

**No final deverás confirmar que submetes corretamente no SIGEX o código fonte dos teus programas utilizando o nome indicado no enunciado. Quaisquer cópias detetadas serão penalizadas com anulação da prova.**

- 1 Tendo por base as bibliotecas de estruturas de dados apresentadas em Programação 2, implementa as funcionalidades pedidas nas duas alíneas seguintes no ficheiro **prob1.c**. Sempre que conveniente utiliza as funções disponíveis nas estruturas árvore AVL, grafo, e vetor.

- 1.1 [20 pontos] Implementa a função `avl_contafilhosunicos` para uma **árvore AVL** (definida pelo nó raiz) que devolve o número de nós da árvore que têm um e apenas um filho.

```
int avl_contafilhosunicos(no_avl *no)
```

O parâmetro da função é o apontador para o nó raiz da árvore e o retorno é o número de nós da árvore que têm um e apenas um filho.

Indica ainda num comentário no início do código da função qual a complexidade do algoritmo que implementaste (não é necessário justificar).

Depois de implementada a função, o programa deverá apresentar:

```
Numero de filhos unicos: 18
```

- 1.2 [20 pontos] Implementa a função `grafo_distanciamedia` que determina qual a distância média entre dois conjuntos de vértices de um **grafo**. Isto é, a média da distância entre cada vértice da região de origem e cada vértice da região de destino.

```
float grafo_distanciamedia(grafo *g, vetor *regiao_origem, vetor  
                           *regiao_destino)
```

O primeiro parâmetro da função é o apontador para grafo, o segundo é apontador para vetor com os índices dos vértices de origem, e o terceiro é apontador para o vetor com os índices dos vértices de destino. Para calcular a distância entre quaisquer dois vértices, considera a distância mais curta entre esses vértices. A função deve retornar a distância média ou -1 se não for bem-sucedida.

Depois de implementada a função, o programa deverá apresentar:

```
Distancia media entre regioes: 3.25  
Distancia media entre regioes: 1.50
```

**\*\*\* Submeta o ficheiro prob1.c no SIGEX \*\*\***

- 2 Tendo por base as bibliotecas de estruturas de dados apresentadas em Programação 2, implementa as funcionalidades pedidas nas duas alíneas seguintes no ficheiro **prob2.c**. Sempre que conveniente utiliza as funções disponíveis nas estruturas tabela de dispersão e vetor.

- 2.1 [25 pontos] Implementa a função `mais_frequente` que determina qual a string mais frequente num vetor. As strings guardadas no vetor estão no formato `Cidade,Região`. Pretende-se neste caso determinar a região mais frequente utilizando uma tabela de dispersão. Nota que a implementação da tabela de dispersão foi alterada para guardar valores inteiros (a chave mantém-se como string).

```
int mais_frequente(vetor *v, char *maisfreq)
```

O primeiro parâmetro da função é o vetor contendo as strings e o segundo parâmetro deverá ser usado para guardar a região mais frequente. A função deve retornar 0 se for bem sucedida, ou -1 em caso contrário.

Depois de implementada a função, o programa deverá apresentar:

```
Regiao mais frequente: bay of dragons
```

- 2.2 [15 pontos] Indica a complexidade da solução implementada na alínea anterior e uma justificação clara e sucinta (máximo 50 palavras) no comentário assinalado para o problema 2.2 no ficheiro `prob2.c`. Para complementar a resposta poderás incluir comentários no problema 2.1 indicando a complexidade das respetivas instruções.

**\*\*\* Submeta o ficheiro prob2.c no SIGEX \*\*\***