

1.3/Quick.h

```

1  /*----- File: Quick.c -----+
2  |Quick Sort                      |
3  |                               |
4  |                               |
5  | Implementado por Guilherme C. Pena em 14/11/2023 |
6  +-----+ */
7  #ifndef QUICK_H
8  #define QUICK_H
9
10
11 #include <stdio.h>
12 #include <stdlib.h>
13 #include <time.h>
14
15
16 int* copiaVetorQuick(int* v, int n){
17     int i;
18     int *v2;
19     v2 = (int*) malloc (n*sizeof(int));
20     for(i=0; i<n; i++) v2[i] = v[i];
21     return v2;
22 }
23 void imprimeVetorQuick(int* v, int n){
24     int i, prim = 1;
25     printf("[");
26     for(i=0; i<n; i++)
27         if(prim){ printf("%d", v[i]); prim = 0; }
28         else printf(", %d", v[i]);
29     printf("]\n");
30 }
31
32 void preencheAleatorioQuick(int* v, int n, int ini, int fim){
33     int i;
34     for(i=0; i<n; i++){
35         v[i] = ini + rand() % (fim-ini + 1);
36         //v[i] = (n-i); //Para o pior caso
37     }
38 }
39
40 void trocaQuick(int* a, int *b){
41     int aux = *a;
42     *a = *b;
43     *b = aux;
44 }
45
46 int particao(int *v, int ini, int fim, long *comp, long *mov){
47     int i = ini, j = fim;
48     int pivo = v[(ini+fim)/2];
49     while (1) {
50
51         while(v[i] < pivo){ i++; (*comp)++; } // Incrementa comparação
52
53         while(v[j] > pivo){ j--; (*comp)++; } // Incrementa comparação
54
55         if(i < j){
56             (*comp)++;
57             (*mov)++;

```

```
58     trocaQuick(&v[i], &v[j]); // Troca os elementos encontrados
59     i++;
60     j--;
61 } else {
62     return j; // Retorna o local onde foi feita a partição
63 }
64 }
65 }
66
67 void QuickSort(int *v, int ini, int fim, int n, long *comp, long *mov){
68     if(ini < fim ){
69         int q = particao(v, ini, fim, comp, mov);
70         QuickSort(v, ini, q, n, comp, mov);
71         QuickSort(v, q+1, fim, n, comp, mov);
72     }
73 }
74
75 #endif
```