

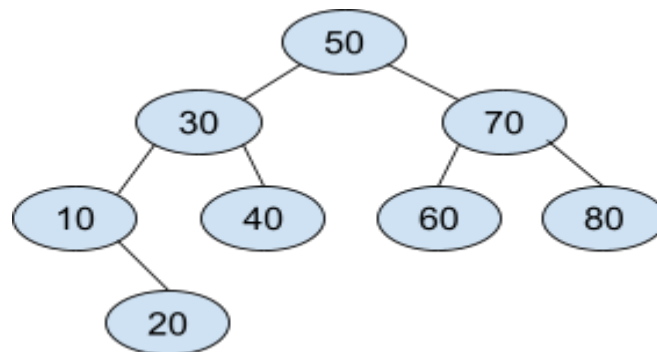
1. Estruturas de Dados (8 valores)

MinHeap Considere uma min-heap com 12 elementos armazenada no seguinte array.

0	1	2	3	4	5	6	7	8	9	10	11
5	15	8	21	16	12	35	22	23	34	42	13

1. Qual a posição no array do valor 42 após a remoção (na min-heap) dos dois menores elementos?
2. Qual o valor do array na posição 3 se o elemento da min-heap na posição 8 (23) for alterado para 7?

AVL Considere a seguinte AVL com 8 nodos e altura 4.



1. Depois de inserir nesta AVL os elementos 25 e 45 (por esta ordem), qual a altura da árvore resultante?
2. Das seguintes sequências de inserções na árvore, qual ou quais provocam uma mudança do nodo raiz da árvore? **A**: 75, 65, 55 ou **B**: 5, 45, 15?

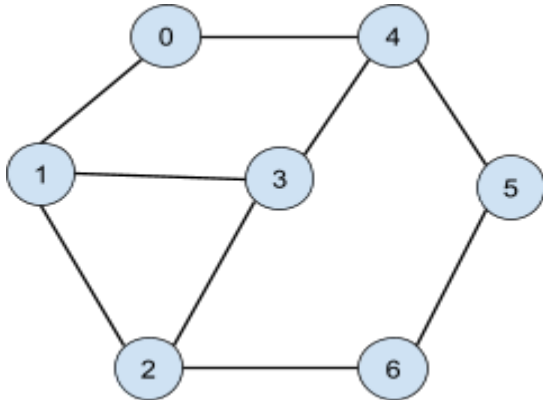
THash Considere a seguinte tabela de hash de inteiros, implementada com open-addressing e linear probing num array com 10 elementos, onde a função de hash é o resto da divisão inteira ($\text{hash}(x) = x \% 10$) e onde o estado de cada célula é **Free**, **Deleted** ou **Used**.

0	1	2	3	4	5	6	7	8	9
U 20	F 74	U 42	U 23	D 14	U 52	U 36	F 17	F 34	F 80

1. Em que posição seria feita a inserção do elemento 75?

2. A consulta do elemento 74 implica o acesso a quantos elementos do array?

Grafos Seja G o seguinte grafo não pesado e não orientado representado usando listas de adjacência ordenadas **por ordem crescente** do destino. (*Let G be the following unweighted and undirected graph represented using adjacency lists sorted in ascending order of the destination.*)



1. Numa travessia Depth-First iniciada no vértice 0 qual o último vértice a ser visitado?
2. Numa travessia Breadth-First iniciada no vértice 3 qual o último vértice a ser visitado?

2. Tabelas de Hash (4 valores)

Numa tabela de hash implementada usando open address e linear probing, o probe-count de cada chave define-se como o número de células que foram consultadas para inserir essa chave.

```
#define HSize ...  
#typedef struct bucket {int status; int value} THash [HSize];
```

Defina a função `int max_probe (THash h)` que calcula o maior probe-count associado às chaves armazenadas na tabela `h`. Assuma que existe uma função `int hash (int k)` que calcula o hash de uma chave `k`.

3. Min-heap (4 valores)

Defina uma função `int toArray (ABin a, int h[], int N)` que, dada uma árvore binária de inteiros que satisfaz as propriedades das min-heaps a converte para um array (escrevendo no máximo `N` elementos).

4. Grafos (4 valores)

Defina a função `int componentesOK (GrafoL g, int c[])` que, dado um grafo não orientado e um array com a identificação, para cada vértice, da componente ligada a que pertence, testa se esse array está bem construído. Por outras palavras, testa se para todos os vértices i e j , $c[i] == c[j]$ sse há um caminho de i para j . Use para isso (se for esse o caso) as funções apresentadas nas aulas (DF, BF, Dijkstra, ou FW). Qual a complexidade da função apresentada?