

## Problema: Binary search

### 1 - Objetivo

Dado um vetor ordenado identificar a menor posição (devido a valores repetidos) de um valor dado.

### 2 - Análise

De acordo com o tempo de execução e o possível maior tamanho do vetor, uma busca linear não será suficiente, por isso deve-se implementar uma busca binária ou utilizar as funções prontas de “lower\_bound” ou “upper\_bound” devido ao seu tempo de busca ser  $n \cdot \lg(n)$ , sendo que  $\lg$  é log na base 2 e  $n$  o tamanho do vetor.

### 3 - Solução

Ao realizar apenas uma busca binária, deve-se realizar uma outra busca linear até encontrar o menor índice em que o valor desejado está.

Caso queira utilizar algumas das funções mencionadas na análise, a melhor será utilizar a “lower\_bound” pois já devolverá o índice desejado, se utilizar o “upper\_bound” deverá realizar a busca linear da mesma forma que a busca binária.

Em ambos os casos deve-se realizar uma verificação do valor retornado.

A seguir está a proposta da solução na linguagem C++ com busca binária implementada:

```
#include <bits/stdc++.h>
```

```
using namespace std;
```

```
int buscab(int *v, int n, int valor) {
```

```
    int init = 0;
```

```
    int fim = n-1;
```

```
    while(init <= fim) {
```

```
        int mid = (init+fim) /2;
```

```
        if(v[mid] == valor)
```

```
            return mid;
```

```

        else if(v[mid] > valor) {
            fim = mid -1;
        } else {
            init = mid +1;
        }
    }
    return -1;
}

int main() {
    ios_base::sync_with_stdio(false);
    cin.tie(NULL);
    int N; cin >> N;
    int qtd, number, v[N]; cin >> qtd;
    for (int i = 0; i < N; i++)
        cin >> v[i];
    for (int i = 0; i < qtd; i++) {
        cin >> number;
        int pos = buscab(v, N, number);
        while(pos > 0 && v[pos] == v[pos-1])
            pos--;
        cout << pos << '\n';
    }
    return 0;
}

```

A seguir está a proposta da solução na linguagem C++ com busca binária utilizando “lower\_bound”:

```

#include <bits/stdc++.h>

using namespace std;

int main() {
    ios_base::sync_with_stdio(false);
    cin.tie(NULL);
    int N; cin >> N;
    int qtd, number, v[N]; cin >> qtd;
    for (int i = 0; i < N; i++)
        cin >> v[i];
    for (int i = 0; i < qtd; i++) {
        cin >> number;
        auto pos = lower_bound(v, v+N, number);
        cout << ((*pos != number || pos == v+N) ? -1 : pos-v) << '\n';
    }
    return 0;
}

```