

Implementando Algoritmos de Ordenação

Lista03

Informações

- Entrega pelo Portal COMP em [<http://trab.dc.unifil.br/moodle/>](http://trab.dc.unifil.br/moodle/);

Para realizar o trabalho, siga o roteiro:

1. O método de ordenação por seleção consiste em buscar pelo menor valor do arranjo e trocá-lo de posição com o primeiro valor. Em seguida, encontre o segundo menor valor – ou o menor valor do arranjo a partir da segunda posição – e troque-o de posição com o segundo valor, e assim por diante, com todos os $n - 1$ valores do arranjo, onde n é o `arranjo.length`.

Com base nessa descrição, faça:

- (a) Implemente o método `ordenarPorSelecao` da classe `Ordenacoes`.
 - (b) No documento texto de descrição do projeto (onde há seu nome), responda às seguintes perguntas:
 - i. Por que só é necessário ordenar até o elemento $n - 1$?
 - ii. Quantas vezes cada laço `for` ou `while` é executado se o arranjo inicial estiver completamente desordenado?
 - iii. Quantas vezes cada laço `for` ou `while` é executado se o arranjo inicial já estiver corretamente ordenado?
2. O método de ordenação da bolha faz referência a bolhas de ar subindo em uma panela com água em ebulição. Isso pois o seu funcionamento é da seguinte forma:
 1. Compare o primeiro elemento com o segundo e se ele for maior faça a troca;
 2. Compare o segundo elemento com o terceiro e se ele for maior faça a troca;
 3. Repita este processo até o fim do arranjo;
 4. Repita os processos anteriores, porém somente até o penúltimo elemento do arranjo;
 5. Repita o processo anterior, porém somente até o antepenúltimo elemento do arranjo;
 6. Na realidade, repita o processo até que você percorra o arranjo realizando trocas até seu segundo elemento;
 7. A qualquer momento que o arranjo for percorrido e tiver sido realizada nenhuma troca, o arranjo já estará ordenado, portanto interrompa o processo.

Com base nessa descrição, faça:

- (a) Implemente o método `ordenarPorBolha` da classe `Ordenacoes`.
- (b) No documento texto de descrição do projeto (onde há seu nome), responda às seguintes perguntas:

- i. Quantas vezes cada laço `for` ou `while` é executado se o arranjo inicial estiver completamente desordenado?
 - ii. Quantas vezes cada laço `for` ou `while` é executado se o arranjo inicial já estiver corretamente ordenado?
 - iii. Qual método realiza mais trocas, bolha ou seleção? Por quê?
3. O método de ordenação por inserção se assemelha à maneira como organizamos as cartas de um baralho nas mãos. Imagine que você tenha recebido dez cartas escondidas (viradas para baixo), e vá adicionando uma por uma em sua mão, inserindo cada carta já ordenadamente. Este processo pode ser descrito por:
 1. Retire a primeira carta do monte e coloque na sua mão;
 2. Retire a segunda carta do monte, coloque-a no final de sua mão e compare com a primeira carta. Se for menor que ela, faça a troca entre elas;
 3. Retire a terceira carta do monte e coloque-a no final de sua mão. Compare-a com a segunda carta e se for menor, coloque a segunda carta no lugar da terceira. Neste caso, compare-a com a primeira carta, se for menor coloque a primeira carta na segunda posição e a carta nova na primeira posição, senão coloque a carta nova na segunda posição;
 4. Repita o processo até que todas as cartas do monte tenham sido inseridas.

Com base nessa descrição, faça:

- (a) Crie o método `public int[] ordenarPorInsercaoCom2Arranjos(int[] monteComprado)` na classe `Ordenacoes`, e implemente o método da inserção utilizando, além de `int[] monteComprado`, da assinatura do método, o arranjo `int[] mao`, que é retornado ordenado.
 - (b) Agora implemente o método `ordenarPorInsercao` da classe `Ordenacoes` utilizando apenas o arranjo da assinatura do método. Fique à vontade para criar quantas variáveis (não arranjos) forem necessárias.
 - (c) No documento texto de descrição do projeto (onde há seu nome), responda às seguintes perguntas:
 - i. Quantas vezes cada laço `for` ou `while` é executado se o arranjo inicial estiver completamente desordenado?
 - ii. Quantas vezes cada laço `for` ou `while` é executado se o arranjo inicial já estiver corretamente ordenado?
4. Não se esqueça de escrever os *Javadocs* para todas as classes e métodos criados. Não se esqueça também de testar os métodos com *JUnit*.