Instituto Federal de Goiás - Campus Goiânia
Bacharelado em Sistemas de Informação
Estruturas de Dados II

**Nota**

Nome do Aluno: _____ Data: __/__/____

Prof. Renan Rodrigues de Oliveira

# Exercícios
# Métodos Sofisticados de Ordenação

1. Mostre um exemplo de entrada que demonstra que o *SellSort* não é estável.

2. Dada a sequência de números: 3 4 9 2 5 1 8. Ordene em ordem crescente utilizando o algoritmo *MergeSort*, apresentando a sequência dos números a cada passo (teste de mesa).

3. Mostre o resultado da operação *particao*(*v*, 0, 15) do *QuickSort* para o vetor a seguir. Utilize o elemento *v*[*inicio*] como sendo o pivô em cada chamada da função.

33 22 55 33 44 22 99 66 55 11 88 77 33 88 66 66.

4. Um vetor *a*[*p*..*r*] é bipartido se existe *i* em *p*..*r* tal que a[*p*..*i*-1] ≤ *a*[*i*] ≤ *a*[*i*+1..*r*]. Escreva um programa em C que decida se um dado vetor *a*[*p*..*r*] é bipartido. Em caso afirmativo, o seu programa deve devolver o índice *i* que caracteriza a bipartição.

5. Escreva uma função que decida se um vetor *v* é ou não um *heap*.

6. Use o algoritmo *HeapSort* para ordenar o vetor 16 15 14 13 12 11 10 9 8 7 6 5 4. Mostre o estado do vetor no início de cada iteração.

7. Escreva uma versão do *HeapSort* que rearranje um vetor *A*[1..*n*] em ordem decrescente.

8. O algoritmo *CountSorting* ordena vetores em tempo linear para o tamanho do vetor inicial, não realiza comparações e é estável. Pesquise e explique o funcionamento deste algoritmo, apresentando seus aspectos positivos e negativos. Por fim, apresente uma implementação em C deste algoritmo. Para simplificar a implementação, suponha que o tamanho e maior valor do vetor não é maior que 1000.