Exercícios sobre algoritmos elementares de ordenação

14 de novembro de 2013

Exercício 1

Qual é o vetor resultante após as 4 primeiras trocas ao executar ordenação por seleção com o seguinte array inicial?

```
72 12 62 69 27 67 41 56 33 74
```

Exercício 2

Implemente o algoritmo de ordenação por inserção visto em aula e conte o número total de cópias de valores do vetor dentro do **while** ao executar no seguinte array:

```
72 12 62 69 27 67 41 56 33 74
```

Exercício 3

Considere o seguinte algoritmo de ordenação:

```
1  void saltoSort(int vetor[], int n, int salto) {
2
3    int chave, i, j;
4
5    for (j = salto; j < n; j++) {
6      chave = vetor[j];
7      i = j - salto;
8
9      while (i >= 0 && vetor[i] > chave) {
10       vetor[i+salto] = vetor[i];
11       i = i - salto;
12     }
13
14     vetor[i+salto] = chave;
15   }
16 }
```

Quando utiliza-se o algoritmo **saltoSort** com **salto=k** diz-que o vetor está k-ordenado. Porquê? Modifique o **saltoSort** para contar o número de cópias de valores do **vetor** dentro do **while**.

Exercício 4

Em um vetor aleatório com 50 números inteiros, conte o número de cópias realizadas pelo **saltoSort** para torná-lo 1-ordenado.

Exercício 5

Em um vetor aleatório com 50 números inteiros, conte o número total de cópias realizadas pelo **saltoSort** para torná-lo 13-ordenado e depois 4-ordenado e depois 1-ordenado.

Exercício 6

Qual dos seguintes métodos utiliza mais cópias dentro do **while**? Justifique.

- apenas um **saltoSort** para obter um vetor 1-ordenado,
- fazer um vetor ficar 13-ordenado e depois 4-ordenado e depois 1-ordenado?

Exercício 7

A sequência de Knuth $\{1, 4, 13, 53, 213, \ldots\}$ é definida da seguinte forma:

$$k(0) = 1$$
$$k(i) = 3 \times k(i-1) + 1$$

Se um vetor tem tamanho n, produza a sequência de Knuth de até n e, do maior elemento produzido até o menor, obtenha a $K(i)$-ordenação do vetor. Esse algoritmo é denominado ShellSort.

Exercício 8

Escreva o array resultante após realizar a segunda etapa de ordenação do Shellsort usando o incrementos $3x + 1$ proposto por Knuth usando o seguinte array inicial:

```
36 39 88 32 95 83 81 18 14 57
```

Exercício 9

Realize a análise empírica do Shellsort para sequências de números aleatórios utilizando como modelo $a \times x^b$.

Exercício 10

Realize a análise empírica do algoritmo de ordenação por inserção para sequências de números aleatórios utilizando como modelo $a \times x^b$.

Exercício 11

Quais das afirmações a seguir sobre algoritmos de ordenação elementar são verdadeiras? Pode haver mais de uma, marcar todas. A menos que especificado, assumir que as implementações de ordenação são aquelas vistas em aula.

- O número de comparações para a ordenação por inserção ordenar um array de $N/2$ números 1 seguidos por $N/2$ zeros (e.g., 1 1 1 1 0 0 0 0) é aproximadamente 1/4 N^2
- A ordem de crescimento do número de comparações para o Shellsort (com incremento 3x+1 proposto por Knuth) ordenar um array de N chaves distintas é $N^{(3/2)}$
- Logo após a passada de 4-ordenação no Shellsort (com incremento 3x+1 proposto por Knuth), o array está 4-ordenado, 5-ordenado e 6-ordenado.
- A ordem de crescimento do número de comparações para a ordenação por inserção ordenar um array de N chaves da forma 1 0 3 2 5 4 7 6 9 8 é linear.
- O número de comparações para o Shellsort (com incremento 3x+1 proposto por Knuth) ordenar um array de N chaves distintas é N