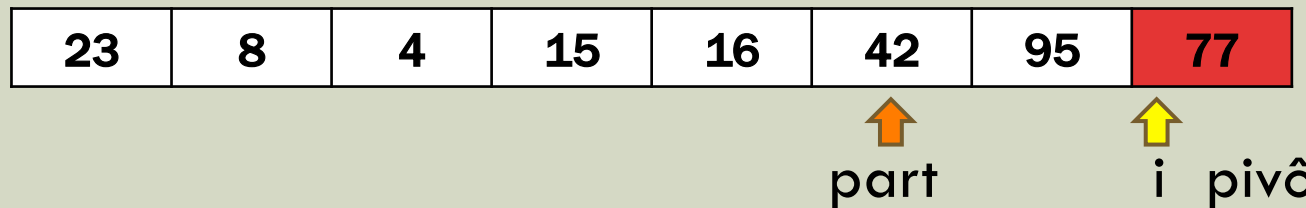
- Item $i < \text{pivô}$?
 - Se sim; avançar $part$, trocar i com $part$, avançar i .
 - Senão; avançar i .
- Ao final, avançar $part$ e trocar $pivô$ com $part$.

QUICKSORT



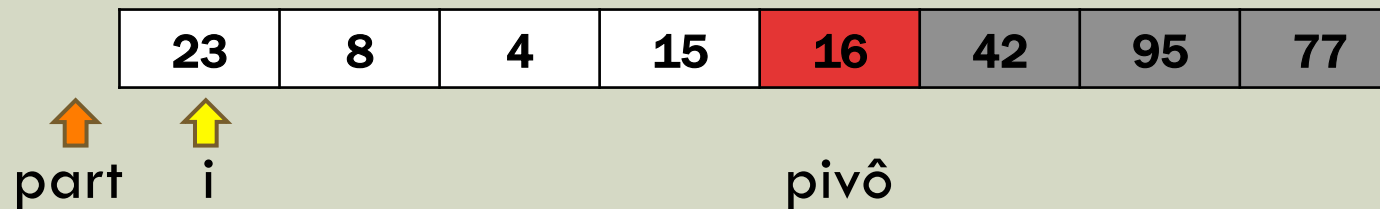
- Item $i < \text{pivô}$?
 - Se sim; avançar $part$, trocar i com $part$, avançar i .
 - Senão; avançar i .
- Ao final, avançar $part$ e trocar $pivô$ com $part$.

QUICKSORT



- Ordenação parcial:
 - maiores à direita;
 - e menores à esquerda da partição.
- Recursivamente, ordenar partições da direita e da esquerda.

QUICKSORT



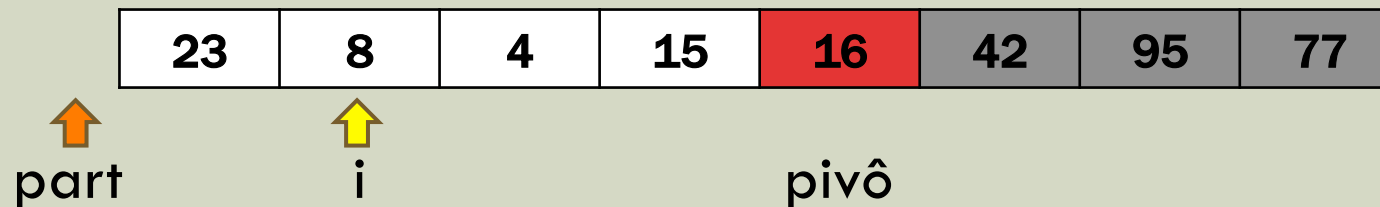
- Item $i < \text{pivô}$?
 - Se sim; avançar $part$, trocar i com $part$, avançar i .
 - Senão; avançar i .
- Ao final, avançar $part$ e trocar pivô com $part$.

Downloaded from <http://ajph.org/> on November 10, 2015



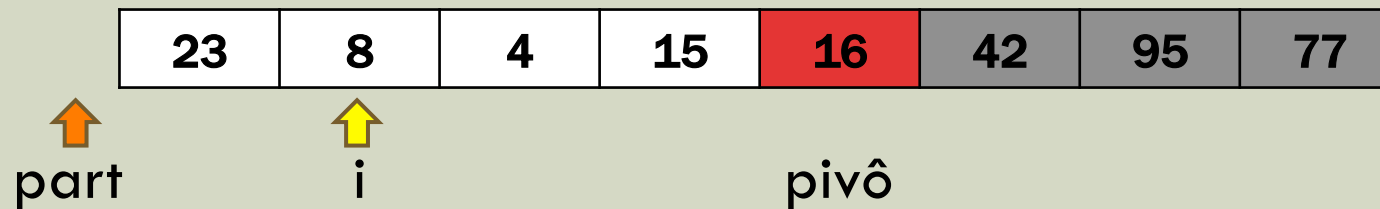
- Se sim; avançar *part*, trocar *i* com *part*, avançar *i*.
- Senão; avançar *i*.

QUICKSORT



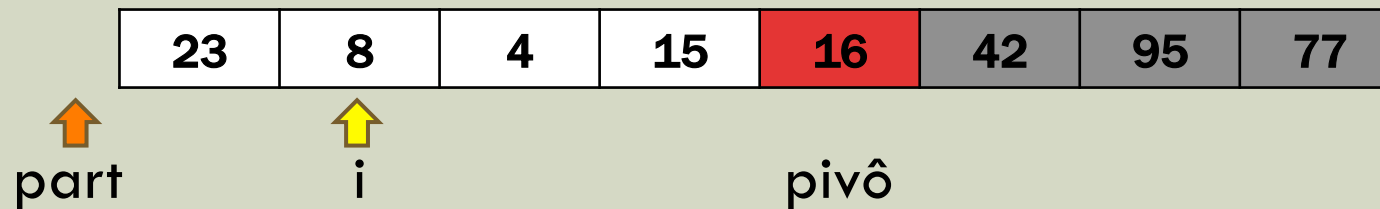
- Item $i < pivô$?
 - Se sim; avançar $part$, trocar i com $part$, avançar i .
 - **Senão; avançar i .**
- Ao final, avançar $part$ e trocar $pivô$ com $part$.

