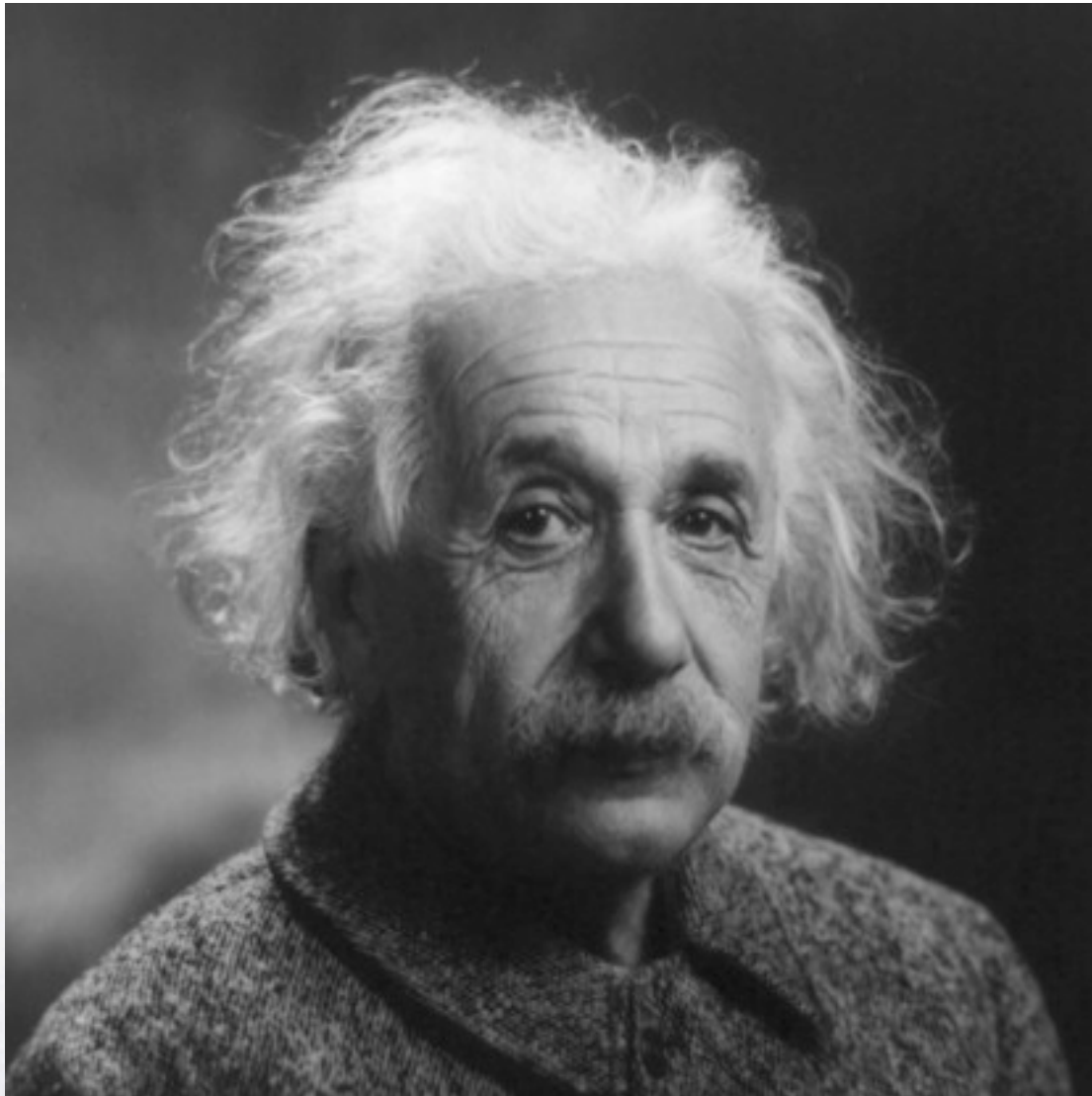




Ordenação de dados

Glauco Vinicius Scheffel, Msc

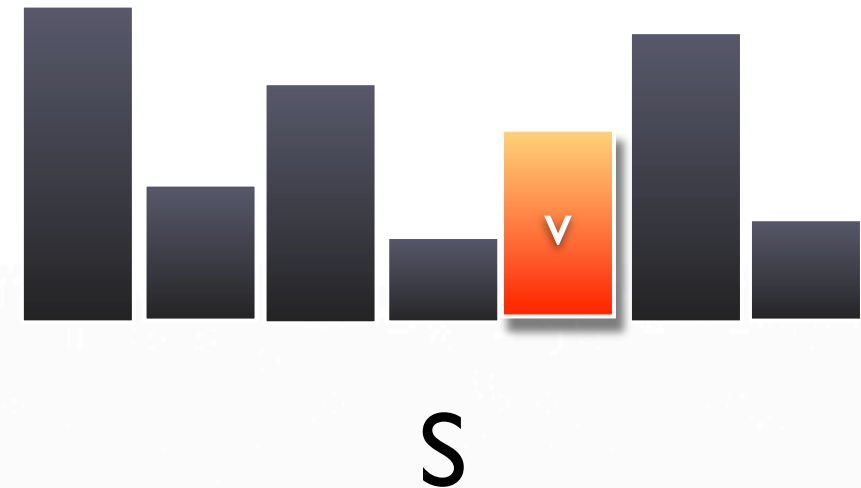


"Eu nunca ensino aos meus alunos. Somente tento criar condições nas quais eles possam aprender." Albert Einstein



