



Nome do Aluno: Luiz Antônio Rodrigues dos Santos

Data: 02/04/2018

Prof. Renan Rodrigues

Métodos Sofisticados de Ordenação

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

8	10	12	10	1	2	3	20	3	2
---	----	----	----	---	---	---	----	---	---

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 4 9 2 | 5 1 8 | | | | | |
| 3 4 | 9 2 | 5 1 | 0 |
| 3 | 4 | 9 | 2 | 5 | 1 | 0 |
| 3 4 | 2 9 | 1 5 | 0 |
| 2 3 4 9 | 0 1 5 |
| 0 1 2 3 4 5 9 |
```

3. Mostre o resultado da operação partição (v, 0, 15) do QuickSort para o vetor a seguir. Utilize o elemento v[início] 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.
33 22 55 33 44 22 66 66 55 11 33 77 88 88 66 99
11 22 55 33 44 22 66 66 55 33 33 77 88 88 66 99
11 22 55 33 44 22 66 66 55 33 33 77 88 88 66 99
11 22 55 33 44 22 33 33 55 66 66 77 88 88 66 99
11 22 22 33 44 55 33 33 55 66 66 77 88 88 66 99
11 22 22 33 44 55 33 33 55 66 66 77 88 88 66 99
11 22 22 33 33 33 44 55 55 66 66 77 88 88 66 99
11 22 22 33 33 33 44 55 55 66 66 77 88 88 66 99
11 22 22 33 33 33 44 55 55 66 66 77 88 88 66 99
11 22 22 33 33 33 44 55 55 66 66 77 66 88 88 99
11 22 22 33 33 33 44 55 55 66 66 66 77 88 88 99
11 22 22 33 33 33 44 55 55 66 66 66 77 88 88 99
```

4. Um vetor $a[p..r]$ é bipartido se existe i em $p..r$ tal que $a[p..i-1] \leq a[i] \leq a[i+1..r]$. 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. Código em anexo.

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

Código em anexo.

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.

Código em anexo.

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

Código em anexo.

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.]

Counting sort é um algoritmo de ordenação estável cuja complexidade é $O(n)$, as chaves podem tomar valores entre 0 e $M-1$. Se existirem k_0 chaves com valor 0, então ocupam as primeiras k_0 posições do vetor final: de 0 a k_0-1 .

A ideia básica do counting sort é determinar, para cada entrada x , o número de elementos menor que x . Essa informação pode ser usada para colocar o elemento x diretamente em sua posição no array de saída. Por exemplo, se há 17 elementos menor que x , então x pertence a posição 18. Esse esquema deve ser ligeiramente modificado quando houver vários elementos com o mesmo valor, uma vez que nós não queremos colocar eles na mesma posição.

Um ponto positivo é que tem complexidade linear, e um ponto negativo é precisar usar vetores auxiliares e ordenar somente números inteiros pelo fato de seus valores servirem como índices no vetor de contagem.

Código em anexo.