QUICKSORT



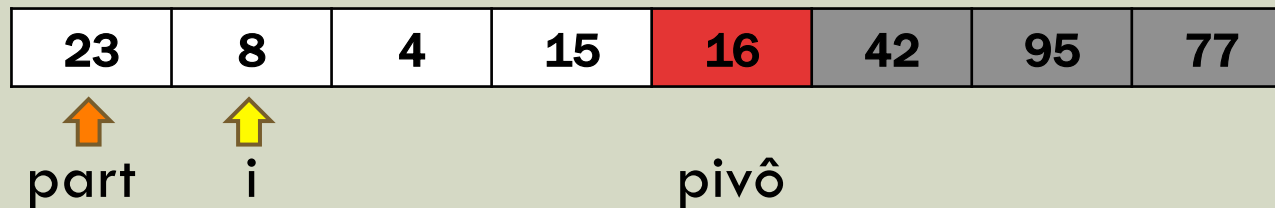
- **Item $i < \text{pivô}$?**
 - Se sim; avançar *part*, trocar *i* com *part*, avançar *i*.
 - Senão; avançar *i*.
- Ao final, avançar *part* e trocar *pivô* com *part*.

QUICKSORT



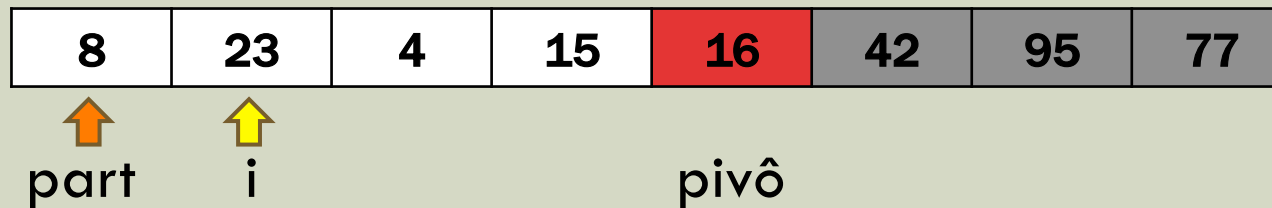
- Item $i < \text{pivô}$?
 - Se sim; avançar *part*, trocar *i* com *part*, avançar *i*.
 - Senão; avançar *i*.
- Ao final, avançar *part* e trocar *pivô* com *part*.

QUICKSORT



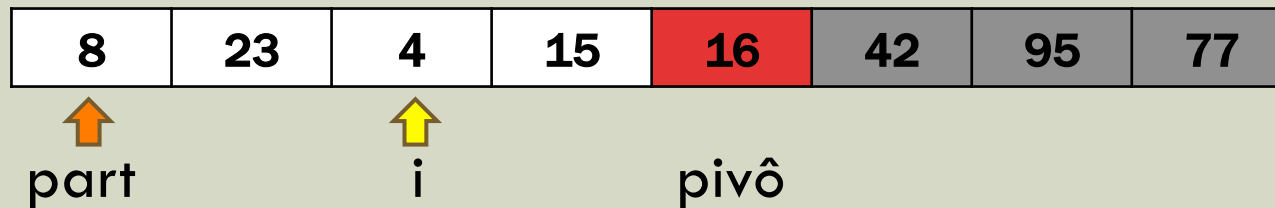
- Item $i < \text{pivô}$?
 - Se sim; avançar *part*, trocar *i* com *part*, avançar *i*.
 - Senão; avançar *i*.
- Ao final, avançar *part* e trocar *pivô* com *part*.

QUICKSORT



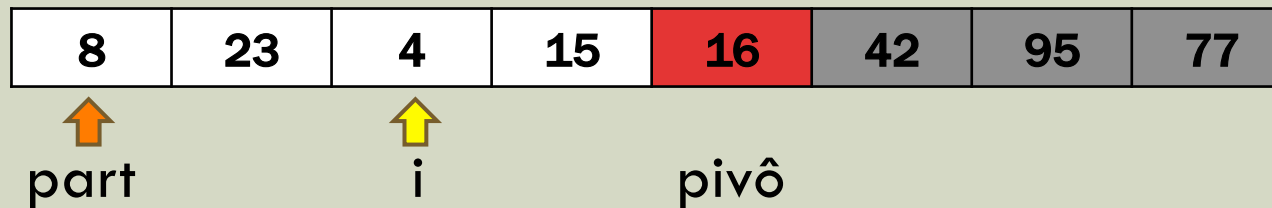
- Item $i < \text{pivô}$?
 - Se sim; avançar $part$, trocar i com $part$, avançar i .
 - Senão; avançar i .
- Ao final, avançar $part$ e trocar $pivô$ com $part$.

QUICKSORT



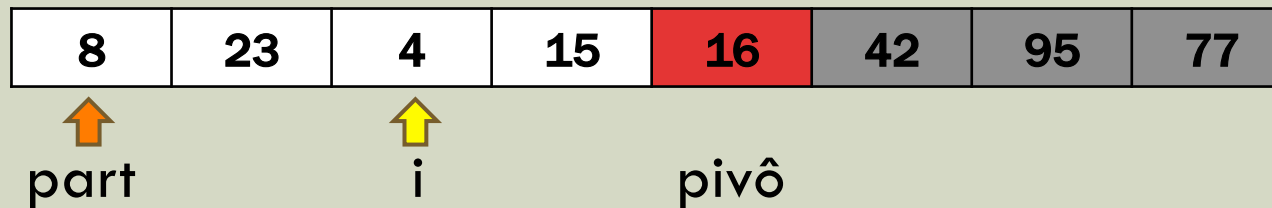
- Item $i < \text{pivô}$?
 - Se sim; avançar *part*, trocar *i* com *part*, avançar *i*.
 - Senão; avançar *i*.
- Ao final, avançar *part* e trocar *pivô* com *part*.

QUICKSORT



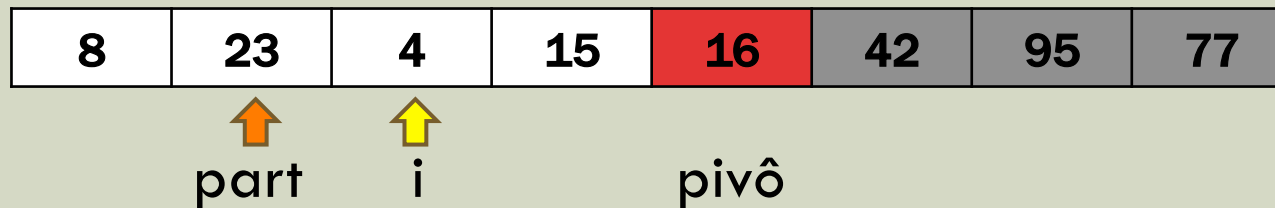
- Item $i < \text{pivô}$?
 - Se sim; avançar $part$, trocar i com $part$, avançar i .
 - Senão; avançar i .
- Ao final, avançar $part$ e trocar $pivô$ com $part$.

