



Universidade Federal de Campina Grande

Centro de Engenharia Elétrica e Informática

Departamento de Sistemas e Computação

Graduação em Ciência da Computação

Variação de Bubblesort

Objetivo: Praticar a implementação de uma variação do bubble sort.

Um programador esperto propôs uma variação para o bubblesort chamada Odd Even Bubblesort. A variação consiste em simular uma divisão lógica do array usando índices pares e ímpares e aplicar dois bubblesorts em cada um deles. Assim, um bubblesort completo é aplicado considerando-se apenas os elementos indexados com os pares (mas trocando o elemento i com $i+1$) e depois um outro bubblesort é aplicado considerando-se apenas os elementos indexados com os ímpares (mas trocando os elementos i e $i+1$). Isso continua até que não haja mais trocas e o array esteja ordenado.

Uma ilustração de execução desse algoritmo é a seguinte:

[02, 05, 06, 04, 13, 03, 12, 19] – SEPARACAO CONSIDERANDO INDICES PARES E IMPARES
[02, 05, 06, 04, 12, 03, 13, 19] – BUBBLESORT APLICADO APENAS NOS ELEMENTOS EM POSICOES PARES
[02, 05, 06, 04, 12, 03, 13, 19]
[02, 05, 04, 06, 12, 03, 13, 19] – HOUVE TROCA
[02, 05, 04, 06, 12, 03, 13, 19]
[02, 05, 04, 06, 03, 12, 13, 19] – HOUVE TROCA
[02, 05, 04, 06, 03, 12, 13, 19]
[02, 05, 04, 06, 03, 12, 13, 19] – BUBBLESORT APLICADO APENAS NOS ELEMENTOS EM POSICOES IMPARES
[02, 05, 04, 06, 03, 12, 13, 19]
[02, 04, 05, 06, 03, 12, 13, 19] – HOUVE TROCA
[02, 04, 05, 06, 03, 12, 13, 19]
[02, 04, 05, 03, 06, 12, 13, 19] – HOUVE TROCA
[02, 04, 05, 03, 06, 12, 13, 19]
[02, 04, 05, 03, 06, 12, 13, 19] – HOUVE TROCA ENTÃO O PROCESSO SE REPETE
[02, 04, 05, 03, 06, 12, 13, 19]
[02, 04, 05, 03, 06, 12, 13, 19]
[02, 04, 03, 05, 06, 12, 13, 19] – HOUVE TROCA
[02, 04, 03, 05, 06, 12, 13, 19]
[02, 04, 03, 05, 06, 12, 13, 19]
[02, 04, 03, 05, 06, 12, 13, 19]
[02, 04, 03, 05, 06, 12, 13, 19]
[02, 03, 04, 05, 06, 12, 13, 19]
[02, 03, 04, 05, 06, 12, 13, 19]
[02, 03, 04, 05, 06, 12, 13, 19] – HOUVE TROCA ENTÃO O PROCESSO SE REPETE
[02, 03, 04, 05, 06, 12, 13, 19]
[02, 03, 04, 05, 06, 12, 13, 19]
[02, 03, 04, 05, 06, 12, 13, 19] – HOUVE TROCA ENTÃO O PROCESSO SE REPETE
...

Como não vai haver mais trocas porque o array está ordenado o algoritmo para.

Atividades necessárias antes de iniciar o exercício:

1. Crie um projeto no Eclipse chamado LEDA, por exemplo (pode ser qualquer outro nome que lhe convier);
2. Descompacte o arquivo baixado (exceto o PDF) na pasta dos fontes (normalmente **src**) do seu projeto LEDA criado no seu workspace. O arquivo baixado tem a seguinte estrutura:
 - sorting
 - Sorting.java (INTERFACE)
 - Util.java (classe auxiliar contendo o método de swap a ser usada na implementação)
 - variationsOfBubblesort
 - OddEvenBubblesort.java (**IMPLEMENTACAO A SER PREENCHIDA PELO ALUNO**)
3. No Eclipse, selecione a pasta dos fontes no projeto LEDA e faça um refresh (apertar F5). Note que deve aparecer as pastas e arquivos acima.

Agora voce está pronto para começar a trabalhar nas seguintes atividades:

1. Observe a interface Sorting.java. Ela contém a assinatura do método do a ser implementado.
2. Observe também a existência implementação incompleta OddEvenBubblesort.java. Voce precisa implementar o método de acordo com a explicação.
3. *Sua implementação deve funcionar corretamente independentemente do array a ser ordenado!*

Instruções para o envio

Ao terminar o exercício, faça os seguintes passos:

1. Compacte a pasta **sorting** que existe nos fontes de seu projeto LEDA (**src**) e retire a classe MainTest.java desse arquivo compactado. A compactação DEVE ser feita a partir do diretório raiz de seus fontes de forma a preservar a estrutura de pastas que refletem a estrutura dos pacotes (package) Java. Por exemplo, voce deve ter um arquivo compactado NOME_COMPLETO_DO_ALUNO.ZIP com a seguinte estrutura:
 - sorting
 - Sorting.java
 - Util.java
 - variationsOfBubblesort
 - OddEvenBubblesort .java
2. Envie esse arquivo com sua solução para o sistema de submissão e verifique que o contador de submissões será alterado.

Observações finais:

- **A interpretação do exercício faz parte da atividade.**
- **A atividade é individual. A conversa entre alunos é proibida.**
- **É proibido coletar códigos prontos e adaptar. Implemente as questões. Isso é para seu aprendizado.**
- **Caso voce observe qualquer problema no sistema de submissão, contacte o professor imediatamente.**
- **Se voce nao compactar o arquivo seguindo a estrutura de diretórios a compilação não terá sucesso e o sistema mostrará isso. Erro de compactação serão de responsabilidade do aluno. O professor não ajudará o aluno nesse item. É só seguir as instruções deste arquivo.**