

Estrutura de Dados Avançada

Daniel de Sousa Moraes
danielmoraes14@gmail.com

QuickSort

- Aplica o paradigma de divisão e conquista
- A cada iteração escolhe um elemento (pivô) e subdivide o vetor em duas partes, ordenando os subvetores em relação ao pivô.
- Ordena o vetor particionando recursivamente

Partição

- A chave para o algoritmo é o particionamento, que reorganiza os subvetores.
- Localiza o pivô e divide o vetor em duas partes, garantindo:
 - Todos os elementos à esquerda do pivô são **menores ou iguais** a ele.
 - Todos os elementos à direita do pivô são **maiores** que ele.

Partição

```
function partition(V, left, right)
    pivot = V[left]
    i = left-1
    j = right+1
    while true do
        repeat j=j-1 until V[j]<=pivot
        repeat i=i+1 until V[i]>=pivot
        if i<j then
            V[i], V[j] = V[j], V[i]
        else
            return j
        end
    end
end
end
```

QuickSort

```
function quicksort(V, left, right)
  if left < right then
    middle = partition(V, left, right)
    quicksort(V, left, middle)
    quicksort(V, middle+1, right)
  end
end
```

QuickSort – quicksort(1,7)

4	2	5	3	6	7	1
1	2	3	4	5	6	7

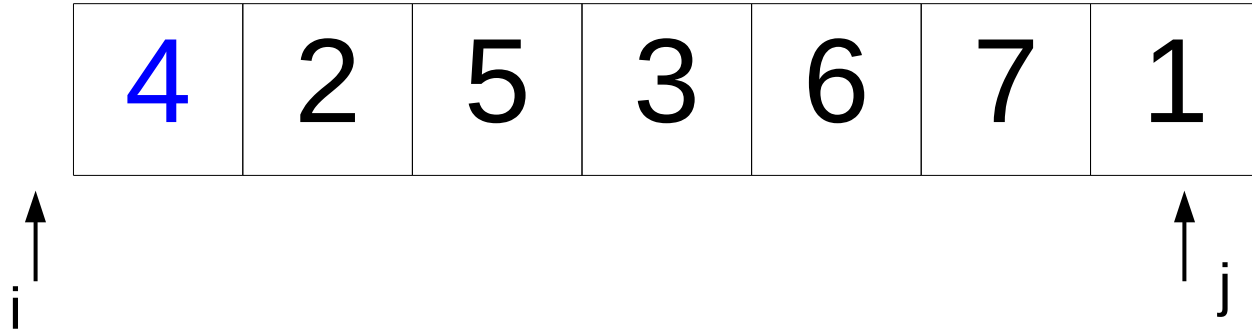
QuickSort – partition(1,7)

4	2	5	3	6	7	1
---	---	---	---	---	---	---

i ↑

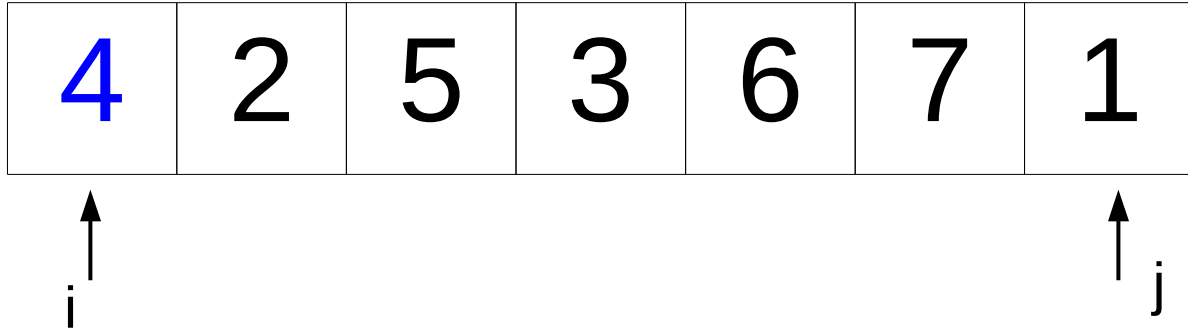
\uparrow
j

QuickSort – partition(1,7)