QUICKSORT



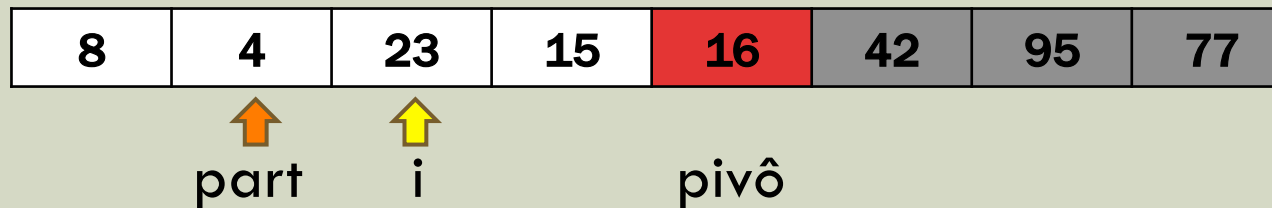
- Item $i < \text{pivô}$?
 - Se sim; avançar *part*, trocar *i* com *part*, avançar *i*.
 - Senão; avançar *i*.
- Ao final, avançar *part* e trocar *pivô* com *part*.

QUICKSORT



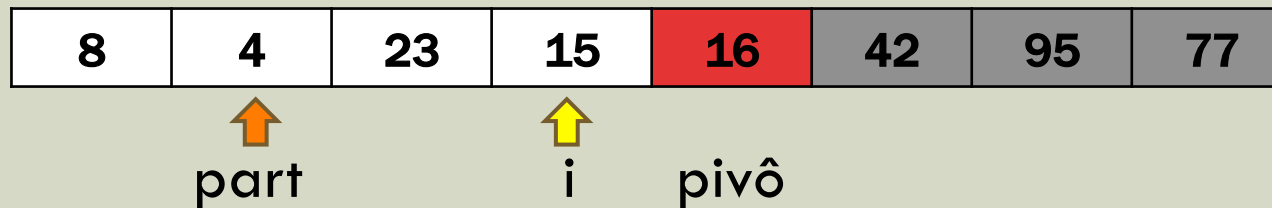
- Item $i < \text{pivô}$?
 - Se sim; avançar *part*, trocar *i* com *part*, avançar *i*.
 - Senão; avançar *i*.
- Ao final, avançar *part* e trocar *pivô* com *part*.

QUICKSORT



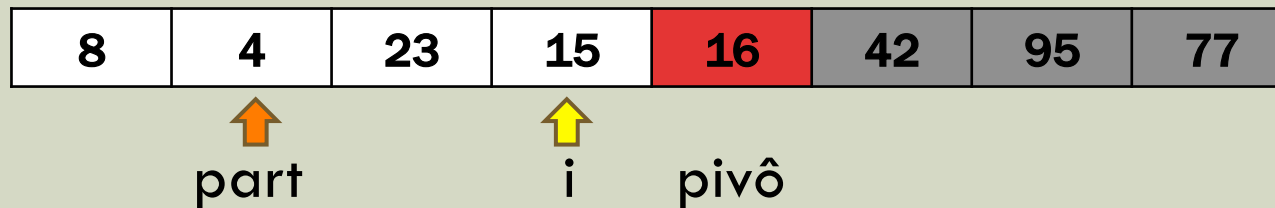
- Item $i < \text{pivô}$?
 - Se sim; avançar *part*, trocar *i* com *part*, avançar *i*.
 - Senão; avançar *i*.
- Ao final, avançar *part* e trocar *pivô* com *part*.

QUICKSORT



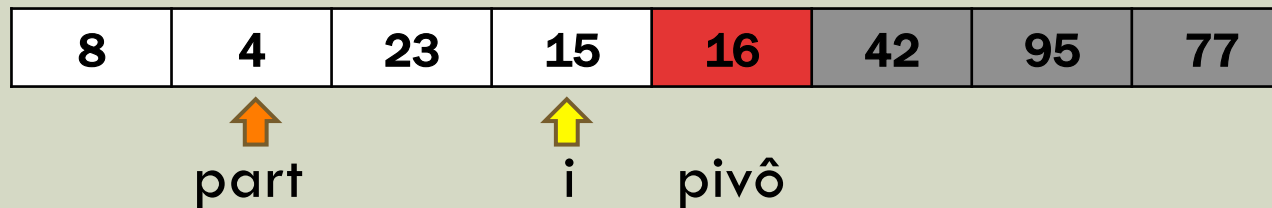
- Item $i < \text{pivô}$?
 - Se sim; avançar *part*, trocar *i* com *part*, avançar *i*.
 - Senão; avançar *i*.
- Ao final, avançar *part* e trocar *pivô* com *part*.

QUICKSORT



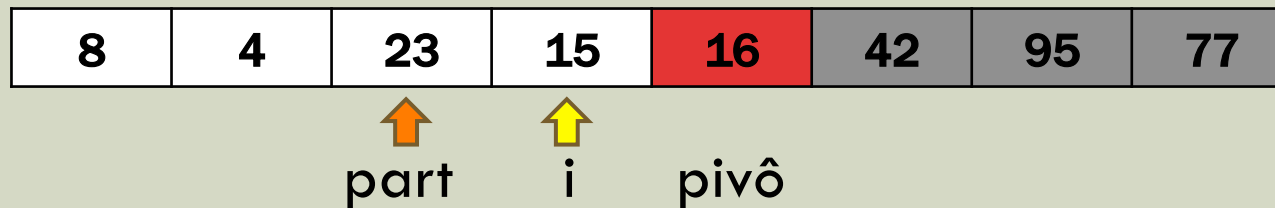
- Item $i < \text{pivô}$?
 - Se sim; avançar $part$, trocar i com $part$, avançar i .
 - Senão; avançar i .
- Ao final, avançar $part$ e trocar $pivô$ com $part$.