Quicksort

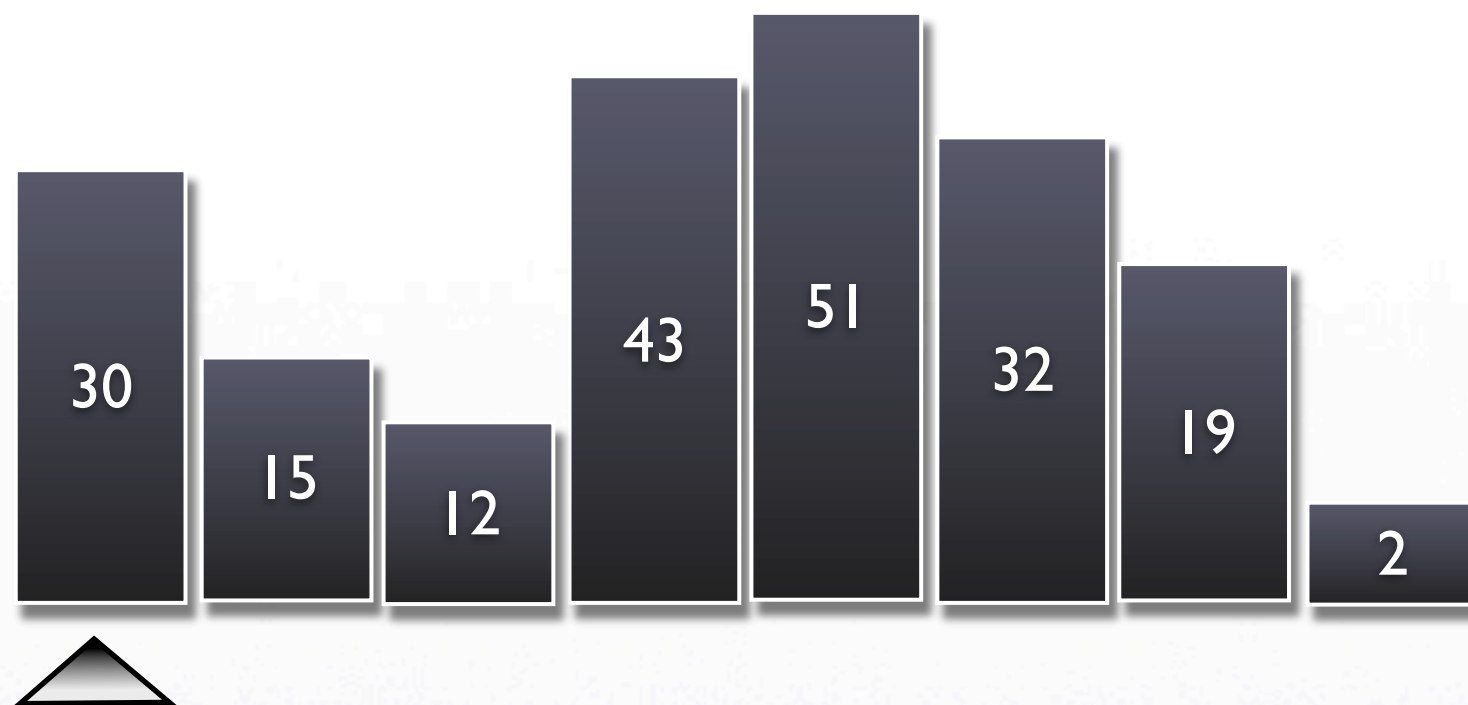
- Dividir:
 - Escolha um elemento (pivô) v no conjunto S
 - Particione $S - \{v\}$ em dois conjuntos disjuntos
$$S1 = \{x \in S - \{v\} \mid x \leq v\}$$
$$S2 = \{x \in S - \{v\} \mid x \geq v\}$$
- Conquistar: recursivamente ordene $S1$ e $S2$
- Combinar: $S1$ ordenado com v seguido de $S2$ ordenado