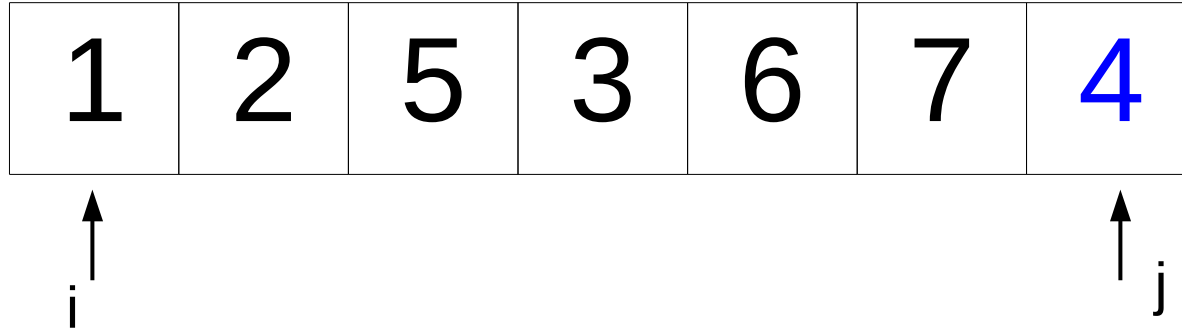
Until $V[j] \leq 4$?

QuickSort – partition(1,7)



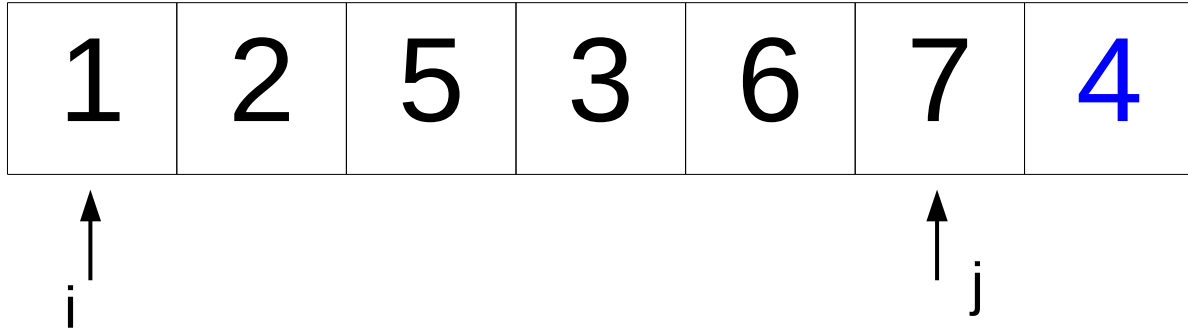
Until $V[i] \geq 5$?

QuickSort – partition(1,7)



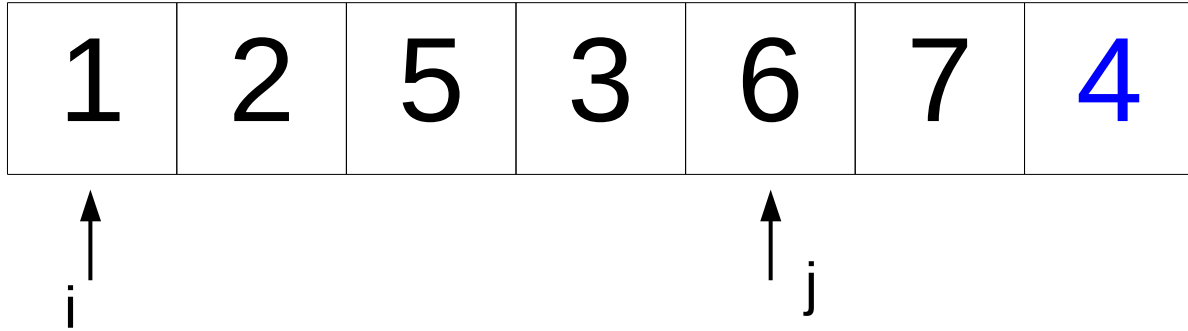
$$i < j ?$$
$$V[i], V[j] = V[j], V[i]$$

QuickSort – partition(1,7)



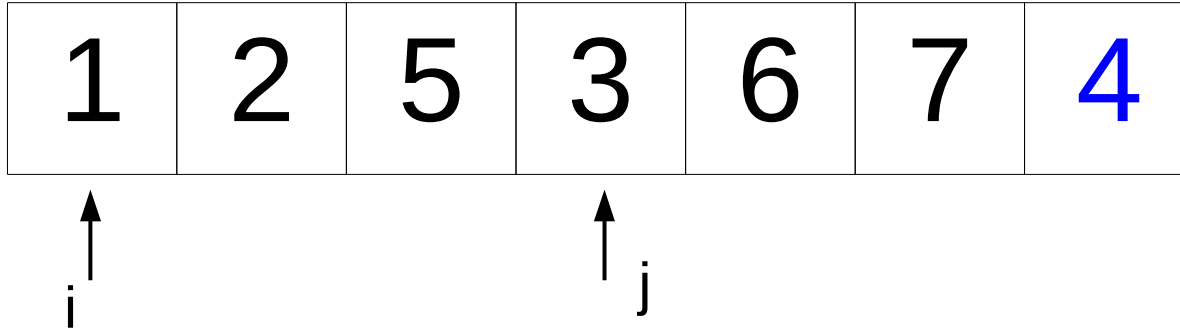
Until $v[j] \leq 4$?

