

**3. [ 7 valores]**

Considere a rede de aeroportos europeus e respectivas rotas entre os mesmos. Cada aeroporto é representado pelo seu nome e as rotas incluem a distância em kms e o tráfego - volume de passageiros transportados.

Supondo que esta rede de aeroportos europeus é representada por um grafo não dirigido usando a classe template ListAdjGrafo<TV,TR>:

- a) Esboce a declaração da classe (não *template*) AeropEur e respectivas classes TV e TR indicando apenas os atributos das classes (não é necessário indicar os métodos das classes).

Na classe AeropEur defina os métodos para responder às alíneas a seguir indicadas:

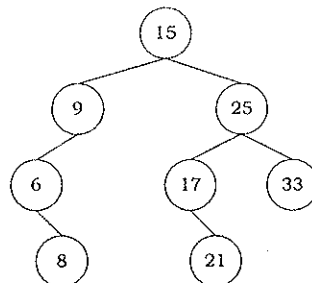
- b) Retornar todos os aeroportos alcançáveis a partir de um determinado aeroporto de origem, apresentando os destinos directos em primeiro lugar  
c) Apresentar a ligação directa (origem, destino, tráfego) com maior volume de tráfego  
d) Apresentar todas as rotas possíveis entre dois aeroportos.

**4. [ 3 valores]**

O método a seguir descrito realiza uma visita numa árvore binária.

```
template <class T>
void ArvBin<T>:: visita (Nodo<T>* raiz)
{
    if (raiz)
    {
        visita (raiz->dir) ;
        cout << raiz->inf ;
        visita (raiz->esq) ;
    }
}
```

- a) Passando-lhe a raiz da árvore abaixo representada, indique a ordem pela qual os nós da árvore são visitados.



- b) A árvore acima é uma árvore AVL? Justifique. Em caso negativo, efectue as rotações necessárias para a transformar numa árvore AVL.

**1. [ 7 valores]**

Relembre a representação em binário de números inteiros não negativos, que faz corresponder por exemplo a 53 a sequência 110101 ( $1 \times 2^5 + 1 \times 2^4 + 1 \times 2^2 + 1 \times 2^0 = 53_{\text{decimal}}$ ). Suponha que cada número é implementado recorrendo à classe template `Lista<T>`, contendo cada célula apenas o valor zero ou um (0 ou 1). A ordem pela qual a sequência de dígitos está guardada corresponde à da leitura do número da esquerda para a direita (i.e., a cabeça da lista aponta para o dígito mais significativo).

- a) Implemente um método recursivo, exterior à classe template `Lista<T>`, que dado um número inteiro decimal cria uma lista com a correspondente representação em binário.
- b) A partir da representação anterior de um número binário, e usando a classe `Lista_Iterador<T>` escreva um método exterior a ambas as classes que converte um número binário no respectivo decimal.
- c) Escreva um método, exterior à classe template `Lista<T>`, que modifica uma representação binária reorganizando os seus dígitos de modo a alternar enquanto possível, os zeros e uns da representação binária. Exemplo: 0011010100001 origina 0101010101000

**2. [ 3 valores]**

Considere o método abaixo:

```
void metodo(Stack<int>& stk, Stack<int>& ret)
{
    while (!stk.vazia())
    {
        int x, t=0 ;

        Stack<int> stk_tmp(stk);
        while (!stk_tmp.vazia())
        {
            stk_tmp.pop(x);
            t += x;
        }
        ret.push(t);
        stk.pop(x);
    }
}
```

- a) Explique o que faz o método, exemplificando com uma stack com 3 valores inteiros.
- b) Analise o código quanto à complexidade temporal usando a notação Big-Oh. Justifique.