

1.2/Quick.h

```
1
2 #ifndef QUICK_H
3 #define QUICK_H
4
5
6 #include <stdio.h>
7 #include <stdlib.h>
8 #include <time.h>
9
10 //Medidas de Complexidade
11 int comp; //Num. de comparacoes
12 int mov; //Num. de movimentacoes
13
14 int* copiaVetorQuick(int* v, int n){
15     int i;
16     int *v2;
17     v2 = (int*) malloc (n*sizeof(int));
18     for(i=0; i<n; i++) v2[i] = v[i];
19     return v2;
20 }
21 void imprimeVetorQuick(int* v, int n){
22     int i, prim = 1;
23     printf("[");
24     for(i=0; i<n; i++)
25         if(prim){ printf("%d", v[i]); prim = 0; }
26         else printf(", %d", v[i]);
27     printf("]\n");
28 }
29
30 void preencheAleatorioQuick(int* v, int n, int ini, int fim){
31     int i;
32     for(i=0; i<n; i++){
33         v[i] = ini + rand() % (fim-ini + 1);
34         //v[i] = (n-i); //Para o pior caso
35     }
36 }
37
38 void trocaQuick(int* a, int *b){
39     int aux = *a;
40     *a = *b;
41     *b = aux;
42 }
43
44 //Versao do livro
45 int particiona(int *v, int ini, int fim){
46     int esq, dir, pivo, aux;
47     esq = ini; dir = fim;
48     pivo = v[ini];
49     while(esq < dir){
50         while(esq <= fim && v[esq] <= pivo) esq++;
51         while(dir >= 0 && v[dir] > pivo) dir--;
52         if(esq < dir) trocaQuick(&v[esq], &v[dir]);
53     }
54     v[ini] = v[dir];
55     v[dir] = pivo;
56     return dir;
57 }
```

```
58
59 int particao(int *v, int ini, int fim){
60     int i = ini, j = fim;
61     int pivo = v[(ini+fim)/2];
62     while (1) {
63         comp++;
64         while(v[i] > pivo){ i++; comp++; } // Modificado para ordenar em ordem
decrecente
65
66         comp++;
67         while(v[j] < pivo){ j--; comp++;} // Modificado para ordenar em ordem
decrecente
68
69         if(i<j){
70             trocaQuick(&v[i], &v[j]); // Troca os elementos encontrados
71             mov++;
72             i++;
73             j--;
74         }else
75             return j; // Retorna o local onde foi feita a partiç o
76     }
77 }
78
79 void QuickSort(int *v, int ini, int fim, int n){
80     if(ini < fim ){
81         int q = particao(v, ini, fim);
82         //printf("Parts: (%d, %d) e (%d, %d): ", ini, q, q+1, fim);
83         //imprimeVetor(v, n);
84         QuickSort(v, ini, q, n);
85         QuickSort(v, q+1, fim, n);
86     }
87 }
88
89 #endif
```