Esta é a versão simplificada onde não existem elementos duplicados nos conjuntos



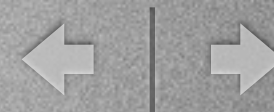
Quicksort - selecionar pivô



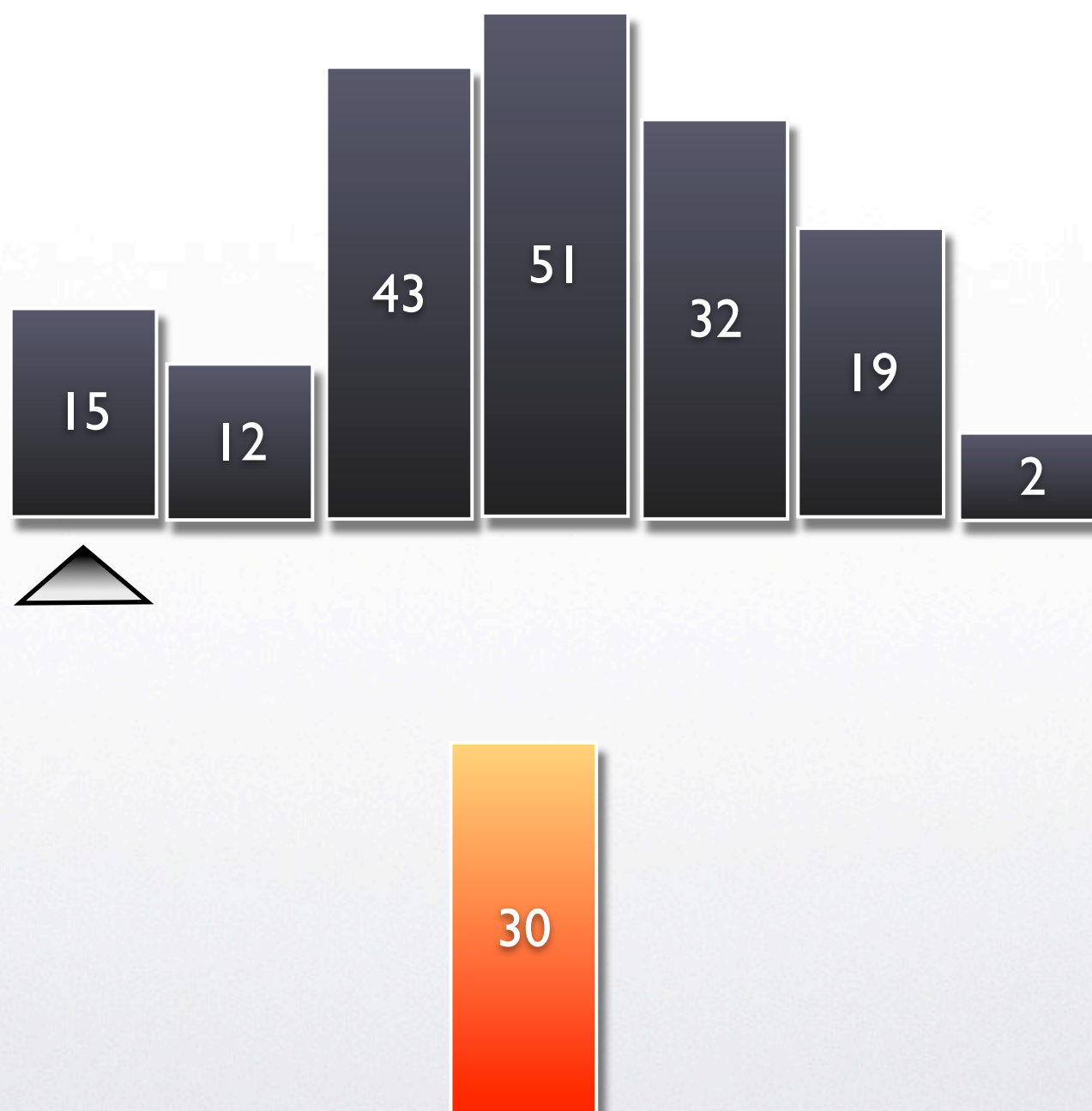


Quicksort - selecionar pivô





