

1. Observe que a busca binária também admite uma interpretação recursiva: se não se trata do elemento do meio, então faça uma busca (também binária) na metade esquerda (ou na direita) do intervalo. Assim sendo, escreva uma função

```
template <typename T> int busca_binaria (T *v, int i, int f, T e)
```

que procure o elemento "e" no intervalo $v[i..f]$, de forma semelhante à função escrita em sala (isto é, retornando a posição do elemento, caso encontrado, ou retornando -1 caso não encontrado). Se $f < i$, o intervalo $[i..f]$ deve ser interpretado como vazio (logo, o elemento não está presente). Em todo caso, o vetor não deve ser alterado.

Observação: a função deve possuir como pré-condição que, se $i \leq f$, então o intervalo $v[i..f]$ é um intervalo de índices válidos do vetor apontado por v .

Ao final, teste a sua função.

2. Escreva uma variação da implementação de dicionário da aula 14, em que os elementos sejam mantidos ordenados no vetor, de forma a permitir que a busca binária seja utilizada para procurar pelas chaves. Implemente a remoção por meio de um deslocamento à esquerda dos elementos à direita daquele que é removido; a inserção deve ser implementada de forma simétrica (deslocando à direita).

Teste a sua implementação.