



Universidade Federal de Campina Grande

Centro de Engenharia Elétrica e Informática

Departamento de Sistemas e Computação

Graduação em Ciência da Computação

Exercício sobre Combsort

Objetivo: Praticar a implementação do combsort.

Relembre o algoritmo do combsort visto em sala de aula, com a seguinte estratégia:

- A distancia começa por $gap = tamanho/fator$ ($fator=1.25$)
- A entrada é ordenada com as trocas considerando elementos distanciados por gap
- gap é atualizado ($gap = gap/fator$) até 1
- Quando $gap=1$ combo sort continua até o array estar todo ordenado

Uma ilustração de execução desse algoritmo é a seguinte:

[02, 05, 06, 04, 13] – $gap = 5/1.25 = 4$, $i = 0$, compara array[i] com array[i+gap]

[02, 05, 06, 04, 13] – $gap = 4/1.25 = 3$, $i = 0$, compara array[i] com array[i+gap]

[02, 05, 06, 04, 13] – $i = 1$, compara array[i] com array[i+gap]

[02, 05, 06, 04, 13] – $gap = 3/1.25 = 2$, $i = 0$, compara array[i] com array[i+gap]

[02, 05, 06, 04, 13] – $i = 1$, compara array[i] com array[i+gap]

[02, 04, 06, 05, 13] – $i = 1$, inverte array[i] com array[i+gap]

[02, 05, 06, 04, 13] – $i = 2$, compara array[i] com array[i+gap]

[02, 05, 06, 04, 13] – $gap = 2/1.25 = 1$, $i = 0$, compara array[i] com array[i+gap] (se comporta como bubblesort)

[02, 04, 05, 06, 13] – $gap = 2/1.25 = 1$, $i = 0$, compara array[i] com array[i+gap] (se comporta como bubblesort)

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 CONTENDO A ASSINATURA DO METODO DE ORDENAÇÃO)
 - SortingImpl.java (IMPLEMENTAÇÃO BASE A SER HERDADA POR TODAS AS IMPLEMENTAÇÕES)
 - Util.java (CLASSE AUXILIAR CONTENDO O METODO DE SWAP A SER USADO NAS IMPLEMENTACOES)
 - variationsOfBubblesort
 - Combsort.java (IMPLEMENTAÇÃO 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. 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 (INTERFACE CONTENDO A ASSINATURA DO METODO DE ORDENAÇÃO)

--SortingImpl.java (IMPLEMENTAÇÃO BASE A SER HERDADA POR TODAS AS IMPLEMENTAÇÕES)

-- Util.java (CLASSE AUXILIAR CONTENDO O METODO DE SWAP A SER USADO NAS IMPLEMENTACOES)--
variationsOfBubblesort

--- Combsort.java

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.**