QuickSort – partition(1,7)



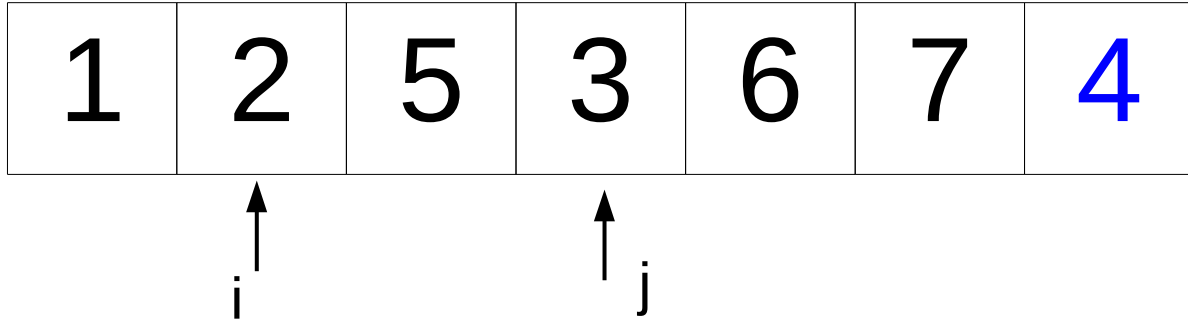
Until $v[j] \leq 4$?

QuickSort – partition(1,7)



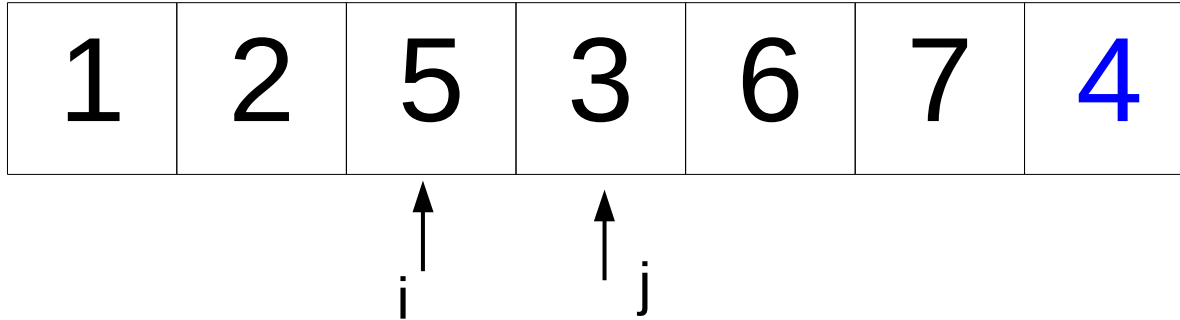
Until $v[j] \leq 4$?

QuickSort – partition(1,7)



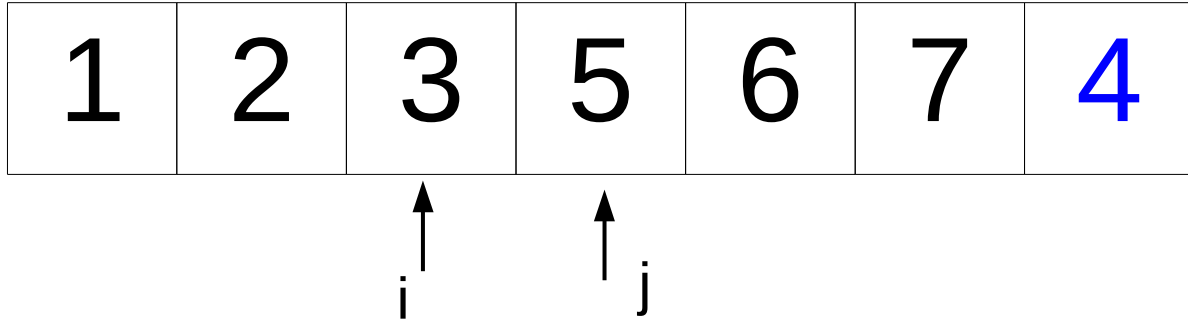
Until $v[i] \geq 4$?

