QUICKSORT

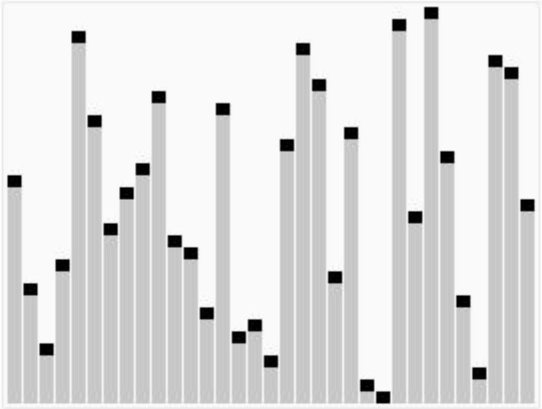
Ewelly Fabiane Cunha de Sousa
Miller Raycell Monteiro Correia

Criação do quicksort

- Criado por Charles Anthony Richard Hoard em 1960
- Propôs o algoritmo para realizar a tradução de um dicionário do russo para o inglês
- Publicou o algoritmo em 1962 após melhoramentos

Método do quicksort

- Algoritmo baseado no método de divisão e conquista
- O algoritmo funciona escolhendo um pivô, que separa o vetor em duas partições
- Rearranja a lista de forma que todos os elementos anteriores ao pivô sejam menores que ele, e todos os elementos posteriores ao pivô sejam maiores que ele. Ao fim do processo o pivô estará em sua posição final e haverá duas sub listas não ordenadas. Essa operação é denominada *partição*
- Recursivamente ordene a sub lista dos elementos menores e a sublista dos elementos maiores;



Pseudo código

```
algorithm quicksort(A, lo, hi) is
    if lo < hi then
        p := particiona(A, lo, hi)
        quicksort(A, lo, p – 1)
        quicksort(A, p + 1, hi)
```

```
algorithm particiona(A, lo, hi) is
    pivot := A[hi]
    i := lo - 1
    for j := lo to hi - 1 do
        if A[j] < pivot then
            i := i + 1
            swap A[i] with A[j]
    if pivot < A[i + 1] then
        swap A[i + 1] with A[hi]
    return i + 1
```

Cálculo da função de tempo

$$T(n) <= 2T(n/2) + \Theta(n)$$

$$<= 2T(n/2) + n$$

$$<= 2[2T(n/2/2) + n/2] + n$$

$$<= 4T(n/4) + n + n$$

$$<= 4[2T(n/4/2) + n/4] + n + n$$

Cálculo da função de tempo

$<= 8T(n/8) + n + n + n$

$<= 2^k T(n/2^k) + kn$ $\quad$ $, com\ n = 2^k$

$<= nT(n/n) + kn$ $\quad$ $, k = lg(n)$

$<= nT(1) + nlgn$ $\quad$ $, T(1) = 0$

$T(n) <= nlgn$

Cálculo da função de tempo

$T(n) <= nlgn$

Complexidade = $O(nlgn)$