

* BUSCA BINARIA *

$\log_2 m \rightarrow$ TEMPO QUE LEVA PARA BUSCAR O ELEMENTO EM UMA LISTA DE m ELEMENTOS.

```
public class BinarySearch {  
    public static final int NOT_FOUND = -1;  
    public static int binarySearch (Comparable[] A, Comparable x) {  
        int low = 0;  
        int high = A.length - 1;  
        int mid;  
        while (low <= high) {  
            mid = (low + high) / 2;  
            if (A[mid].compareTo(x) < 0) {  
                low = mid + 1;  
            } else if (A[mid].compareTo(x) > 0) {  
                high = mid - 1;  
            } else {  
                return mid;  
            }  
        }  
        return NOT_FOUND;  
    }  
    public static void main (String[] args) {  
        int size = 8;  
        Comparable[] A = new Integer[size];  
        for (int i = 0; i < A.length; i++) A[i] = new Integer(i * 2);  
        for (int i = 0; i < A.length; i++) System.out.println("encontre" + i + " em" +  
            binarySearch(A, new Integer(i)));  
    }  
}
```

- EXERCÍCIOS

1.1. SUPONHA QUE VOCE TENHA UMA LISTA COM 128 NOMES E ESTEJA FAZENDO UMA PESQUISA BINÁRIA. QUAL SERIA O NÚMERO MÁXIMO DE ETAPAS QUE VOCE LEVARIA P/ ENCONTRAR O NOME DESEJADO?

SUPONDO QUE A LISTA TA EM ORDEM, TEMOS

$$\log_2 m \therefore m = 128 \therefore \log_2 128 = 7 //$$

7 VEZES //

1.2. SUPONHA QUE VOCE DUPLIQUE O TAMANHO DA LISTA. QUAL SERIA O NÚMERO MÁXIMO DE ETAPAS AGORA?

$$\log_2 m \therefore m = 128 \cdot 2 = 256 \therefore \log_2 256 = 8 //$$

8 VEZES //