QuickSort – partition(1,7)



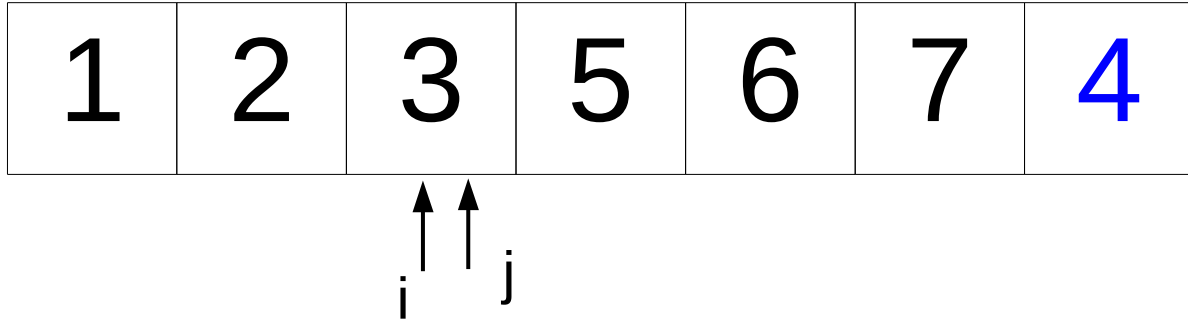
Until $v[i] \geq 4$?

QuickSort – partition(1,7)



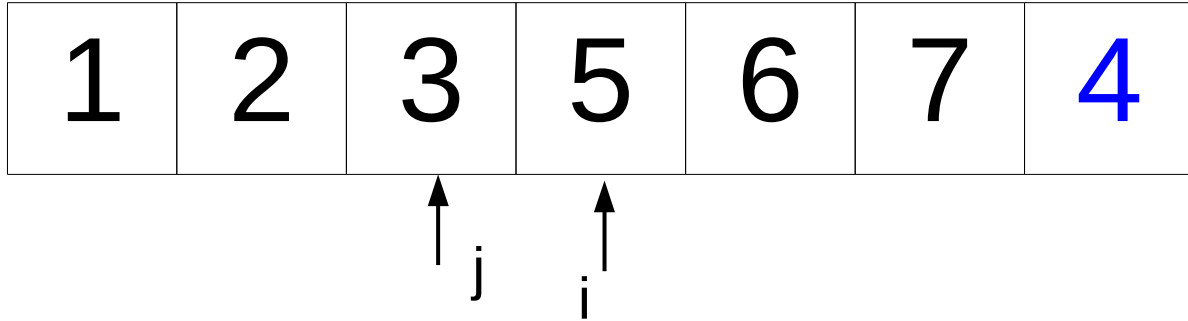
$i < j ?$
 $V[i], V[j] = V[j], V[i]$

QuickSort – partition(1,7)



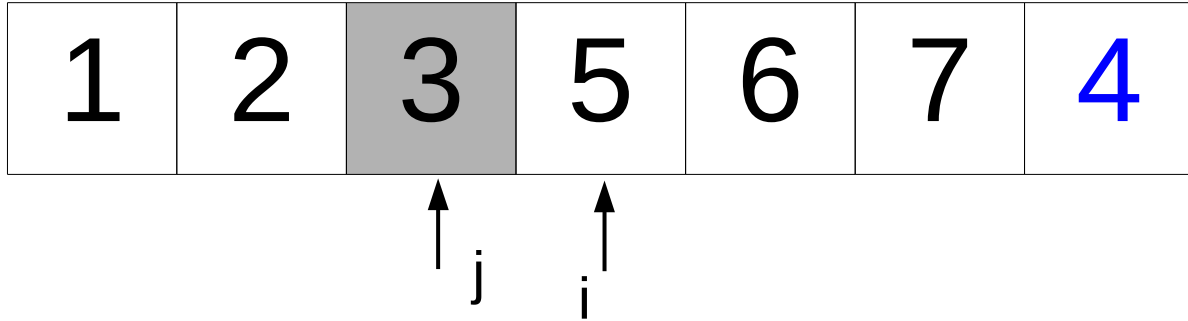
Until $v[j] \leq 4$?

QuickSort – partition(1,7)



