



## ***Prova 2 – Árvores e Grafos***

### **Instruções:**

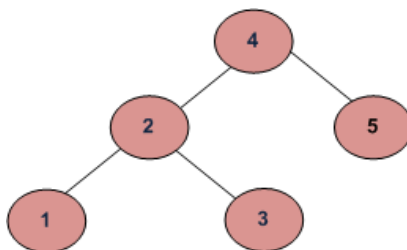
1. Para resolver as questões da prova, considere as implementações das bibliotecas de *lista*, *pilha*, *fila*, *arvore* e *grafo* apresentadas e discutidas em aula durante o semestre corrente.
2. **Provas e/ou questões idênticas (ou com indícios de cópia) terão a nota zerada.**
3. Ao término da prova, compacte a implementação de cada questão em um único arquivo (*com o seu nome completo*) e submeta no moodle na atividade denominada “**Prova 2 – Arvores e Grafos**”.

---

**Questão 1 - (2,5 pontos)** Dada uma árvore binária de busca, escreva uma função que verifica se a árvore contém um *fim de linha* (*dead end*).

*Fim de linha* significa que existe uma folha a partir da qual não conseguimos inserir novos valores.

No exemplo da árvore abaixo, não podemos colocar outros nós como filhos da folha 3.



Os parâmetros **min** e **max** indicam o intervalo de valores permitidos para o nó corrente. Inicialmente, estes parâmetros abrangem o intervalo todo.

**Dica:** Percorra a árvore recursivamente e quando os valores de **min** e **max** coincidirem, significa que não podemos adicionar outro nó naquele ramo, ou seja, encontramos um *fim de linha*.

```
int deadEnd (pDArvore arvore, void *min, void *max, FuncaoComparacao pfc)
```

---

---

**Questão 2 - (2,5 pontos)** Implemente uma função que verifica se uma dada árvore binária é **perfeita**.

Uma árvore binária **perfeita** é uma árvore binária *totalmente preenchida* (i.e., todos os nós, exceto as folhas, tem 2 filhos) em que todos os nós-folha tem a mesma profundidade.

**int** ehPerfeita ( **pDArvore** arvore)

---

**Questão 3 - (2,5 pontos)** Um grafo H é um **subgrafo** de um grafo G se  $V_H \subseteq V_G$  e  $A_H \subseteq A_G$ , onde V é o conjunto de vértices e A é o conjunto de arestas.

Implemente uma função que determina se um dado grafo H é subgrafo de G.

**int** subGrafo ( **pDGrafo** G, **pDGrafo** H)

---

**Questão 4 - (2,5 pontos)** Um grafo G é dito ser **regular** quando todos os seus vértices tem o mesmo grau.

Implemente uma função que determina se um dado grafo G é regular.

**int** regular ( **pDGrafo** grafoG)