QUICKSORT



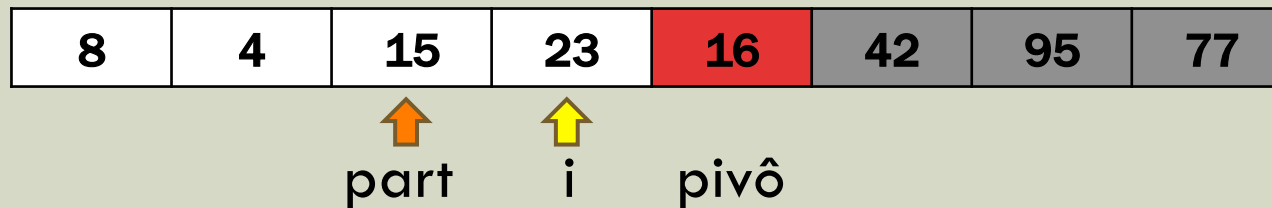
- Item $i < \text{pivô}$?
 - Se sim; avançar *part*, trocar *i* com *part*, avançar *i*.
 - Senão; avançar *i*.
- Ao final, avançar *part* e trocar *pivô* com *part*.

QUICKSORT



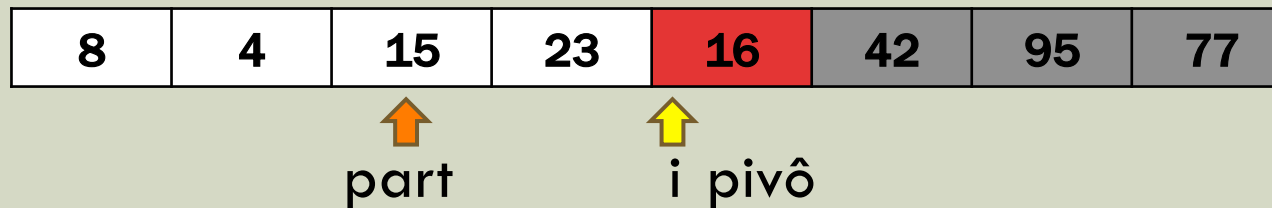
- Item $i < \text{pivô}$?
 - Se sim; avançar *part*, trocar *i* com *part*, avançar *i*.
 - Senão; avançar *i*.
- Ao final, avançar *part* e trocar *pivô* com *part*.

QUICKSORT



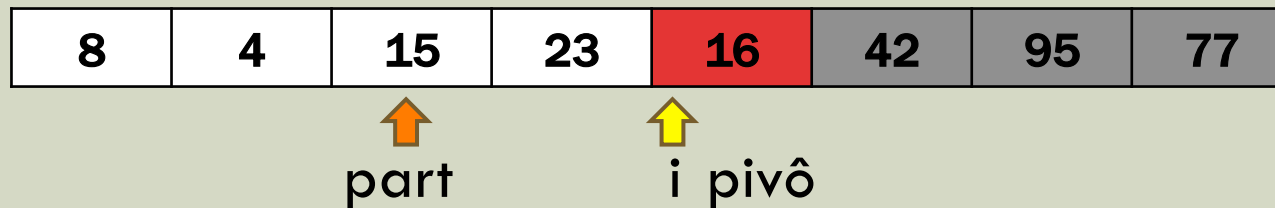
- Item $i < \text{pivô}$?
 - Se sim; avançar *part*, trocar *i* com *part*, avançar *i*.
 - Senão; avançar *i*.
- Ao final, avançar *part* e trocar *pivô* com *part*.

QUICKSORT



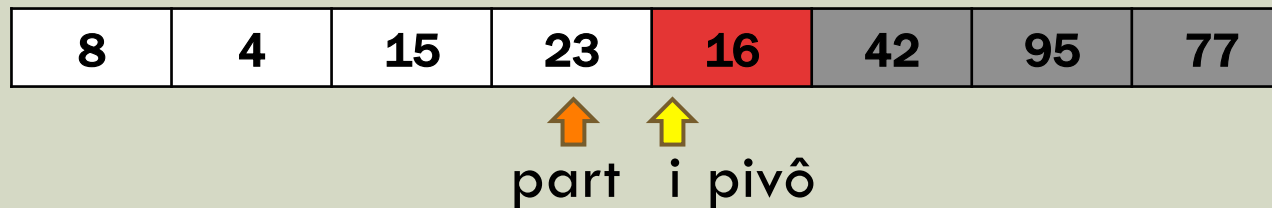
- Item $i < \text{pivô}$?
 - Se sim; avançar *part*, trocar i com *part*, avançar i .
 - Senão; avançar i .
- Ao final, avançar *part* e trocar *pivô* com *part*.

QUICKSORT



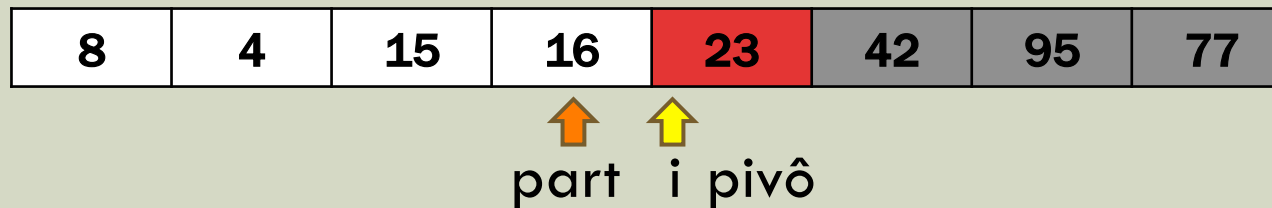
- Item $i < \text{pivô}$?
 - Se sim; avançar $part$, trocar i com $part$, avançar i .
 - Senão; avançar i .
- Ao final, avançar $part$ e trocar $pivô$ com $part$.

QUICKSORT



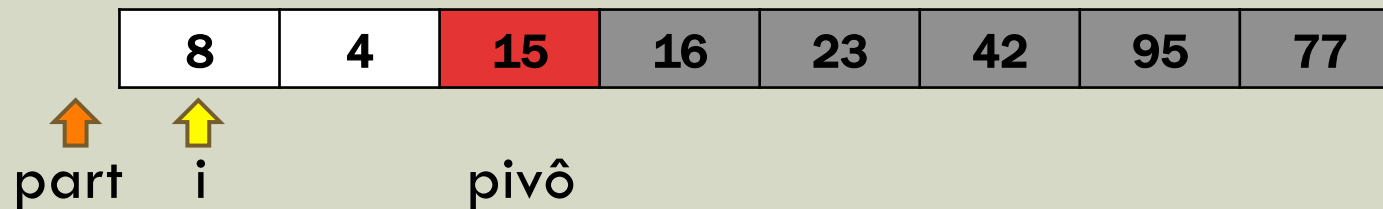
- Item $i < \text{pivô}$?
 - Se sim; avançar $part$, trocar i com $part$, avançar i .
 - Senão; avançar i .
- Ao final, avançar $part$ e trocar $pivô$ com $part$.