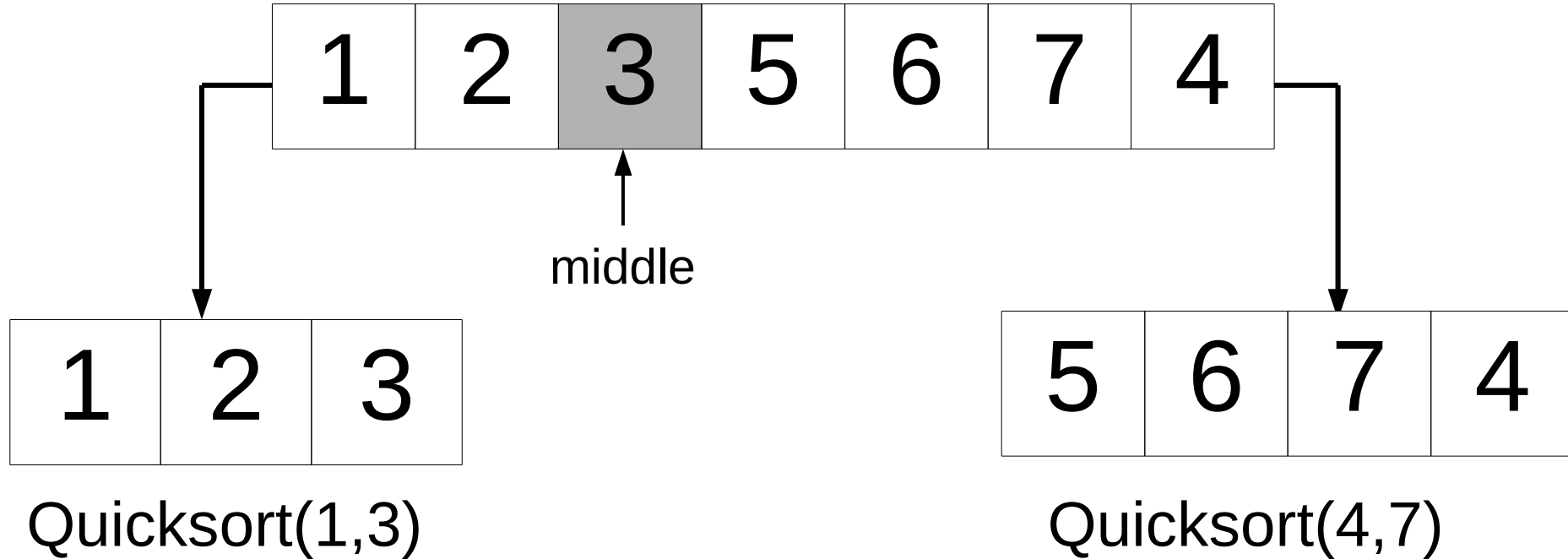
Until $v[i] \geq 4$?

QuickSort – partition(1,7)



$i < j$?
return j

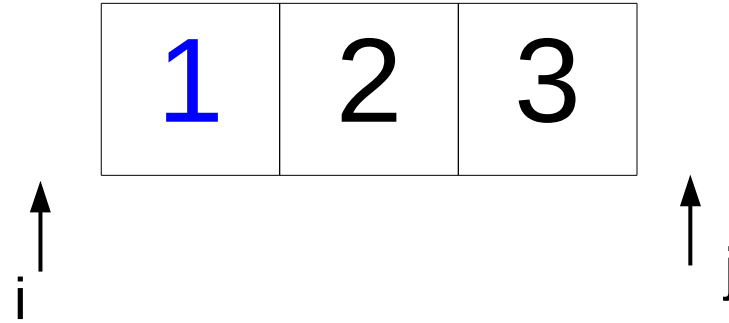
QuickSort – quicksort(1,7)



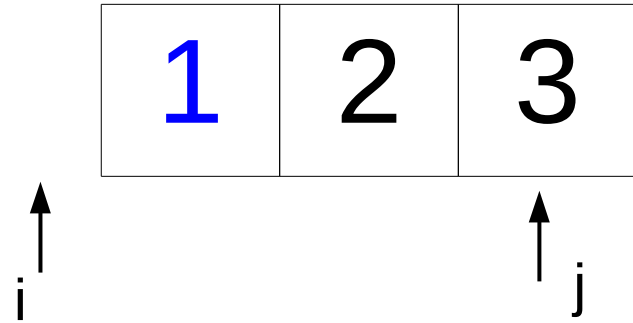
QuickSort – quicksort(1,3)

1	2	3
1	2	3

QuickSort – partition(1,3)

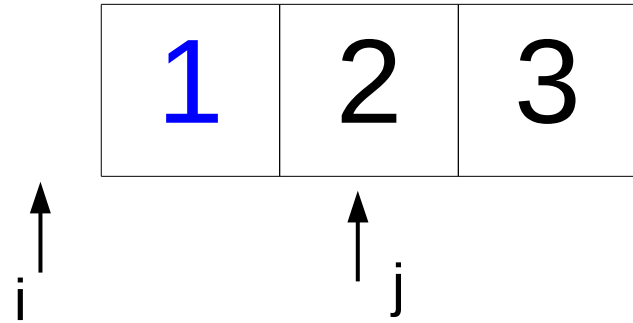


QuickSort – partition(1,3)



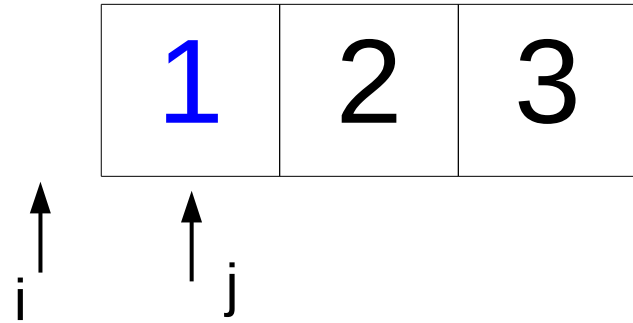
Until $v[j] \leq 1$?

