Algoritmos e Complexidade — Universidade do Minho
Exame – 16 de Janeiro de 2025

1. Correcção (3 valores)

Considere a definição da seguinte função que *comprime* um array somando os seus elementos dois a dois.

```
int soma2a2 (int v[], int N){
    // PRE: N > 0
    int r=1, w=0;
    w = 0; r = 1;
    while (r<N) {
        // INV: ??? Var: ???
        v[w] = v[r]+v[r-1];
        w+=1; r+=2;
    }
    if (N%2 == 1) v[w++] = v[r-1];
    // POS: w == (N+1) / 2
    return w;
    }
```

Apresente um invariante e um variante que lhe permitam provar a correção total da função. Note que a pós-condição apresentada **não** é a pós-condição do ciclo. (*Sugestão: comece por escrever a pós condição apropriada do ciclo*).

2. Complexidade (3 valores)

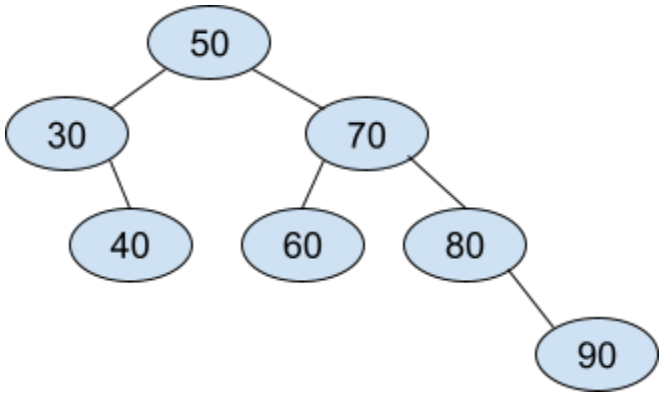
1. Analise a complexidade da função **`soma2a2`** (da questão 1) em termos do número de acessos de leitura e escrita no array.
2. Considere agora a definição da função `soma` que usa a função da questão anterior.

```
int soma (int v[], int N){
    int t;
    if (N == 1) return v[0];
    t = soma2a2 (v,N);
    return soma (v,t);
}
```

   Apresente e resolva uma relação de recorrência que traduza a complexidade da função `soma` (mais uma vez tendo em conta apenas o número de acessos array).

3. Estruturas de Dados (6 valores)

AVL Considere a árvore AVL abaixo.



1. Qual o valor da raiz da árvore depois de serem inseridas as chaves 65, 85 e 20 (por esta ordem)?
   Resposta:

2. Qual o factor de balanço (`Esq`, `Bal` ou `Dir`) da raíz da árvore depois dessas inserções.
   Resposta:

THash Considere a seguinte tabela de hash de inteiros, implementada com open-addressing e linear probing num array com 10 elementos- hash é o resto da divisão inteira (hash (x) = x % 10) e o estado de cada célula é **F**ree, **D**eleted ou **U**sed.

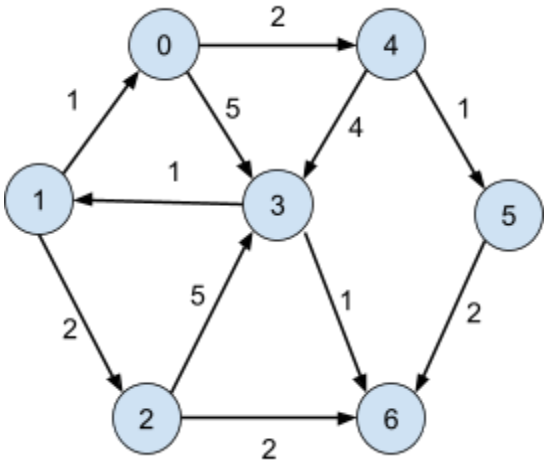
0	1	2	3	4	5	6	7	8	9
D 10	**U** 70	**D** 32	**D** 12	**U** 54	**U** 43	**F** 65	**D** 17	**U** 27	**U** 38

1. Após uma *garbage collection* em que posição ficará a chave 43?
   Resposta*:*

2. Após essa operação de garbage collection foi adicionada uma chave (aleatória) entre 100 e 199 (inclusivé). Em média, quantas posições do array consulta essa inserção?
   Resposta:

3. Após a operação de *garbage collection* referida na alínea 1. foram adicionadas as chaves 47 e 18 (por esta ordem). Em que posição fica armazenada a chave 47?
   Resposta:

Grafos Seja `g` o seguinte grafo pesado e orientado representado usando listas de adjacência ordenadas **por ordem crescente** do destino.



1. Numa travessia Depth-First iniciada no vértice 4 qual o último vértice a ser visitado?
   Resposta:

2. Após a execução de `dijkstraSP (g, 0, pais, pesos)` qual o valor de `pais[6]`?
   Resposta:

4. Min-heap (4 valores)

1. Defina uma função **`ABin`** `fromArray (int h[], int N)` que constrói a árvore binária correspondente à min-heap armazenada num array `h` com `N` elementos.
2. Analise a complexidade da definição apresentada.

5. Grafos (4 valores)

1. Defina uma função **`int`** `middle_point (GrafoL g, int a, int b)` que, dado um grafo (**não pesado e orientado**) e dois vértices, calcula o vértice `v` que minimiza a soma das distâncias de `a` a `v` e de `b` a `v`, A função retorna -1 se não existir tal vértice. Use se precisar as funções estudadas nas aulas TP. Para as funções que usar, comece por explicitar o seu tipo.
2. Qual a complexidade da função apresentada?