UNIVERSIDADE VILA VELHA
ESPÍRITO SANTO

Nome do Aluno:	**Filipe Vieira Rocha**
Disciplina: Estrutura de Dados	**Professor: Gustavo Rocha**
Turma: CC4N	

Análise de Complexidade

1-Shellsort

Complexidade de Tempo:

Melhor caso: O(n log n) (quando os dados já estão quase ordenados e os gaps ajudam a reduzir os movimentos).

Caso médio: Depende da sequência de gaps utilizada. Para a sequência n/2, n/4,..1, o tempo médio é aproximadamente $O(n^{3/2})$

Pior caso: O(n2) (para uma sequência de gaps ineficiente e dados em ordem inversa).

Complexidade de Espaço:

Espaço adicional necessário:O(1), pois o algoritmo é in-place (não utiliza estrutura auxiliar).

2-Quicksort

Complexidade de Tempo:

Melhor caso: O(nlogn) (quando o pivô divide o array em partes quase iguais a cada iteração).

Caso médio: O(nlogn), considerando distribuições aleatórias de dados.

Pior caso: $O(n^2)$ (quando o pivô sempre divide o array de maneira desequilibrada, como em um array já ordenado).

Complexidade de Espaço:

Espaço adicional necessário: O(logn), devido à pilha de recursão.

3-Heapsort

Complexidade de Tempo:

Melhor caso: O(nlogn), pois a construção do heap e a extração de elementos têm essa complexidade.

Caso médio: O(nlogn), pois o algoritmo é independente da distribuição dos dados.

Pior caso: O(nlogn), pois não há variações extremas dependendo da entrada.

Complexidade de Espaço:

Espaço adicional necessário: O(1), pois o algoritmo é in-place.