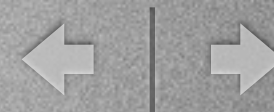
Quicksort - criar conjuntos



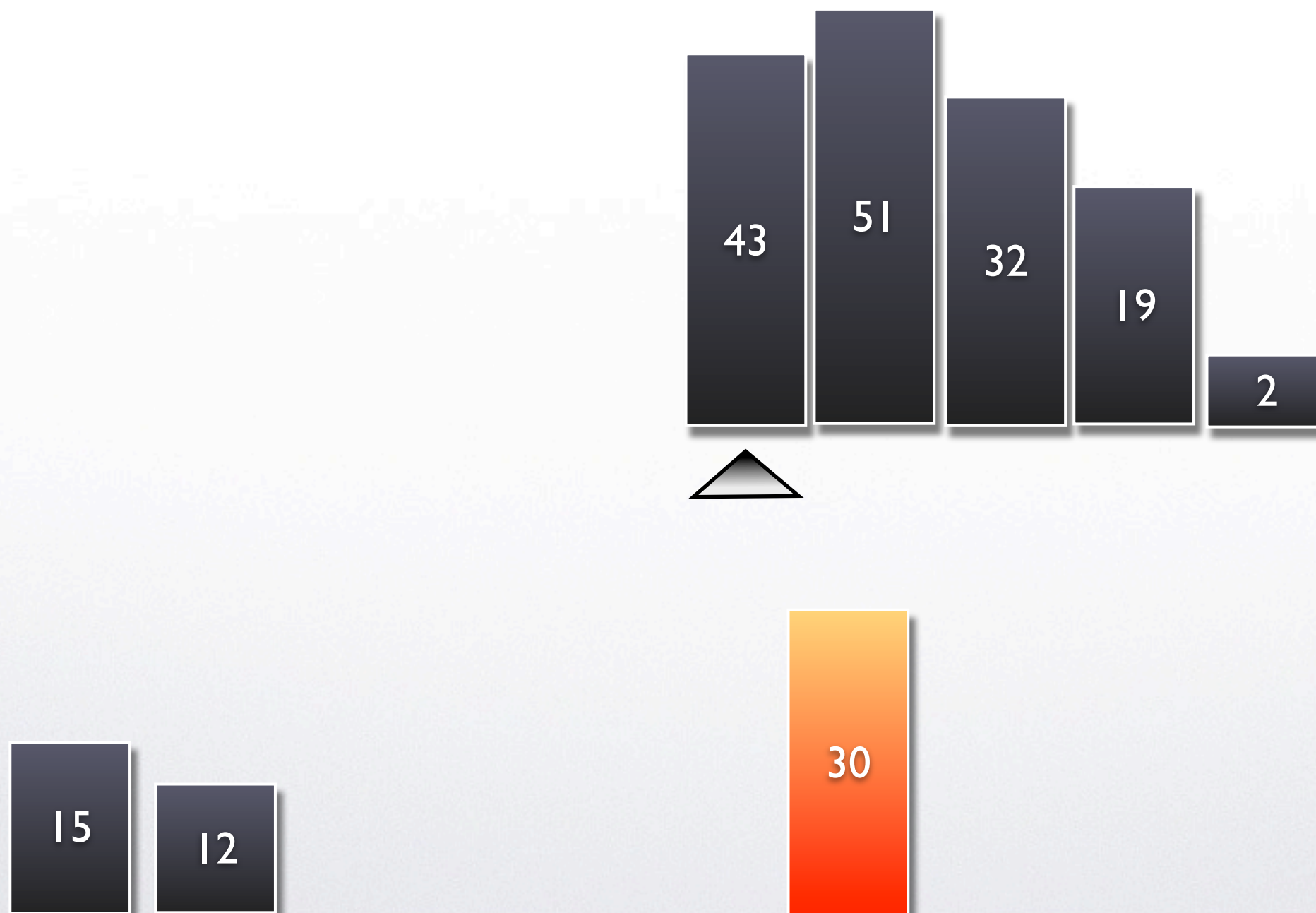


Quicksort - criar conjuntos



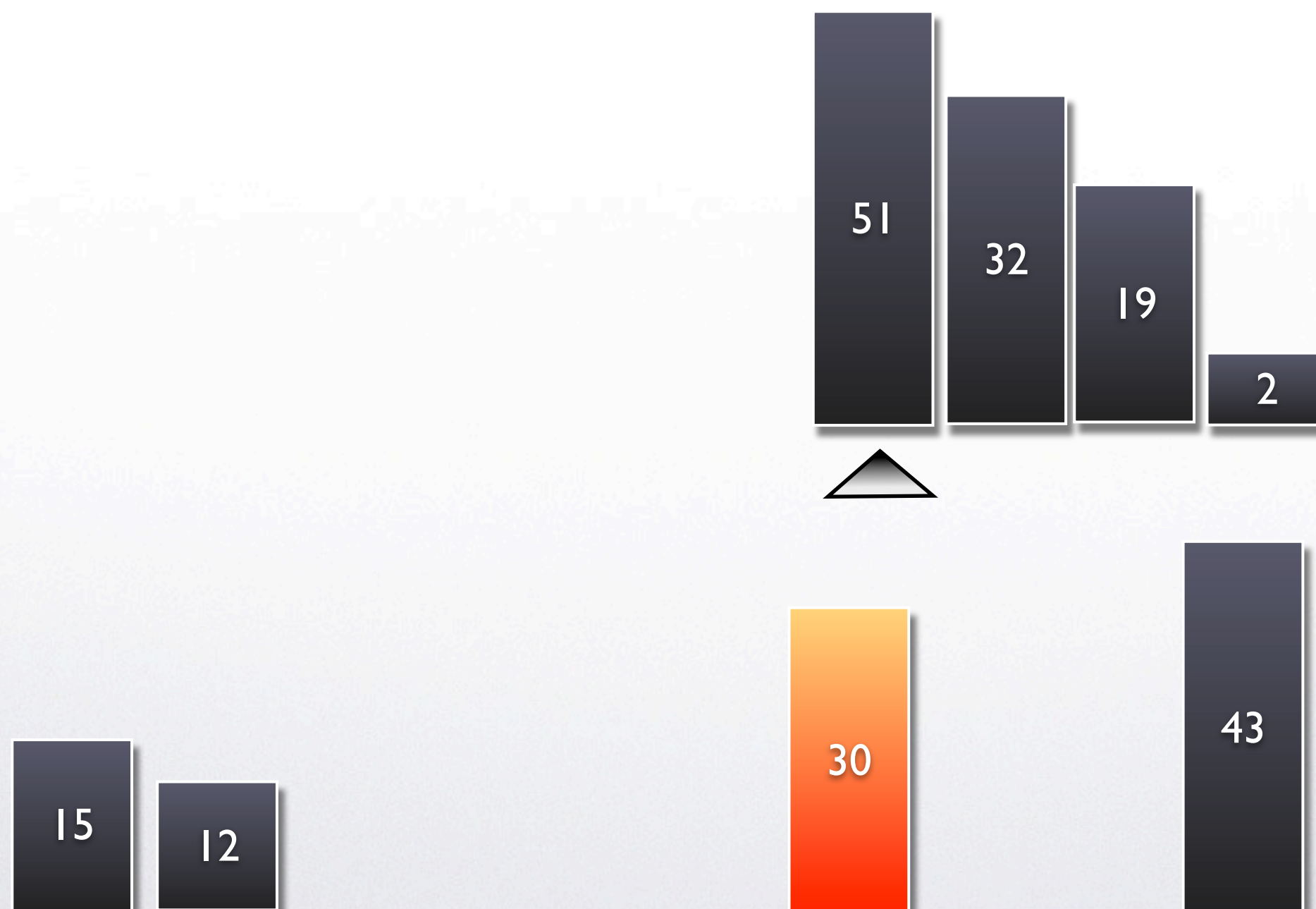


Quicksort - criar conjuntos



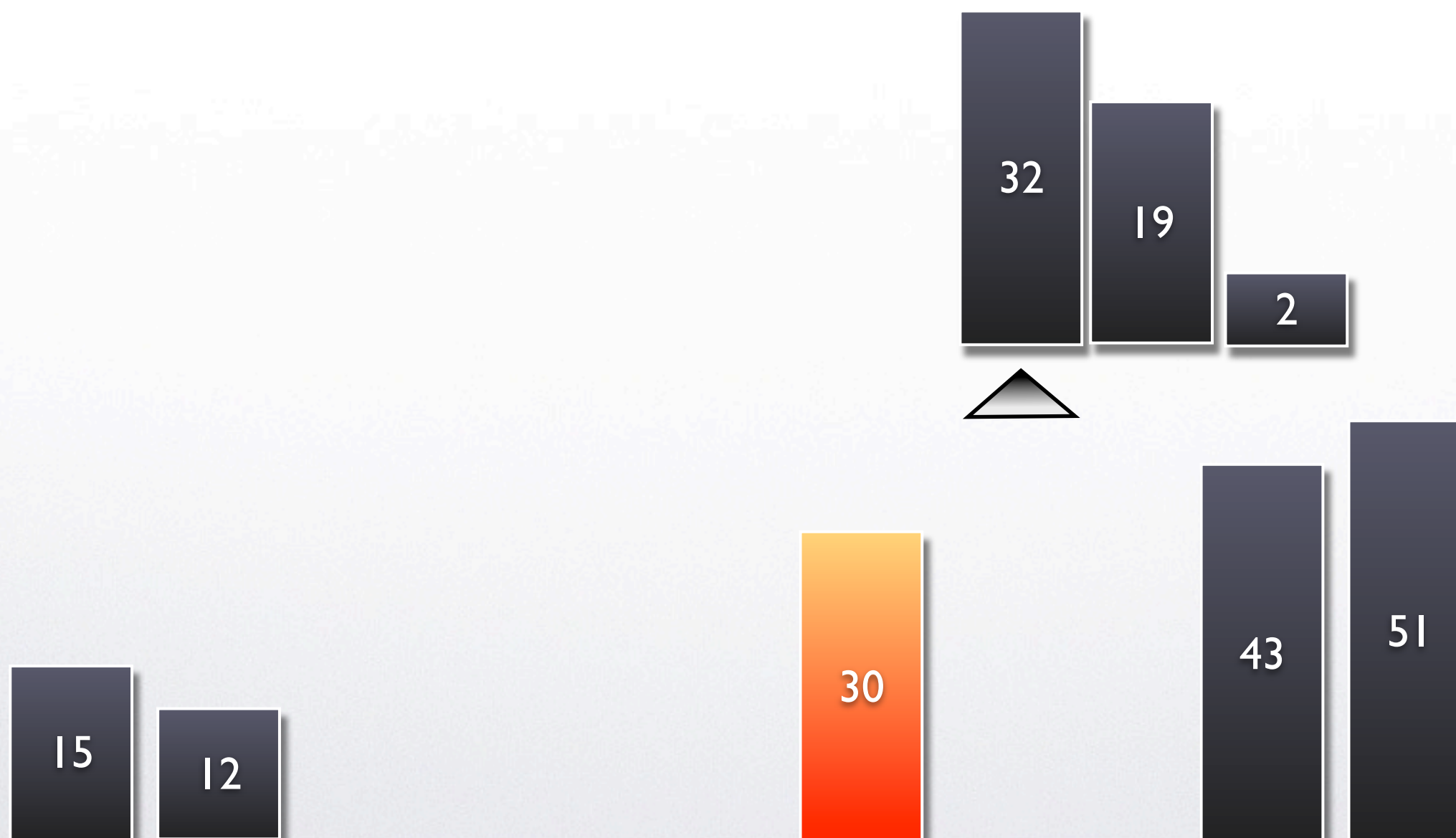