QUICKSORT



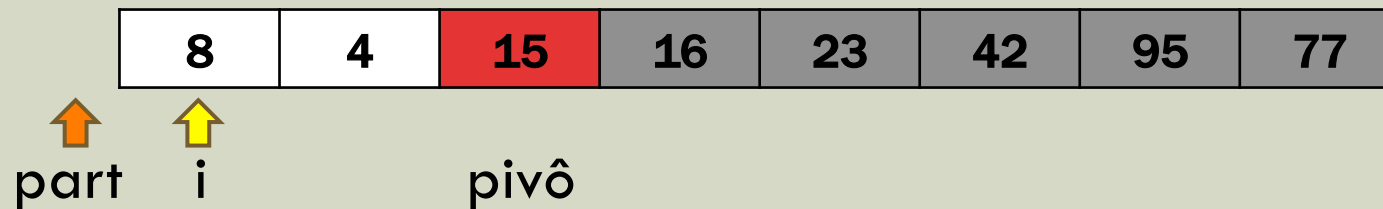
- Item $i < \text{pivô}$?
 - Se sim; avançar $part$, trocar i com $part$, avançar i .
 - Senão; avançar i .
- Ao final, avançar $part$ e trocar $pivô$ com $part$.

QUICKSORT



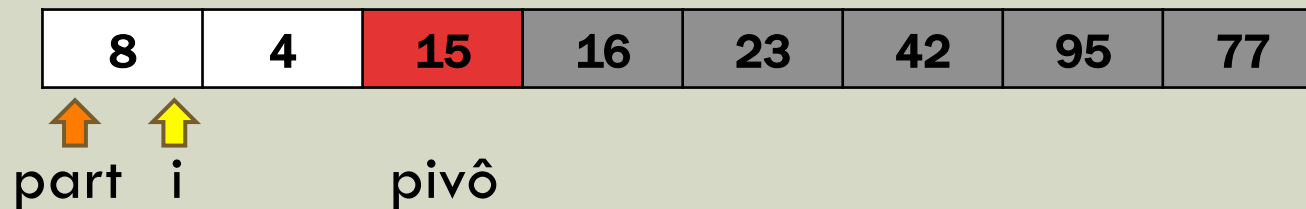
- **Item $i < \text{pivô}$?**
 - Se sim; avançar $part$, trocar i com $part$, avançar i .
 - Senão; avançar i .
- Ao final, avançar $part$ e trocar $pivô$ com $part$.

QUICKSORT



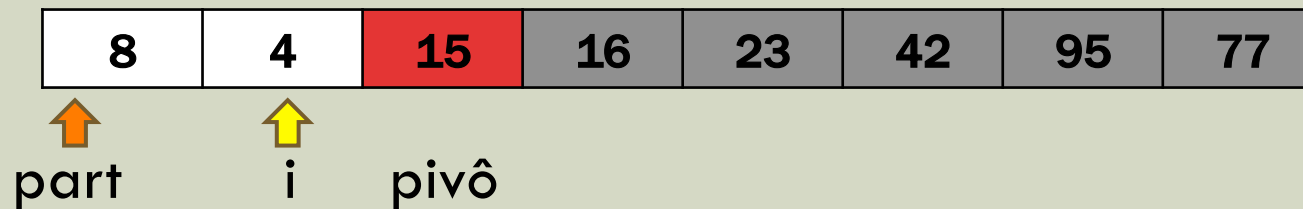
- Item $i < \text{pivô}$?
 - Se sim; avançar *part*, trocar *i* com *part*, avançar *i*.
 - Senão; avançar *i*.
- Ao final, avançar *part* e trocar *pivô* com *part*.

QUICKSORT



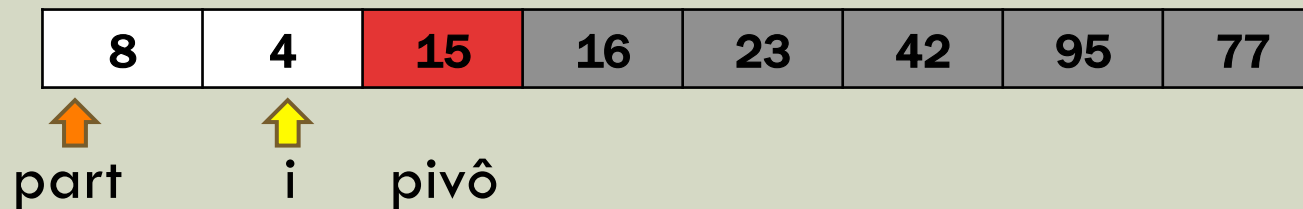
- Item $i < \text{pivô}$?
 - Se sim; avançar *part*, trocar i com *part*, avançar i .
 - Senão; avançar i .
- Ao final, avançar *part* e trocar *pivô* com *part*.

QUICKSORT



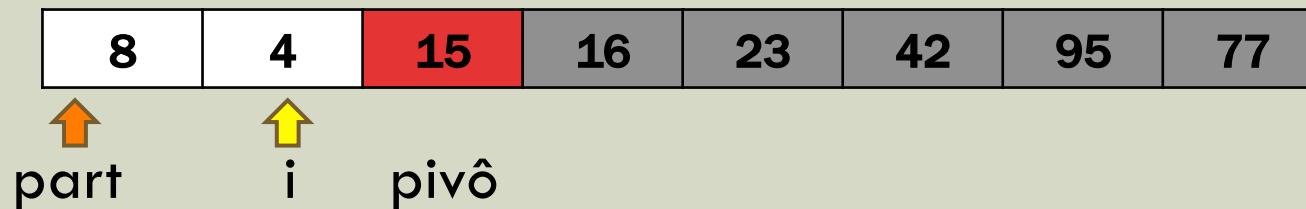
- Item $i < \text{pivô}$?
 - Se sim; avançar *part*, trocar *i* com *part*, avançar *i*.
 - Senão; avançar *i*.
- Ao final, avançar *part* e trocar *pivô* com *part*.

