Universidade Federal de Campina Grande
Centro de Engenharia Eletrica e Informática
Departamento de Sistemas e Computação
Graduação em Ciência da Computação

Exercício sobre heap binária

Objetivo: Praticar a implementação de heap binária**.**

Relembre o conceito de heap binária visto em sala de aula.

1. O arquivo baixado tem a seguinte estrutura:
   - adt
   -- heap
   --- GenericHeap.java (INTERFACE DE UMA HEAP GENÉRICA)
   --- MinHeap.java (INTERFACE DE UMA MIX HEAP)
   --- MinHeapImpl.java (IMPLEMENTACAO PARCIAL DE UMA MIN HEAP)
   --- Util.java (CLASSE UTILITÁRIA)
2. No Eclipse, selecione a pasta dos fontes no projeto LEDA e faça um refresh (apertar F5). Note que deve aparecer um pacote adt.heap contendo or arquivos mencionados acima.

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

1. Observe a interface GenericHeap.java. Ela descreve os serviços de uma Heap genérica.
2. Observe a interface MinHeap.java. Ela descreve os serviços de umaMin Heap.
3. Observe também a existência implementação incompleta MinHeapImpl. Voce precisa implementar os métodos incompletos. Note que sua implementacao parcial de heap ja contem dis atributos: INITIAL_SIZE (o temanho inicial da heap) e INCREASING_FACTOR (fator de crescimento). Sua heap comeca com o tamanho inicial. Quando ela atinge o tamanho maximo entao ela aumenta de tamanho somando-se INCREASING_FACTOR ao tamanho atual da heap. Isso acontece toda vez que a heap enche.
4. Note que na sala de aula foi mostrado a max-heap. Modificar para uma min-heap significa apenas inverter a operação de comparação (elementos menores vão sendo empurrados para “cima” da heap ao invés dos maiores). Em termos de código isso é uma alteração simples nas operações de comparação entre valores da heap (em todos os métodos). Apenas o método heapsort é o que sofre mais alterações. Na abordagem vista em sala o maximo da heap é sempre trocado com o ultimo, a heap diminui de tamanho e o heapify é aplicado a partir da raiz. Isso se repete até que a heap tenha apenas 1 elemento. Já na min-heap, o heapsort tem a seguinte estrutura (em pseudo-código):

```
Heapsort(A)
  result = array with size length(A)
int index = 0
BuildHeap(A)
  while(!isEmpty()){
    result[index++] = extractRootElement()
  }
}
//ao final o array result estáordenado com oselementos da heap
```

5. Concentre-se em implementar conforme descrito na interface e pense em cenários para testar suas implementações. Alguns cenários interessantes são: testar insercoes e verificar se o tamanho da heap muda e se o elemento inserido esta na posicao correta da heap. O memso com a remocao. Nos testes do heapsort, voce pode inclusive adaptar os testes usados pelos algoritmos de ordenacao para testar o heapsort. Procure testar todos os metodos da heap. Voce pode até testar sua heap comparando ela com uma implementação pronta de Java.

Instruções para o envio

Ao terminar o exercício, faça os seguintes passos:

1. Compacte o roteiro por meio do Maven e envie o mesmo utilizando o link disponibilizado na página.

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.**