Quicksort - criar conjuntos



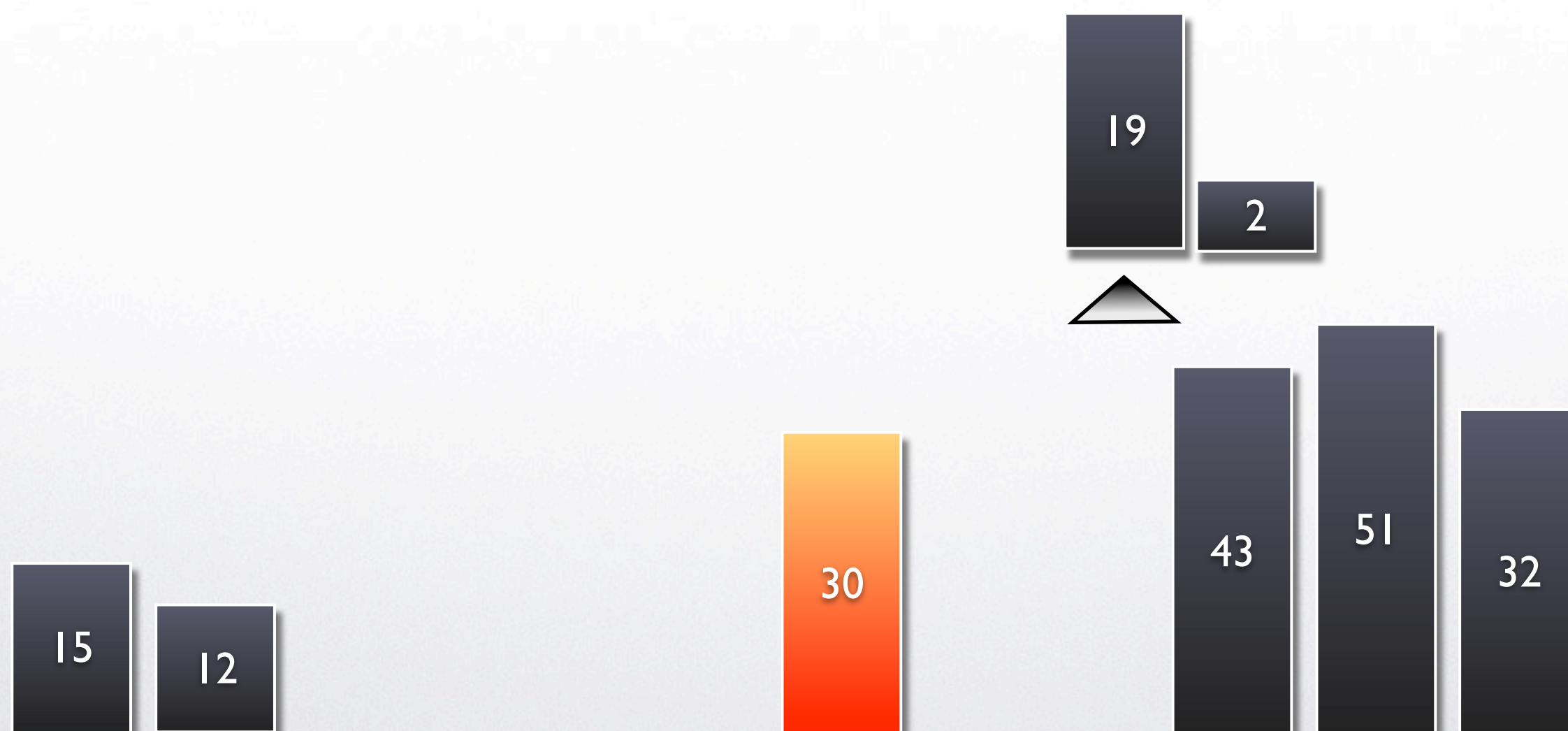


Quicksort - criar conjuntos



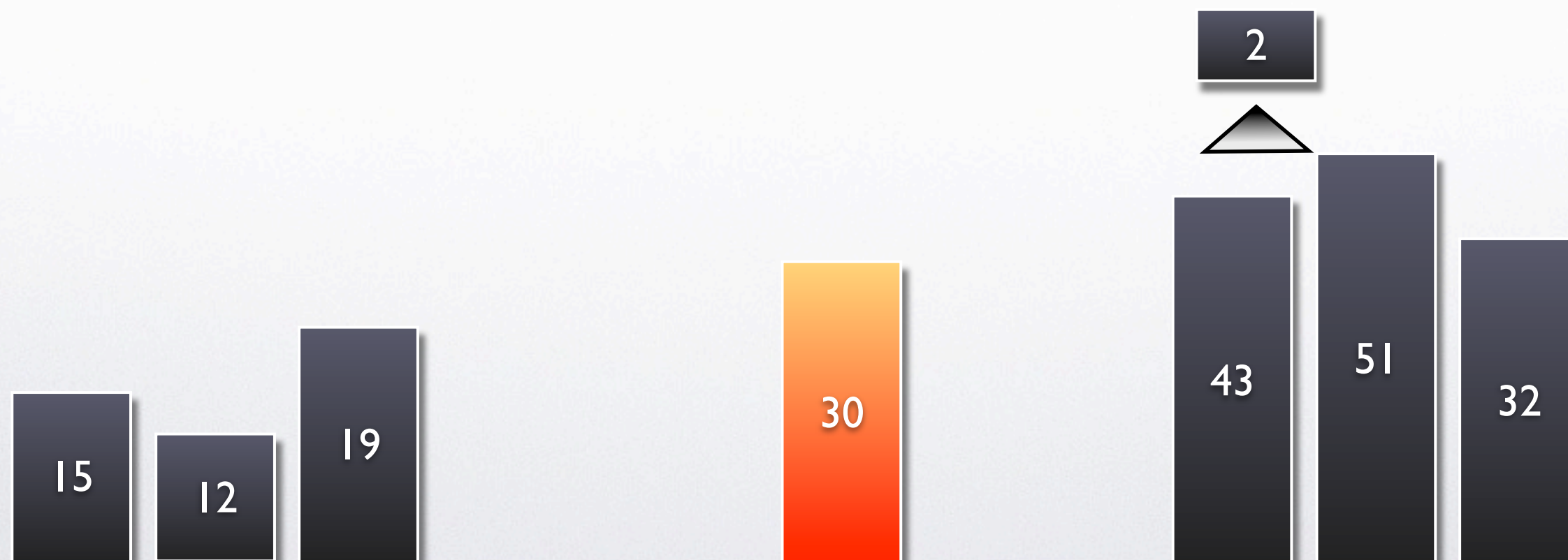


Quicksort - criar conjuntos



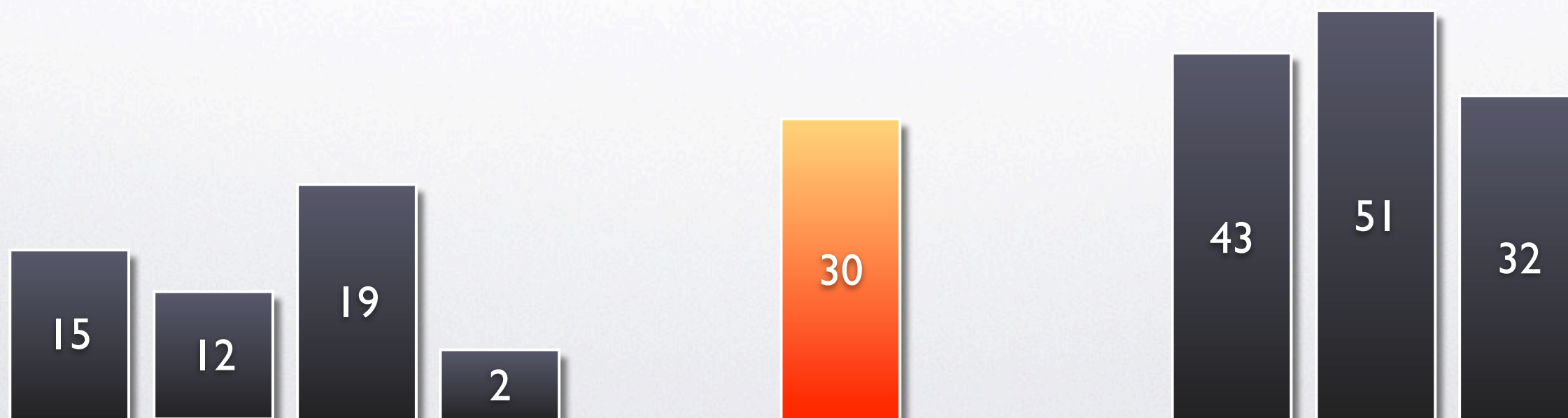


Quicksort - criar conjuntos