QuickSort – partition(1,3)



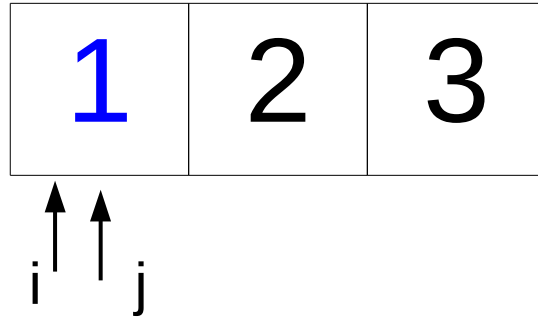
Until $v[j] \leq 1$?

QuickSort – partition(1,3)



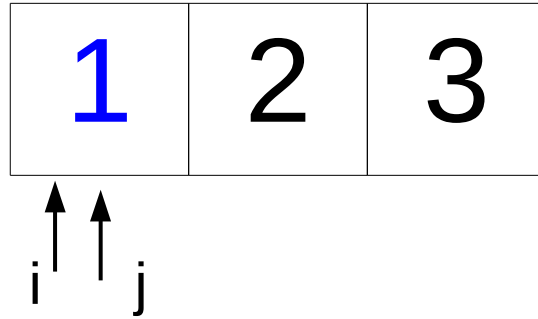
Until $v[j] \leq 1$?

QuickSort – partition(1,3)



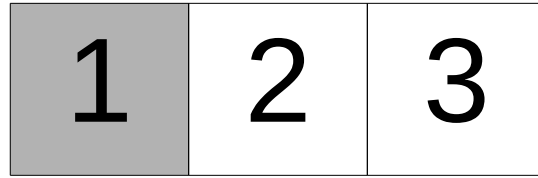
Until $v[i] \geq 1$?

QuickSort – partition(1,3)



$i < j$?
Return j

QuickSort – quicksort(1,3)



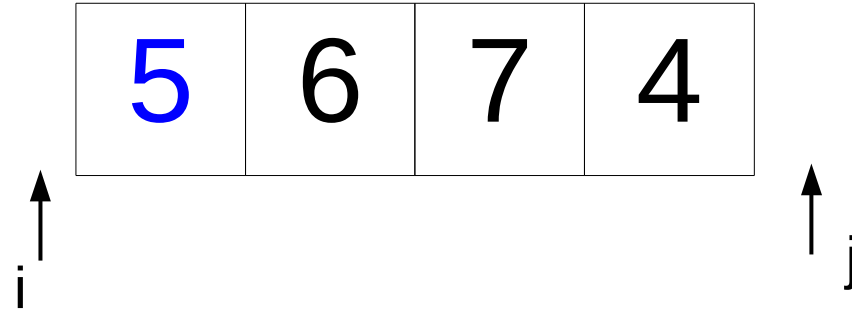
↑
middle

Quicksort(1,1)
Quicksort(2,3)

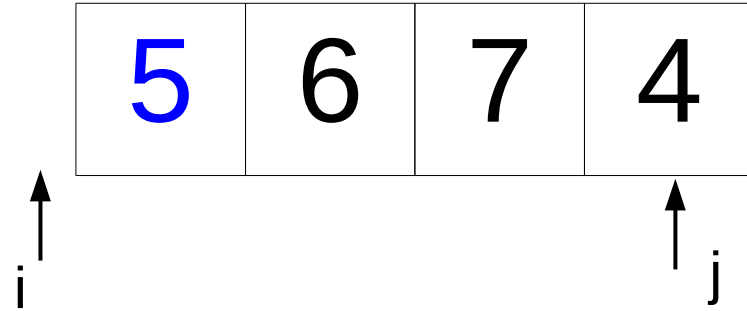
QuickSort – quicksort(4,7)

5	6	7	4
1	2	3	4

QuickSort – partition(4,7)

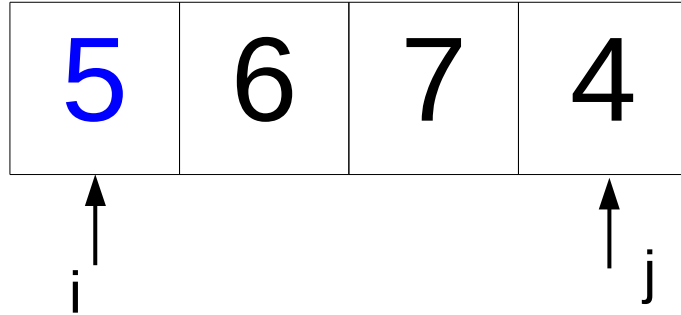


QuickSort – partition(4,7)



