CI1056
2º semestre
Prova 3
28/11/2019
Limite de tempo: 100 minutos

Nome: Vinicius T. J. Date
GRR: _____
Prof.: _____

Profs. Castilho & Grégio

É PROIBIDO UTILIZAR QUALQUER MEIO PARA CONSULTA QUE NÃO SUA PRÓPRIA MEMÓRIA.

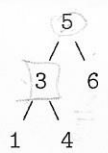
- A interpretação das questões faz parte da prova.
- Serão aceitos algoritmos escritos em pseudo código, Pascal ou C.
- As respostas podem ser escritas usando lápis ou caneta.
- Evite rasuras e escreva de forma legível, para que o professor consiga entender e corrigir adequadamente.
- O nome do aluno deve constar em todas as folhas de resposta.

---

1. (40 pontos) Sobre *Heaps*, responda as questões a seguir.
   (a) (9 pontos) Implemente as funções `PAI`, `ESQUERDO` e `DIREITO`, com os devidos argumentos de entrada e tipo de dado de saída, se houver.
   (b) (11 pontos) Implemente um algoritmo que construa uma `MIN HEAP`. Assuma que quaisquer funções que precisem ser utilizadas dentro do seu algoritmo já estão prontas (isto é, apenas chame as corretamente).
   (c) (5 pontos) O vetor V = {23, 17, 14, 6, 13, 10} se enquadra em algum dos tipos existentes de `heap`? Se sim, qual?
   (d) (15 pontos) Aplique o algoritmo que você fez no item 2 para construir uma `MIN HEAP` sobre o vetor V do item 3.

2. (10 pontos) O `Bucket Sort` é um algoritmo de ordenação cujo tempo de execução médio é $O(n)$. Tal custo é alcançado devido ao algoritmo assumir que a entrada foi gerada por um processo aleatório que distribui uniformemente os elementos em um intervalo $[0, 1)$. Esse intervalo é dividido em n subintervalos, também chamados de *buckets*. Mostre a saída da aplicação do `Bucket Sort` no vetor V = {.94, .12, .75, .7, .65, .5, .58, .24, .25, .34}, com $n = 5$, ilustrando o vetor e as listas do algoritmo após a ordenação.

3. (50 pontos) Sobre árvores binárias de busca (BST), responda as questões a seguir.
   (a) (10 pontos) Mostre a `struct` que representa um nó de uma BST capaz de guardar um número inteiro;
   (b) (15 pontos) Escreva uma função **recursiva** para imprimir a BST da figura em ordem crescente. A entrada dessa função é o *nó raiz* da árvore.



   (c) (15 pontos) Escreva uma função que insere elementos em uma BST, observando suas propriedades.
   (d) (10 pontos) Use a função que você criou no item anterior para inserir o nó de valor 2 na BST da figura e mostre as chamadas feitas durante a execução da função e o desenho final da árvore com o elemento inserido.