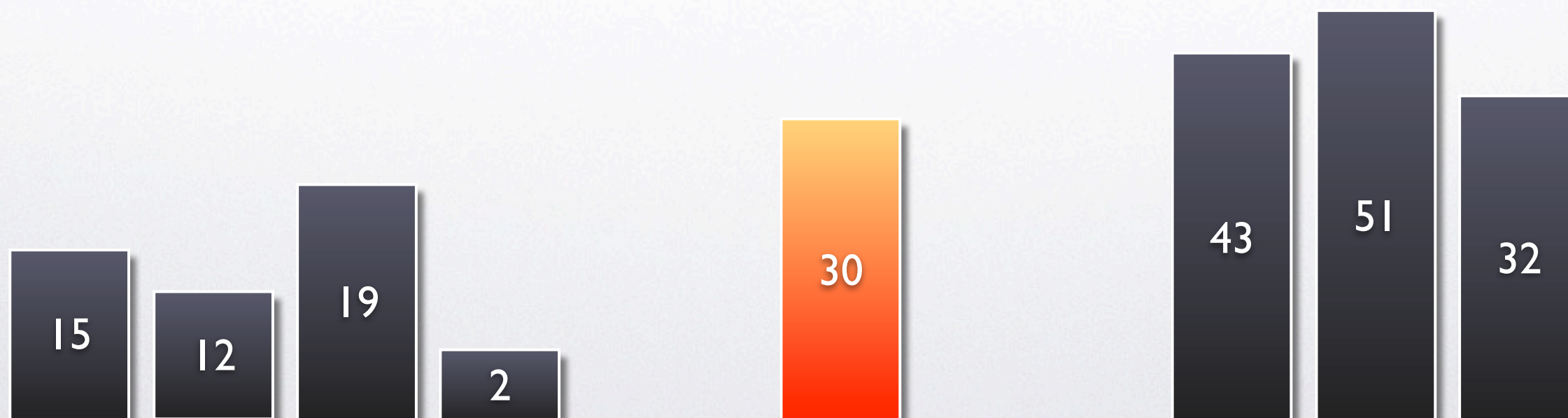


Quicksort - criar conjuntos



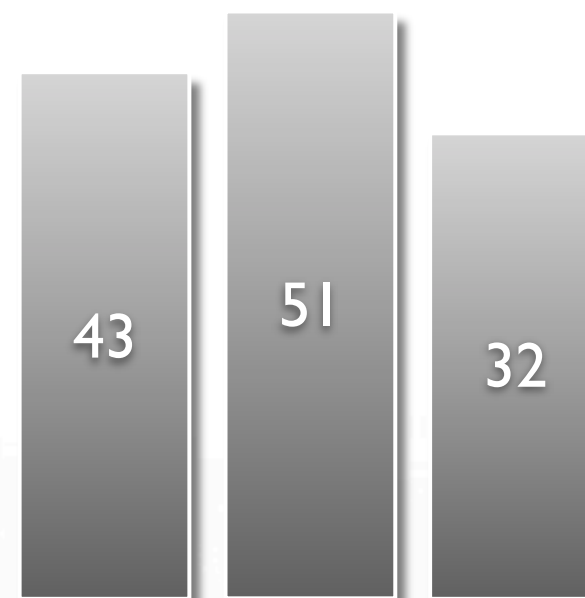
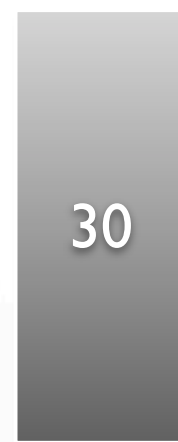
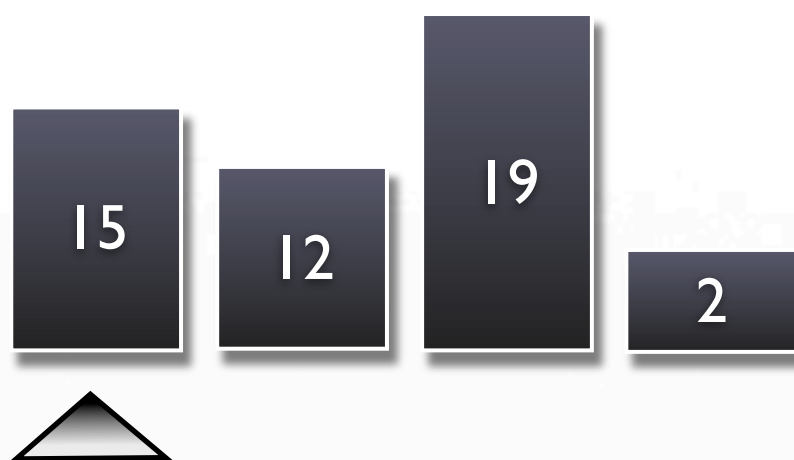


Quicksort - criar conjuntos



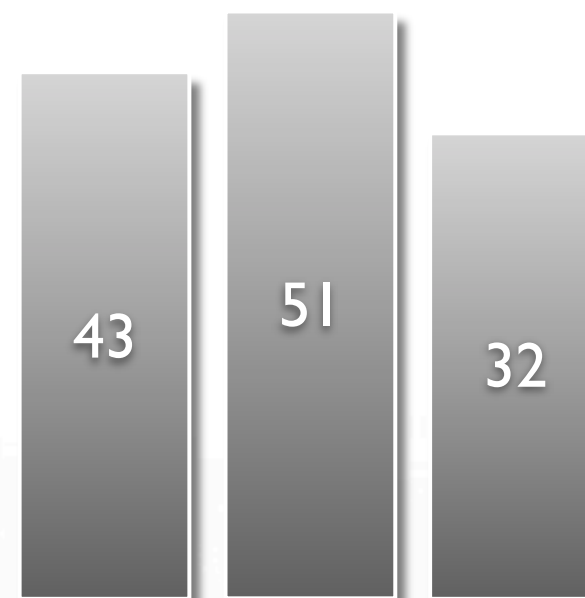