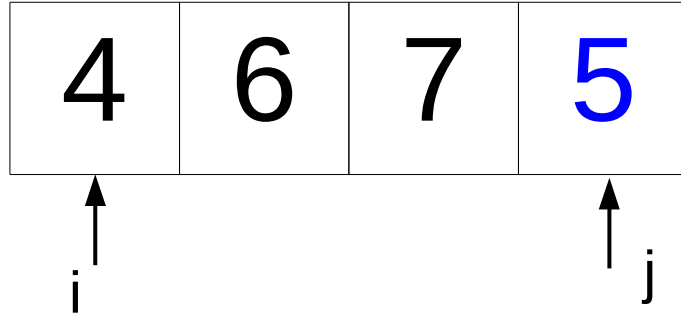
Until $v[j] \leq 5$?

QuickSort – partition(4,7)



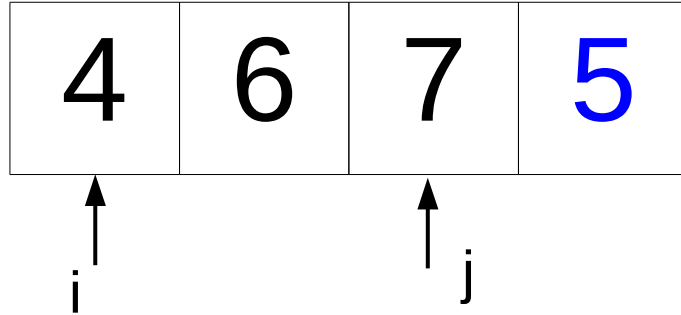
Until $v[i] \geq 5$?

QuickSort – partition(4,7)



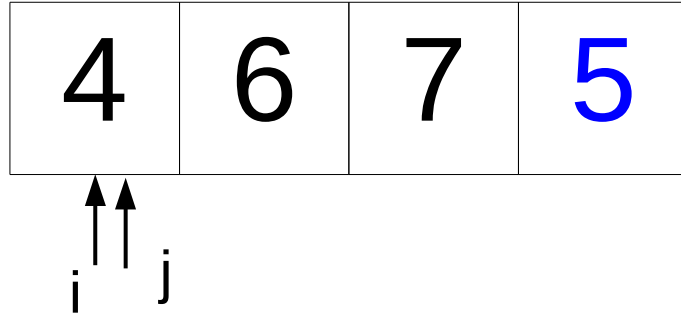
$i < j ?$
 $v[i], v[j] = v[j], v[i]$

QuickSort – partition(4,7)



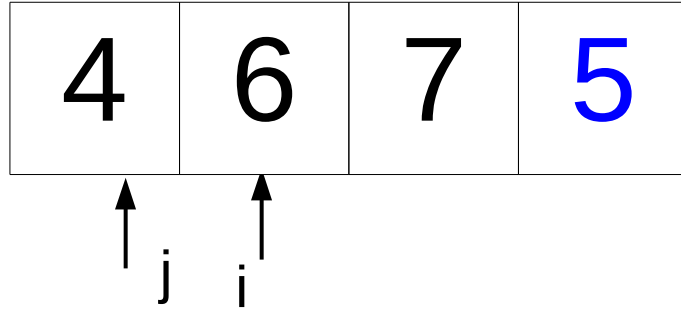
Until $v[j] \leq 5$?

QuickSort – partition(4,7)



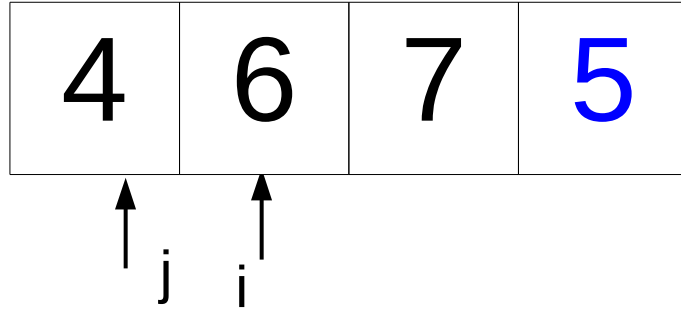
Until $v[j] \leq 5$?

