

Tempo: 1:40 h

[2pt] 1. Para cada afirmação, indique se é VERDADEIRA (V) ou FALSA (F). Em seguida, justifique sua resposta de forma clara e concisa

- ☒ a. A estrutura de dados **BinaryHeap** representa implicitamente uma árvore binária completa usando um **array**, e a propriedade de **heap** garante que o menor valor na fila de prioridade é sempre armazenado na posição 0 (a raiz).
- ☒ b. O algoritmo de ordenação **merge-sort** é um exemplo clássico de divisão e conquista que, após dividir o **array** em duas metades e ordená-las recursivamente, mescla as metades ordenadas diretamente no **array** original sem a necessidade de um **array** auxiliar significativo, otimizando o uso de espaço.
- ☒ c. Teoremas fundamentais em algoritmos de ordenação baseados em comparações indicam que qualquer algoritmo desse tipo pode, no pior caso, ordenar  $n$  elementos distintos em um tempo de comparações inferior a  $O(n \log n)$ .
- ☒ d. Ao contrário de estruturas como Treaps e Skiplists, as árvores rubro-negras se destacam por oferecerem garantias de tempo de pior caso de  $O(\log n)$  para todas as operações  $\text{add}(x)$  e  $\text{remove}(x)$ , o que as torna uma escolha superior em aplicações onde a previsibilidade do desempenho é crucial.

[1,5pt] 2. Responda as seguintes questões:

- ☒ a. Considere uma árvore binária com 255 nós. Qual a altura máxima e mínima dessa árvore?
- ☒ b. Explique como uma Árvore Scapegoat decide qual nó será o "bode expiatório" para a reconstrução.
- ☒ c. Considere uma árvore Scapegoat. O que acontece quando colocamos o valor de  $\alpha$  próximo de 0,5? e de 1? Ele poderia ser 0?

[1pt] 3. Defina uma **heap** e uma **árvore binária de busca**. Quais são as principais diferenças conceituais e estruturais que distinguem uma da outra,

[1pt] 4. Considere o seguinte algoritmo para cálculo do número de nós de uma árvore binária.

```
tamanho(u)
{
    if (u = nil) return 0;
    return X
}
```

Sabendo que  $u$  representa um nó da árvore, e  $\text{nil}$  indica uma referência nula, qual deve ser o valor de  $X$  (isto é, quais comandos devem ser escritos no lugar de  $X$ ) para que o algoritmo funcione corretamente e retorne o número total de nós da árvore enraizada em  $u$ ?

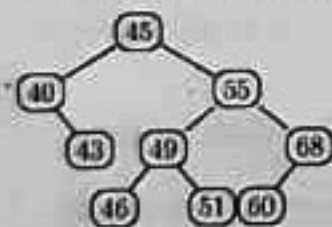
[1,5pt] 5. Escreva um algoritmo que encontre o maior valor de uma árvore binária.

[1pt] 6. Considere os seguintes nós, definidos por pares (**valor, prioridade**), sendo inseridos na ordem dada em uma **treap com heap mínima**:

(35,10), (50,8), (15,6), (30,18), (40,4), (45,2), (32,12), (42,11)

- 1. Desenhe a Treap resultante.
- 2. Desenhe a Treap após a retirada do nó (40,4)

[1pt] 7. Dada a seguinte árvore, forneça o resultado do seu percurso em pré-ordem, em-ordem, e pós-ordem e em profundidade.



[1pt] 8. As árvores **rubro-negras** são uma simulação eficiente das árvores 2-4. Explique como um nó em uma árvore 2-4 com dois, três ou quatro filhos é representado em uma árvore rubro-negra em termos de cor e estrutura. Além disso, descreva como a propriedade de altura preta em árvores rubro-negras se relaciona com a propriedade de altura em árvores 2-4

Obs.: Qualquer tentativa de desonestidade intelectual, conhecida popularmente como cola, será considerada falta grave e punida com nota zero na prova.