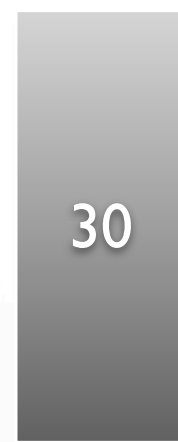
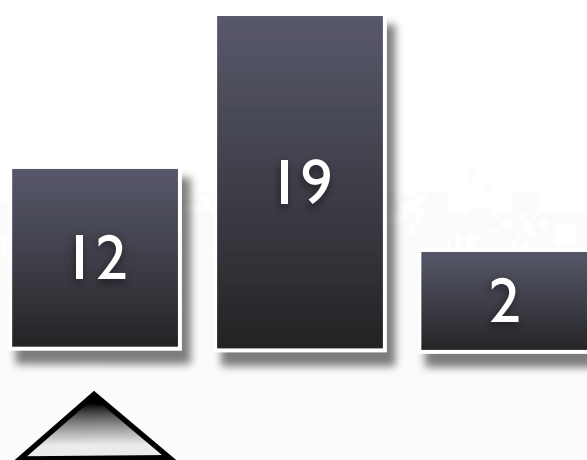


Quicksort - novo pivô



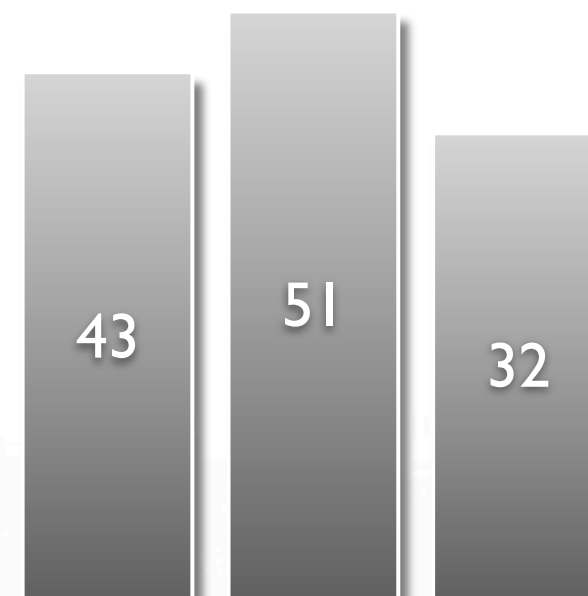
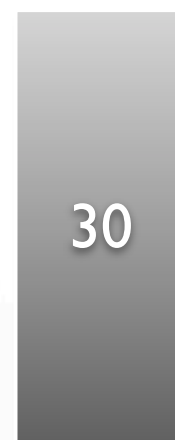
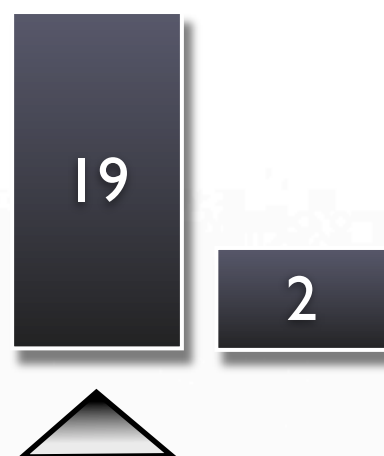