QUICKSORT



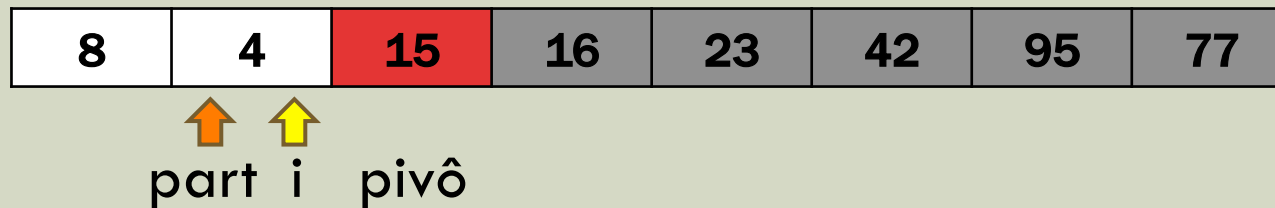
- Item $i < \text{pivô}$?
 - Se sim; avançar $part$, trocar i com $part$, avançar i .
 - Senão; avançar i .
- Ao final, avançar $part$ e trocar $pivô$ com $part$.

QUICKSORT



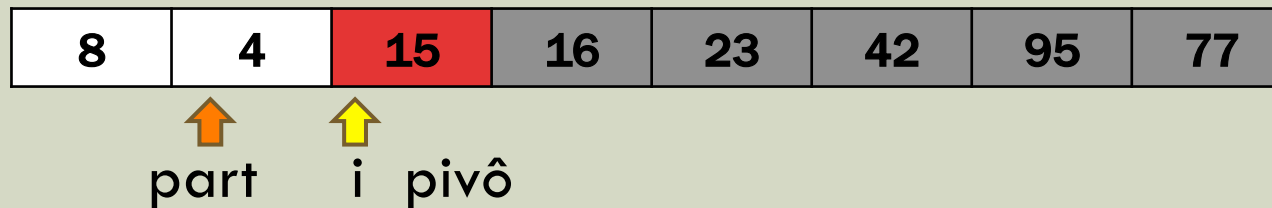
- Item $i < \text{pivô}$?
 - Se sim; avançar *part*, trocar *i* com *part*, avançar *i*.
 - Senão; avançar *i*.
- Ao final, avançar *part* e trocar *pivô* com *part*.

QUICKSORT



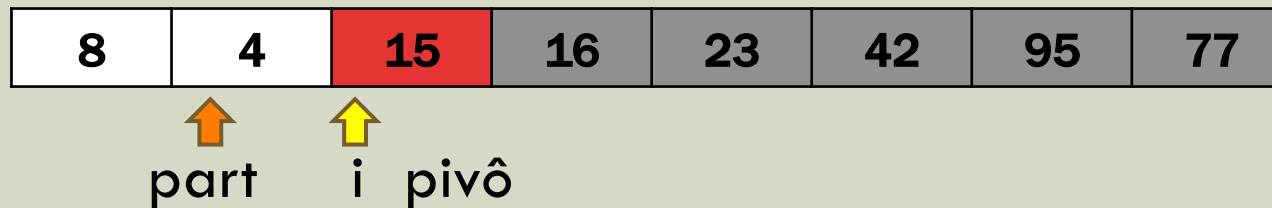
- Item $i < pivô$?
 - Se sim; avançar $part$, trocar i com $part$, avançar i .
 - Senão; avançar i .
- Ao final, avançar $part$ e trocar $pivô$ com $part$.

QUICKSORT



- Item $i < \text{pivô}$?
 - Se sim; avançar *part*, trocar i com *part*, avançar i .
 - Senão; avançar i .
- Ao final, avançar *part* e trocar *pivô* com *part*.

QUICKSORT



- Item $i < \text{pivô}$?
 - Se sim; avançar $part$, trocar i com $part$, avançar i .
 - Senão; avançar i .
- Ao final, avançar $part$ e trocar $pivô$ com $part$.

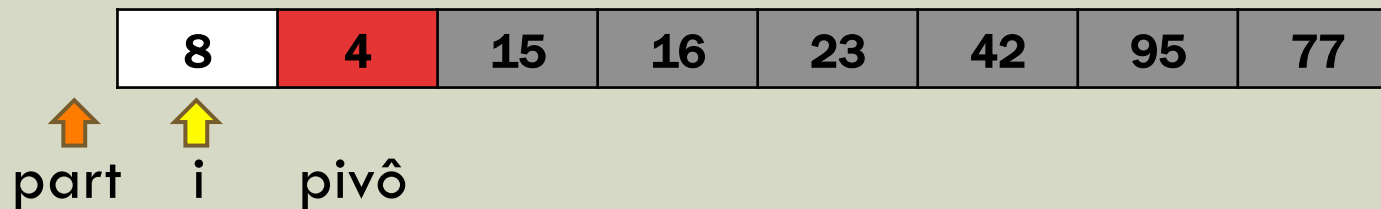
QUICKSORT

8	4	15	16	23	42	95	77
---	---	----	----	----	----	----	----

 
part i pivô

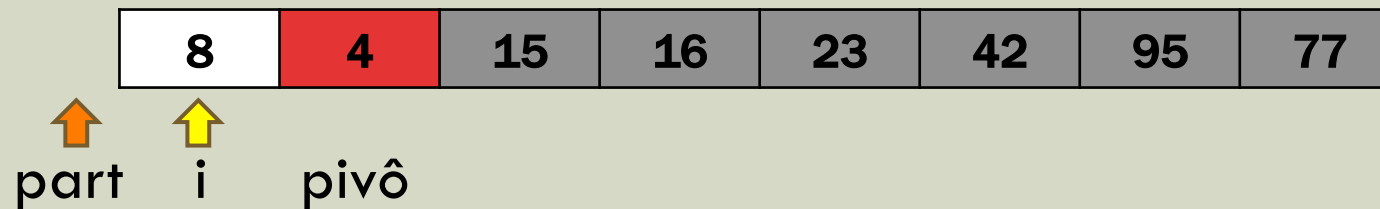
- Item $i < \text{pivô}$?
 - Se sim; avançar $part$, trocar i com $part$, avançar i .
 - Senão; avançar i .
- Ao final, avançar $part$ e trocar $pivô$ com $part$.

QUICKSORT



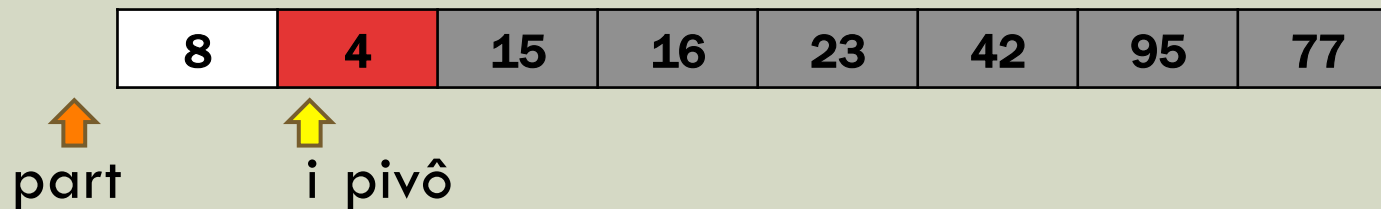
- Item $i < \text{pivô}$?
 - Se sim; avançar $part$, trocar i com $part$, avançar i .
 - Senão; avançar i .
- Ao final, avançar $part$ e trocar $pivô$ com $part$.

