

## Trabalho 1 - Teste de Software

Lucas Cardoso Azevedo

## Classes de Equivalência

Variável	Classes de equivalência válidas	Classes de equivalência inválidas
Vetor de inteiros	Vetor de inteiros com tamanho T, tal que $2 \leq T \leq 20$	Vetor de não inteiros de tamanho T, tal que $T < 2$ ou $T > 20$
Comprimento	$2 \leq \text{Comprimento} \leq 20$	Comprimento $< 2$ e Comprimento $> 20$
Complexidade	"On", "On2" ou "Onlogn"	Qualquer outro
Algoritmo	$0 \leq \text{Algoritmo} \leq 7$	Algoritmo $< 0$ e Algoritmo $> 7$

## Testes Resultantes

#	Nome do Teste	Caso de Teste	Descrição do Teste
1	ParametrosValidos	Entrada: sort({1, 2, 3, 4, 5}, 5, ["On", "On2", "Onlogn"], [0,1,2,3,4,5,6,7]), Saída Esperada: 0	Testes com valores válidos para todos algoritmos
2	ComplexidadeInvalida	Entrada: sort({1, 2, 3, 4, 5}, 5, "teste", [0,1,2,3,4,5,6,7]), Saída Esperada: 1	Teste com argumento de complexidade inválida ("teste")
3	ComplexidadeInvalidaVazio	Entrada: sort({1, 2, 3, 4, 5}, 5, "", [0,1,2,3,4,5,6,7]), Saída Esperada: 1	Teste com argumento de complexidade vazio
4	ComplexidadeValidaErrada	Entrada: sort({1, 2, 3, 4, 5}, 5, ["On2", "Onlogn", "On"], [0,1,2,3,4,5,6,7]), Saída Esperada: 1	Testes com argumento de complexidade válidos, porém não correspondentes aos algoritmos
5	AlgoritmoInvalidoLimiteSuperior	Entrada: sort({1, 2, 3, 4, 5}, 5, "On", 9), Saída Esperada: 1	Teste com argumento de algoritmo inválido (9)
6	AlgoritmoInvalidoLimiteInferior	Entrada: sort({1, 2, 3, 4, 5}, 5, "On", -1), Saída Esperada: 1	Teste com argumento de algoritmo inválido (-1)
7	TamanhoMaiorQueVetor	Entrada: sort({1, 2, 3, 4, 5}, 6, ["On", "On2", "Onlogn"], [0,1,2,3,4,5,6,7]), Saída Esperada: 1	Testes com argumento de comprimento inválido (6). valgrind aponta erro de acesso a endereço
8	VetorValidoDesordenado	Entrada: sort({5, 4, 3, 2, 1}, 5, ["On", "On2", "Onlogn"], [0,1,2,3,4,5,6,7]), Saída Esperada: {1, 2, 3, 4, 5}	Testes de ordenamento dos algoritmos para 5 valores
9	TamanhoValidoLimiteInferior	Entrada: sort({2, 1}, 2, ["On", "On2", "Onlogn"], [0,1,2,3,4,5,6,7]), Saída Esperada: {1, 2}	Testes de ordenamento dos algoritmos para 2 valores
10	TamanhoValidoLimiteSuperior	Entrada: sort({19, 17, 15, 13, 11, 9, 7, 5, 3, 1, 20, 18, 16, 14, 12, 10, 8, 6, 4, 2}, 20, ["On", "On2", "Onlogn"], [0,1,2,3,4,5,6,7]), Saída Esperada: {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20}	Testes de ordenamento dos algoritmos para 20 valores
11	TamanhoInvalidoLimiteInferior	Entrada: sort({1}, 1, ["On", "On2", "Onlogn"], [0,1,2,3,4,5,6,7]), Saída Esperada: 1, {1}	Testes com apenas um valor, limite de tamanho inferior inválido
12	TamanhoInvalidoLimiteSuperior	Entrada: sort({21, 20, 19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1}, 21, ["On", "On2", "Onlogn"], [0,1,2,3,4,5,6,7]), Saída Esperada: 1, {21, 20, 19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1}	Testes com 21 valores, limite de tamanho superior inválido
13	VetorValoresIguais	Entrada: sort({9, 9, 9, 9, 9}, 5, ["On", "On2", "Onlogn"], [0,1,2,3,4,5,6,7]), Saída Esperada: 0, {9, 9, 9, 9, 9}	Testes com vetor de valores iguais
14	VetorValidoOrdenado	Entrada: sort({1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20}, 20, ["On", "On2", "Onlogn"], [0,1,2,3,4,5,6,7]), Saída Esperada: 0, {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20}	Testes com vetor de valores já ordenados, verifica mudanças inesperadas
15	VetorValidoDesordenadoMeio	Entrada: sort({13, 12, 11, 10, 9}, 20, ["On", "On2", "Onlogn"], [0,1,2,3,4,5,6,7]), Saída Esperada: 0, {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20}	Testes com vetor de 20 valores, contendo uma região desordenada apenas no meio (do índice 8 ao 12) e o restante ordenado. É passada somente a região desordenada à função sort e testado se o vetor inteiro continua ordenado.
16	VetorVazio	Entrada: sort({}, 0, ["On", "On2", "Onlogn"], [0,1,2,3,4,5,6,7]) Saída Esperada: 1	Testes com vetor vazio e tamanho 0. Ao tentar comparar vetores vazios no teste resulta em erro "Pointless".
17	VetorDeNegativos	Entrada: sort({-5, -4, -3, -2, -1}, 5, ["On", "On2", "Onlogn"], [0,1,2,3,4,5,6,7]) Saída Esperada: Erro (Segmentation fault)	Testes com vetor de inteiros negativos, resulta em segmentation fault. Teoricamente não deveria, pois são parâmetros válidos e não é informada tal limitação (somente valores positivos) na especificação.