Quicksort - criar conjuntos





Quicksort - criar conjuntos



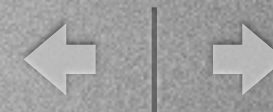
12

15

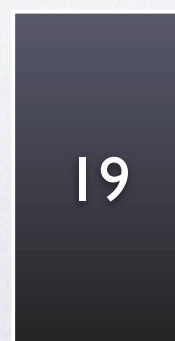
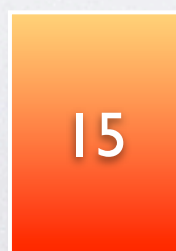
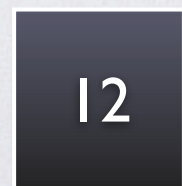
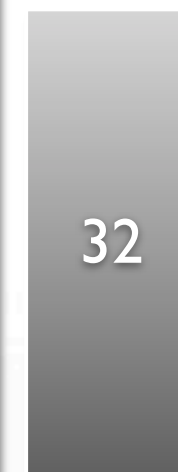
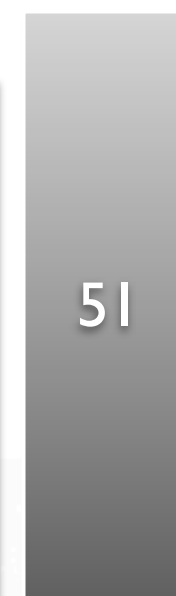
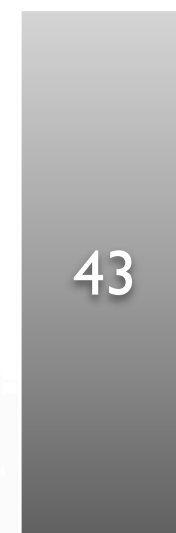
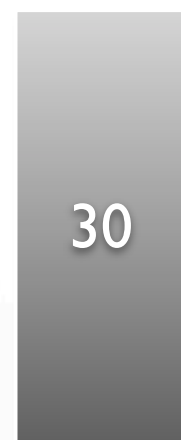


Quicksort - criar conjuntos





Quicksort - criar conjuntos





QuickSort

```
quicksort(a,left,right) {  
    if (right>left) {  
        pivotIndex = left;  
        selecionar o pivô a[pivotIndex];  
        pivotNewIndex=partition  
        (a,left,right,pivotIndex);  
  
        quicksort(a,left,pivotNewIndex-1);  
        quicksort(a,pivotNewIndex+1,right);  
    }  
}
```




Definindo o pivô

- Use o primeiro elemento
 - Se o conjunto for randômico deve funcionar
 - Se o conjunto for pré-ordenado ou em ordem decrescente
 - Todos elementos vão para S1 ou S2
 - O problema irá ocorrer em todas chamadas
 - Performance será semelhante a do bubble
- Escolher o elemento randomicamente
 - Caro
 - Normalmente funciona



Definindo o pivô

- Use a mediana do array (difícil de encontrar)
- Compare apenas 3 elementos
 - O primeiro,
 - O último,
 - O Central
- Para estes 3 elementos garanta que
 - Na primeira posição ficará o menor dos 3
 - Na última posição ficará o maior dos 3
 - E no meio aquele que será usado como pivô



Particionamento

- Usar um array adicional
 - Semelhante ao merge
 - Fácil de programar
 - Ineficiente para o Quick



Partition

(wikipedia sample)

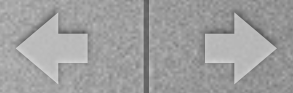
```
int partition(a, left, right, pivotIndex) {  
    pivotValue = a[pivotIndex];  
    swap(a[pivotIndex], a[right]); // Move pivô para o fim  
    // menores que o pivô vão para o início  
    storeIndex = left;  
    for (i from left to right) {  
        if a[i] < pivotValue  
            swap(a[storeIndex], a[i]);  
            storeIndex = storeIndex + 1 ;  
    }  
    swap(a[right], a[storeIndex]); // pivô no seu local  
    return storeIndex;  
}
```




Questões

(resolver em sala)

1. Explique como funciona o QuickSort?
2. Por que QuickSort é considerado não estável?
3. Quais técnicas podem ser usadas para escolher o pivô?
4. Detalhe cada uma das técnicas da questão anterior?
5. Quem criou o QuickSort? Quando? Para quê?
6. Do que depende o tempo de execução do QuickSort? Por quê?
7. No pior caso o QuickSort tem a sua execução comparável a qual algoritmo simples?
8. O que ocorre se a cada seleção o QuickSort usar como pivô o menor elemento do grupo?

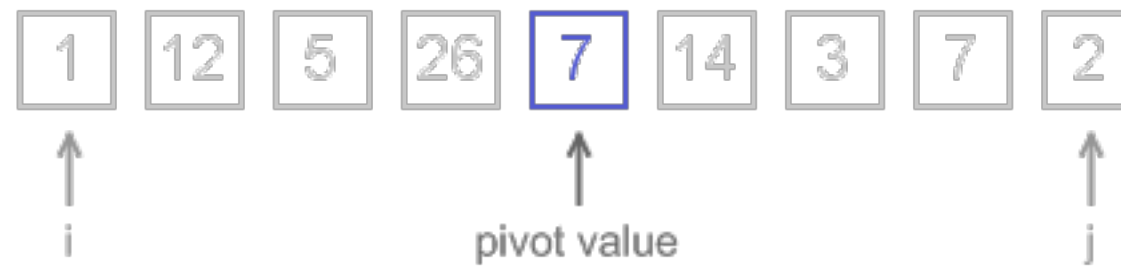


1 12 5 26 7 14 3 7 2 unsorted

1 12 5 26 7 14 3 7 2 pivot value = 7
↑ pivot value ↑
i j

1 12 5 26 7 14 3 7 2 12 ≥ 7 ≥ 2, swap 12 and 2
↑ ↑
i j

1 2 5 26 7 14 3 7 12 26 ≥ 7 ≥ 7, swap 26 and 7



pivot value = 7



$12 \geq 7 \geq 2$, swap 12 and 2



$26 \geq 7 \geq 7$, swap 26 and 7



$7 \geq 7 \geq 3$, swap 7 and 3



$i > j$, stop partition



"Definir uma palavra é
capturar uma borboleta
no ar."

Aurélio Buarque de Holanda Ferreira