QUICKSORT



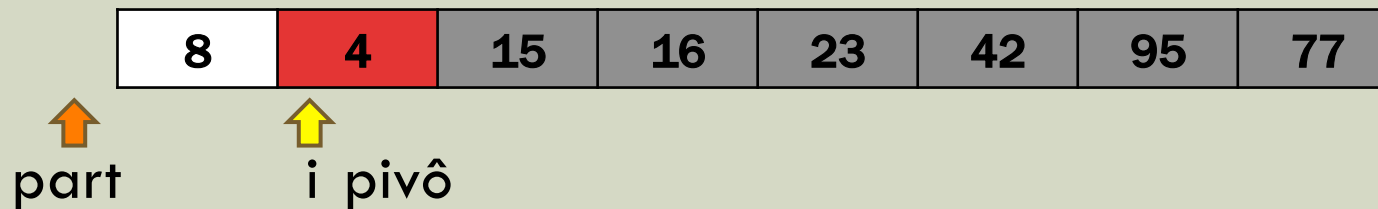
- Item $i < \text{pivô}$?
 - Se sim; avançar *part*, trocar *i* com *part*, avançar *i*.
 - **Senão; avançar *i*.**
- Ao final, avançar *part* e trocar *pivô* com *part*.

QUICKSORT



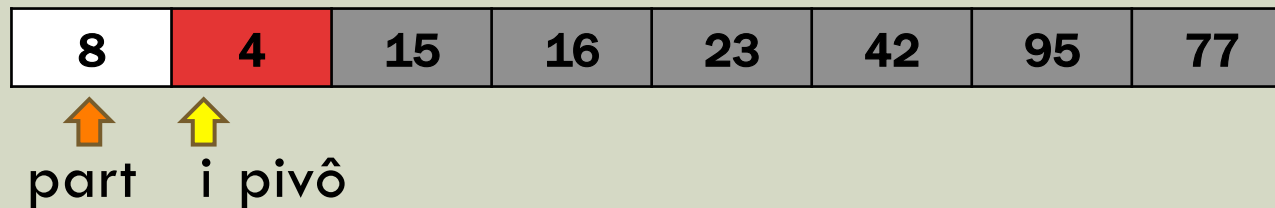
- Item $i < \text{pivô}$?
 - Se sim; avançar *part*, trocar *i* com *part*, avançar *i*.
 - **Senão; avançar *i*.**
- Ao final, avançar *part* e trocar *pivô* com *part*.

QUICKSORT



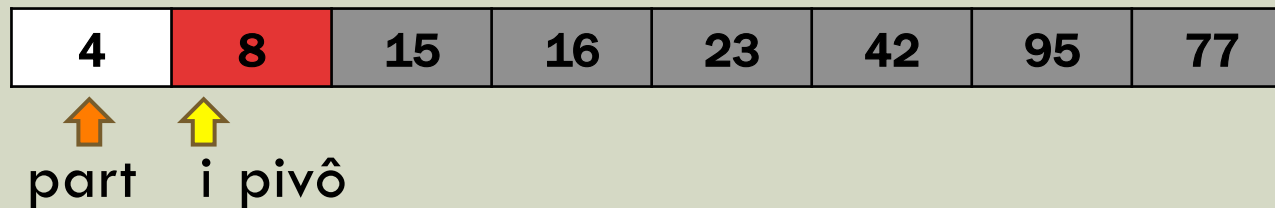
- Item $i < \text{pivô}$?
 - Se sim; avançar $part$, trocar i com $part$, avançar i .
 - Senão; avançar i .
- Ao final, avançar $part$ e trocar $pivô$ com $part$.

QUICKSORT



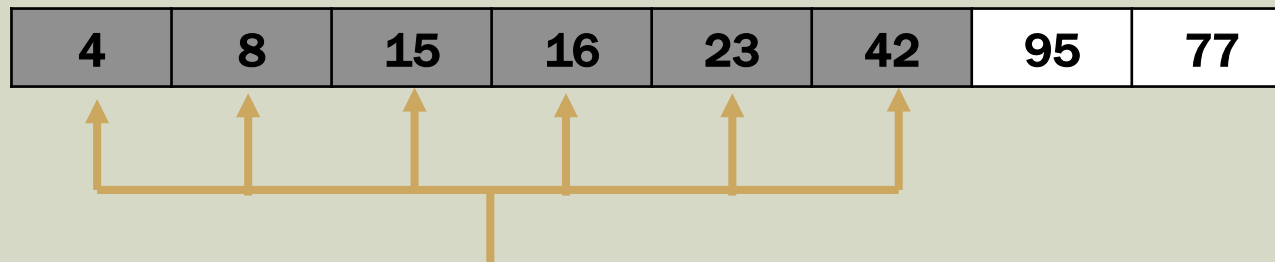
- Item $i < \text{pivô}$?
 - Se sim; avançar $part$, trocar i com $part$, avançar i .
 - Senão; avançar i .
- Ao final, avançar $part$ e trocar $pivô$ com $part$.

QUICKSORT



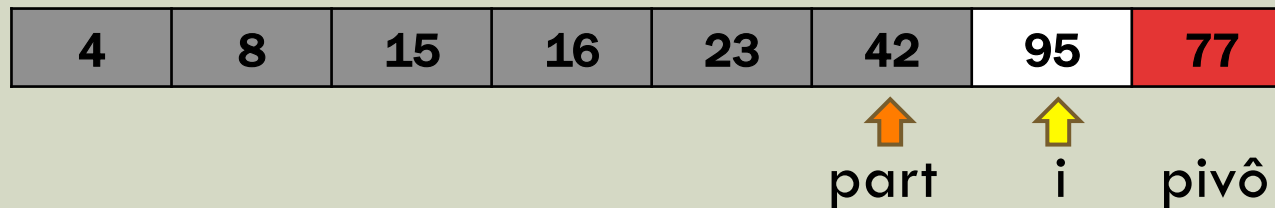
- Item $i < \text{pivô}$?
 - Se sim; avançar $part$, trocar i com $part$, avançar i .
 - Senão; avançar i .
- Ao final, avançar $part$ e trocar $pivô$ com $part$.

