



Universidade de São Paulo

Instituto de Ciências Matemáticas e de Computação

Departamento de Ciências de Computação

SCC0220 – Laboratório de Introdução à Ciência da Computação II
05 de Outubro de 2018

Operações na Ordenação

Descrição

Como se sabe, a eficiência dos diferentes algoritmos de ordenação depende do conjunto de entradas a serem ordenadas. É por essa razão que tais algoritmos são geralmente avaliados em seu melhor caso, pior caso e caso médio.

Uma grande empresa, que trabalha com dados de diferentes fontes, necessita frequentemente ordenar seus dados. Para identificar qual o melhor algoritmo para cada situação, essa empresa forneceu exemplos representativos de cada conjunto de dados (casos de teste) e deseja identificar quais algoritmos de ordenação apresentam melhor e pior eficiência em cada situação. Para calcular a eficiência, a empresa deseja que sejam calculados dois índices:

- C: número de comparações de chaves;
- M: número de movimentações de chaves.

Os valores de C e M se referem ao número de operações de comparação e movimentação que utilizam elementos do vetor de dados. Não se deve considerar as operações que envolvam apenas os índices das chaves.

Sua tarefa, caso decida aceitá-la, será calcular C e M para os algoritmos de ordenação que fore requisitados em cada caso de teste, identificando quais algoritmos apresentam maior e menor C e M.

Entrada

Cada caso de teste contém um conjunto de dados a ser ordenado e quais algoritmos de ordenação você deve usar para verificar a eficiência.

A primeira linha do caso teste irá conter dois números, O e N, separados por espaço. O número O ($1 \leq O \leq 5$) representa a quantidade de algoritmos de ordenação que deverão ser testados e N ($1 < N \leq 30000$) representa o tamanho do conjunto de dados a ser ordenado.

A segunda linha do caso de teste contém O strings, indicando os algoritmos de ordenação que devem ser utilizados. As possibilidades são: INSERTIONSORT, BUBBLESORT, MERGESORT, HEAPSORT, QUICKSORT. Para o caso do QUICKSORT, utilize a escolha do pivô como o elemento do meio do vetor.

Em seguida serão dadas N linhas, cada uma contendo um número inteiro entre 1 e 100000, representando os dados a serem ordenados.

Saída

Para cada caso de teste, a saída deve ser composta de quatro linhas, seguindo o formato:

Menor C: ALGORITMO

Maior C: ALGORITMO

Menor M: ALGORITMO

Maior M: ALGORITMO

A palavra "ALGORITMO" deve ser substituída pelo nome do algoritmo apropriado.

Exemplos de Entrada e Saída

Entrada

```
2 8↵
INSERTIONSORT BUBBLESORT↵
5↵
6↵
7↵
8↵
9↵
10↵
11↵
12↵
```

Saída

```
Menor C: INSERTIONSORT↵
Maior C: BUBBLESORT↵
Menor M: BUBBLESORT↵
Maior M: INSERTIONSORT↵
```

Obs: Ao final do trabalho, todos os casos de teste ficarão abertos no run.codes para quem quiser explorar os dados.