

## Gabriel Machado – 23/06/2023 – Estrutura de Dados I – Trabalho Final

c) Como se comporta a pilha de execução (*callstack*) no algoritmo de *merge sort* para ordenar de modo crescente o seguinte vetor: [6,5,4,3,2,1]:

1. Primeiro passo realizado pelo algoritmo merge sort é a divisão:

- Chama-se a função de ordenação mergeSort com o vetor completo [6,5,4,3,2,1];
- A função mergeSort divide recursivamente o vetor em duas metades aproximadamente iguais até que reste apenas um elemento em cada parte.

Pilha de execução (*callstack*):

mergeSort([6, 5, 4, 3, 2, 1])

2. No segundo passo entramos nas chamadas recursivas

- A pilha de execução começa a empilhar as chamadas recursivas do mergeSort conforme o vetor é dividido.

Pilha de execução (*callstack*):

mergeSort([6, 5, 4, 3, 2, 1])

mergeSort([6, 5, 4])

mergeSort([6, 5])

mergeSort([6])

mergeSort([5])

mergeSort([4, 3, 2, 1])

mergeSort([4, 3])

mergeSort([4])

mergeSort([3])

mergeSort([2, 1])

mergeSort([2])

mergeSort([1])

3. No terceiro passo entramos no processo de fusão – merge:

- Neste estágio, as chamadas recursivas começam a ser desempilhadas e o vetor é combinado e ordenado gradualmente.

Pilha de execução (fusão):

- mergeSort([6, 5, 4, 3, 2, 1]) -> Desempilhado

Mesclamos o 6 e o 5: [5,6]

Mesclamos o 4 e o 3: [3, 4]

Mesclamos o 2 e o 1: [1, 2]

Mesclamos o 'grupo' 3, 4 com o 'grupo' 1, 2: [1, 2, 3, 4]

Mesclamos o 'grupo' 5, 6 com o 'grupo' 1, 2, 3, 4: [1, 2, 3, 4, 5, 6]

- mergeSort([6, 5, 4]) -> Desempilhado

Mesclamos o 6 e o 5: [5, 6]

Mantemos o 4

Mesclamos o 'grupo' 5, 6 e o 4: [4, 5, 6]

- mergeSort([6, 5]) -> Desempilhado

Mantemos o 6

Mantemos o 5

Mesclamos o 6 e o 5: [5, 6]

- mergeSort([4, 3]) -> Desempilhado

Mantemos o 4

Mantemos o 3

Mesclamos o 4 e o 3: [3, 4]

- mergeSort([2, 1]) -> Desempilhado

Mantemos o 2

Mantemos o 1

Mesclamos o 2 e o 1: [1, 2]

#### 4. No quarto passo é o resultado

- Depois dos passos anteriores, o vetor completo é ordenado.

Vetor ordenado: [1, 2, 3, 4, 5, 6]