QuickSort – partition(4,7)



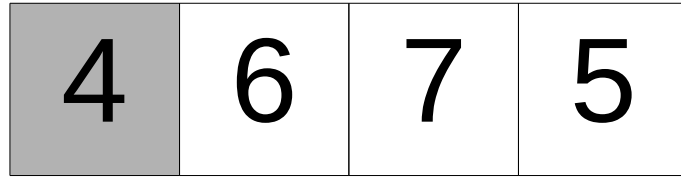
Until $v[i] \geq 5$?

QuickSort – partition(4,7)



$i < j$?
Return j

QuickSort – quicksort(4,7)

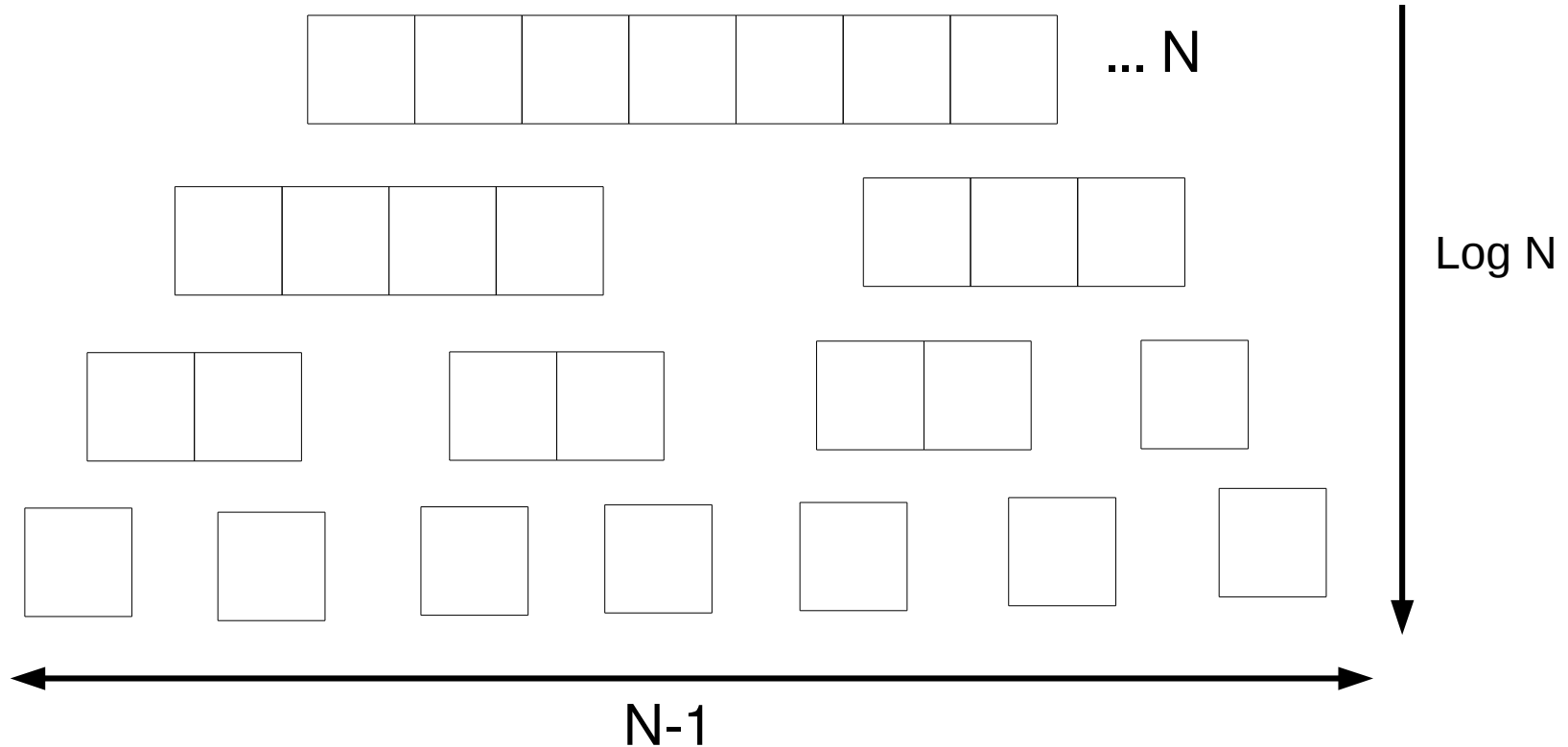


↑
middle

Quicksort(4,4)

Quicksort(5,7)

QuickSort



QuickSort

- $O(N \lg N)$
- $O(N^2)$
- Para evitar que chegue em N^2 operações deve-se melhorar a escolha do pivô (aleatório, valor médio de um subconjunto)

Bibliografia

Cormen, Thomas H. et al. Algoritmos.; [tradução Arlete Simille]. 3ª ed
- Rio de Janeiro - Elsevier, 2011.

Carlos de Salles Soares Neto – Notas de Aula da Disciplina de ED II
- UFMA