

1.1/busca.h

```
1  #ifndef BUSCA_H
2  #define BUSCA_H
3
4  #include <stdio.h>
5  #include <stdlib.h>
6  #include <time.h>
7
8  //Medidas de Complexidade
9  int comp; //Num. de comparacoes
10
11 int* copiaVetor(int* v, int n){
12     int i;
13     int *v2;
14     v2 = (int*) malloc (n*sizeof(int));
15     for(i=0; i<n; i++) v2[i] = v[i];
16     return v2;
17 }
18 void imprimeVetor(int* v, int n){
19     int i, prim = 1;
20     printf("[");
21     for(i=0; i<n; i++)
22         if(prim){ printf("%d", v[i]); prim = 0; }
23         else printf(", %d", v[i]);
24     printf("]\n");
25 }
26
27 void preencheAleatorio(int* v, int n, int ini, int fim){
28     int i;
29     for(i=0; i<n; i++)
30         v[i] = ini + rand() % (fim-ini + 1);
31 }
32
33 void troca(int* a, int *b){
34     int aux = *a;
35     *a = *b;
36     *b = aux;
37 }
38
39 int buscaSequencial(int *v, int n, int elem){
40     int i;
41     for(i=0; i<n; i++){
42         comp++;
43         if(v[i] == elem)
44             return i; //Elemento encontrado
45     }
46     return -1; //Elemento encontrado
47 }
48
49 int particao(int *v, int ini, int fim){
50     int i = ini, j = fim;
51     int pivo = v[(ini+fim)/2];
52     while (1) {
53         while(v[i] < pivo){ i++; } //procura algum >= pivo do lado esquerdo
54         while(v[j] > pivo){ j--; } //procura algum <= pivo do lado direito
55
56         if(i<j){
57             troca(&v[i], &v[j]); //troca os elementos encontrados
```

```
58         i++;
59         j--;
60     }else
61         return j; //retorna o local onde foi feita a particao
62     }
63 }
64
65 void QuickSort(int *v, int ini, int fim){
66     if(ini < fim ){
67         int q = particao(v, ini, fim);
68         QuickSort(v, ini, q);
69         QuickSort(v, q+1, fim);
70     }
71 }
72
73 int rec_buscaBinaria(int *v, int ini, int fim, int elem){
74     if(ini > fim) return -1;
75     int meio = (ini + fim)/2;
76     comp++;
77     if(v[meio] == elem)
78         return meio;
79     else
80         if(elem < v[meio])
81             return rec_buscaBinaria(v, ini, meio-1, elem);
82         else
83             return rec_buscaBinaria(v, meio+1, fim, elem);
84 }
85
86 int it_buscaBinaria(int *v, int ini, int fim, int elem){
87     int meio;
88     while(ini <= fim){
89         meio = (ini + fim)/2;
90         comp++;
91         if(elem == v[meio]) return meio;
92         else
93             if(elem < v[meio])
94                 fim = meio-1;
95             else
96                 ini = meio+1;
97     }
98     return -1;
99 }
100
101 #endif
```