**Algoritmo:** QuickSort(e, d, V)

**Objetivo:**

Permite ordenar um vetor de números [0,20]

pelo método denominado Quicksort.

Algoritmo:

- Dado o vetor V[e, d].

- Dividir em dois sub-vetores: V[e, p-1] e V[p+1, e]

e um pivot V[p].

- Recursivamente efetuar os mesmo para os sub-vetores.

- O valor de p depende dos dados.

**Parâmetros**

**Entrada:**

e (Inteiro T6) - Posição esquerda do vetor (>= 1, <= 999999)

d (Inteiro T6) - Posição direita do vetor (>= 1, <= 999999)

V [N] (Inteiro T6) - Vetor (>= 0, <= 20)

**Saída:**

V [N] (Inteiro T2) - Vetor ordenado (>= 0, <= 20)

**Variáveis**

**Auxiliares:**

p (Inteiro T6) - Índice da partição (>= 1, <= N)

**Data:** 2011-11-06 17:47

**Autor:** Paulo Nunes

**Versão:** 1.1

**Obs:** A saída é mesmo vetor de entrada. Este é uma permutação ou reordenação do vetor de entrada.

**Início:**

/\* Processamento (PROCESSING) \*/

SE (e < d) ENTÃO

p = ParticaoQuickSort(V, e, d)

Quicksort(V, e, p-1)

Quicksort(V, p+1, d)

FIMSE

**Fim.**

**Algoritmo:** ParticaoQuickSort(e, d, V)

**Objetivo:**

Permite dividir um vetor V[e, d] em dois sub-vetores:

- V[e, p-1] e V[p+1, e]

- e um pivot V[p].

- Todos os elementos do sub-vetor V[e, p-1] são ≤ do que o pivot.

- Todos os elementos da sub-vetor V[p+1, e] são ≥ do que o pivot.

**Valor de retorno:**

p (Inteiro T6) - Posição do pivot (>= 1, <= 999999)

**Parâmetros**

**Entrada:**

e (Inteiro T6) - Posição esquerda do vetor (>= 1, <= 999999)

d (Inteiro T6) - Posição direita do vetor (>= 1, <= 999999)

V [N] (Inteiro T6) - Vetor (>= 0, <= 20)

**Saída:**

V [N] (Inteiro T2) - Vetor ordenado (>= 0, <= 20)

**Variáveis**

**Auxiliares:**

i (Inteiro T6) - Índice do vetor - encontra elementos > pivot (>= 1, <= d)

j ( T2) - Índice do vetor - encontra elementos < pivot (>= 1, <= d)

pivot ( T2) - Elemento pivot (>= 0, <= 20)

aux ( T2) - Usada para troca de elementos (>= 0, <= 20)

**Data:** 2011-11-06 18:41

**Autor:** Paulo Nunes

**Versão:** 1.0

**Obs:** A saída é mesmo vetor de entrada. Este é uma permutação ou reordenação do vetor de entrada.

**Início:**

**/\* Processamento (PROCESSING) \*/**

**i ← e**

**j ← d**

**pivot ← V[(e + d) div 2] /\* Escolhe o elemento central como pivot \*/**

**ENQUANTO (i <= j) /\* para quando se cruzam \*/**

**ENQUANTO (V[i] < pivot) FAZER**

**i ← i + 1**

**FIMENQUANTO**

**ENQUANTO (V[J] > pivot) FAZER**

**j ← j - 1**

**FIMENQUANTO**

**SE (i <= j) ENTÃO /\* Troca i com j \*/**

**aux ← X[i]**

**X[i] ← X[j]**

**X[j] ← aux**

**i ← i + 1**

**j ← j - 1**

**FIMSE**

**FIMENQUANTO**

**p ← i-1 /\* ou j+1 \*/**

**RETORNA p**

**Fim.**