

Nome: _____

Prova 3º Bimestre**Informações**

- Quantidade de questões: **3**
- Total de pontos: **100**
- Peso no bimestre: **30%**

Instruções

1. Antes de iniciar, confira se esta cópia contém todas as questões e se estão legíveis.
2. Prova individual, não é permitida troca de quaisquer informações em quaisquer meios de comunicação com outras pessoas.
3. Prova com consulta. Permitido utilizar livros, seus materiais pessoais, trabalhos feitos e entregues, e consultas à Internet. Não é permitido buscar a solução de qualquer problema. Se for pedido para explicar algo e o aluno não souber, sua resposta será anulada e receberá nota zero.
4. A correta interpretação das questões faz parte da avaliação.
5. Esta é uma prova prática, todos os exercícios deverão ser feitos no projeto do BlueJ, disponibilizado pelo Portal de Entregas.
6. Não esqueça de colocar seu nome no arquivo texto do projeto.
7. Ao final, lembre-se de entregar a prova no Portal de Entregas.

Boa Prova!**Tabela de Pontuações – Uso exclusivo do professor!**

Questões	1	2	3	Total
Valor	5	60	35	100
Pontuação				

Para a realização desta prova, crie um novo projeto no BlueJ denominado **Prova3**, e faça todas as atividades deste roteiro dentro do projeto. Lembre-se de adicionar as classes de *unit tests*, distribuídas junto com este roteiro, para verificar a correção de suas implementações.

Questões

1. [5 pontos] Leia o conteúdo previsto pelo Plano de Ensino da disciplina de Laboratório de Computação II, na seção Programa. Transcreva, no documento de descrição de projeto do BlueJ, aqueles conceitos que já foram abordados na nossa disciplina.
2. O quadro abaixo descreve a TAD lista inversamente ordenada:

Operação	Descrição	Tempo
inserir(<i>v</i>)	Insere o elemento <i>v</i> inversamente ordenado.	$O(n)$
buscar(<i>v</i>)	Busca pela posição de <i>v</i> na lista.	$O(\log n)$
removerEm(<i>i</i>)	Remove e retorna o elemento na posição <i>i</i> .	$O(n)$
recuperarEm(<i>i</i>)	Retorna o elemento na posição <i>i</i> .	$O(1)$
numElementos()	Retorna o número de elementos contidos na lista.	$O(1)$

- (a) [40 pontos] Escreva a classe **ListaOrdenadaInv**, que implementa essa TAD. Por simplicidade, considere que esta classe seja capaz de manipular apenas objetos do tipo **Comparable**.
 - (b) [20 pontos] Substitua a utilização restrita de **Comparable** para quaisquer tipos de objetos com *generics*.
3. Escreva os seguintes métodos na classe **UtilitariosHeap**:
 - (a) [35 pontos] **verificarHeap**: Recebe um arranjo de **Comparable**, verifica se ele possui propriedade de *heap*, retornando **true** se for o caso, ou **false** caso não seja completamente *heap*.

Considere que os arranjos recebidos pelo método estão completamente preenchidos, exceto pela primeira posição, que deverá ser ignorada.