QUICKSORT



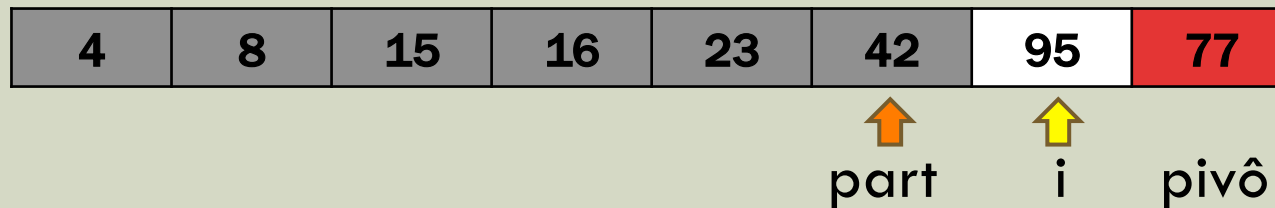
- Partições de tamanho 1:
 - já ordenadas!

QUICKSORT



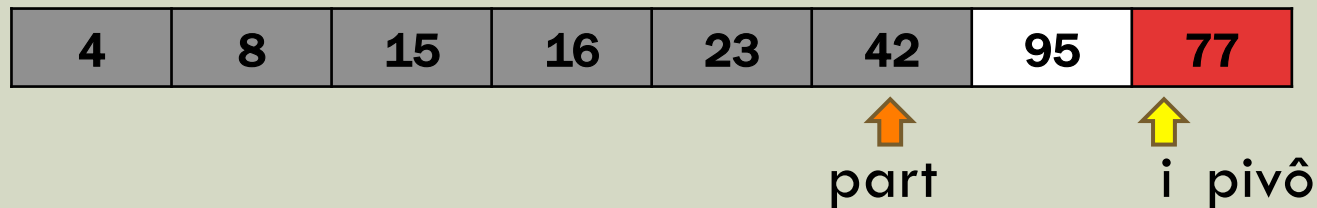
- **Item $i < \text{pivô}$?**
 - Se sim; avançar $part$, trocar i com $part$, avançar i .
 - Senão; avançar i .
- Ao final, avançar $part$ e trocar $pivô$ com $part$.

QUICKSORT



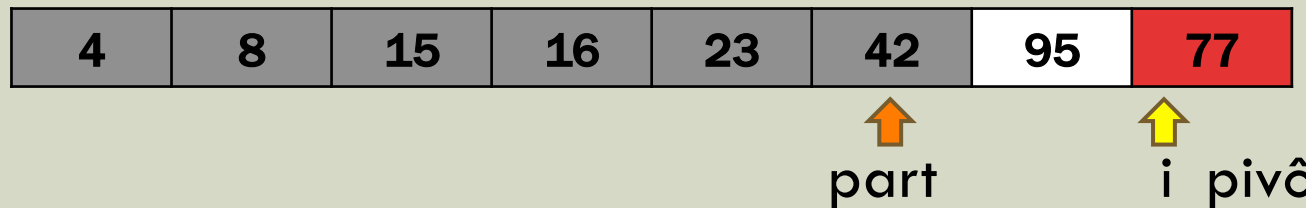
- Item $i < \text{pivô}$?
 - Se sim; avançar *part*, trocar *i* com *part*, avançar *i*.
 - **Senão; avançar *i*.**
- Ao final, avançar *part* e trocar *pivô* com *part*.

QUICKSORT



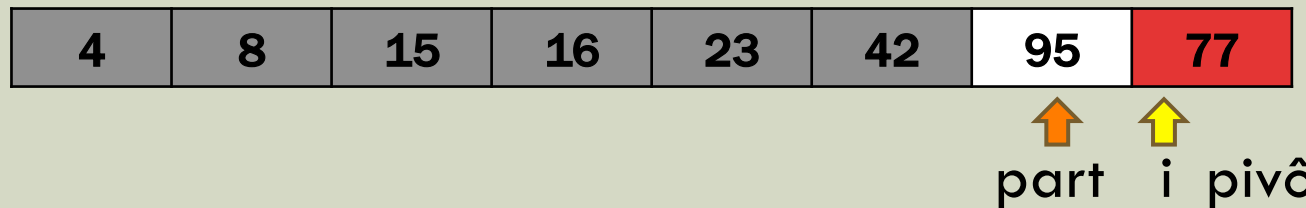
- Item $i < \text{pivô}$?
 - Se sim; avançar *part*, trocar *i* com *part*, avançar *i*.
 - **Senão; avançar *i*.**
- Ao final, avançar *part* e trocar *pivô* com *part*.

QUICKSORT



- Item $i < \text{pivô}$?
 - Se sim; avançar *part*, trocar *i* com *part*, avançar *i*.
 - Senão; avançar *i*.
- Ao final, avançar *part* e trocar *pivô* com *part*.

QUICKSORT



- Item $i < \text{pivô}$?
 - Se sim; avançar $part$, trocar i com $part$, avançar i .
 - Senão; avançar i .
- Ao final, avançar $part$ e trocar $pivô$ com $part$.

Downloaded from <http://ajph.org/> on November 10, 2014

↑ ↑
part i pivô

- PUC Minas - Engenharia de Software - Algoritmos e Estruturas de Dados - Prof.^a Eveline Alonso Veloso

QUICKSORT

4	8	15	16	23	42	77	95
---	---	----	----	----	----	----	----

ALGORITMO QUICKSORT

```
procedimento quicksort(vetor v, int inicio, int fim){  
    int part;  
    if (inicio < fim){  
        part = particao(v, inicio, fim);  
        quicksort(v, inicio, part-1);  
        quicksort(v, part+1, fim);  
    }  
}
```

ALGORITMO QUICKSORT

```
funcao particao(vetor v, int inicio, int fim){  
    int pivot = v[fim];  
    int part = inicio-1;  
    para i de inicio até fim-1 faça {  
        se (v[i] < pivot) então {  
            part = part + 1;  
            trocar v[part] e v[i];  
        }  
    }  
    trocar v[part+1] e v[fim];  
    retornar (part + 1);  
}
```

CONSIDERAÇÕES

- O método não é estável.
- Má escolha do pivô;
 - pode levar o algoritmo a complexidades maiores.
- Pode não se comportar bem;
 - em vetores “quase ordenados”.