1.1/Quick.h

```c
/*---------------- File: Quick.c  ----------------------+
|Quick Sort                                             |
|                                                       |
|                                                       |
| Implementado por Guilherme C. Pena em 14/11/2023      |
+-------------------------------------------------------+ */
#ifndef QUICK_H
#define QUICK_H


#include <stdio.h>
#include <stdlib.h>
#include <time.h>

//Medidas de Complexidade
int comp; //Num. de comparacoes
int mov; //Num. de movimentacoes

int* copiaVetorQuick(int* v, int n){
    int i;
    int *v2;
    v2 = (int*) malloc (n*sizeof(int));
    for(i=0; i<n; i++) v2[i] = v[i];
    return v2;
}
void imprimeVetorQuick(int* v, int n){
    int i, prim = 1;
    printf("[");
    for(i=0; i<n; i++)
        if(prim){ printf("%d", v[i]); prim = 0; }
        else printf(", %d", v[i]);
    printf("]\n");
}

void preencheAleatorioQuick(int* v, int n, int ini, int fim){
    int i;
    for(i=0; i<n; i++){
        v[i] = ini + rand() % (fim-ini + 1);
        //v[i] = (n-i); //Para o pior caso
    }
}

void trocaQuick(int* a, int *b){
    int aux = *a;
    *a = *b;
    *b = aux;
}

//Versao do livro
int particiona(int *v, int ini, int fim){
    int esq, dir, pivo, aux;
    esq = ini; dir = fim;
    pivo = v[ini];
    while(esq < dir){
        while(esq <= fim && v[esq] <= pivo) esq++;
        while(dir >= 0 && v[dir] > pivo) dir--;
        if(esq < dir) trocaQuick(&v[esq], &v[dir]);
```

```
58      }
59      v[ini] = v[dir];
60      v[dir] = pivo;
61      return dir;
62  }
63
64  int particao(int *v, int ini, int fim){
65      int i = ini, j = fim;
66      int pivo = v[(ini+fim)/2];
67      while (1) {
68          comp++;
69          while(v[i] < pivo){ i++; comp++; } //procura algum >= pivo do lado esquerdo
70
71          comp++;
72          while(v[j] > pivo){ j--; comp++;} //procura algum <= pivo do lado direito
73
74          if(i<j){
75              trocaQuick(&v[i], &v[j]); //troca os elementos encontrados
76              mov++;
77              i++;
78              j--;
79          }else
80              return j; //retorna o local onde foi feita a particao
81      }
82  }
83
84  void QuickSort(int *v, int ini, int fim, int n){
85      if(ini < fim ){
86          int q = particao(v, ini, fim);
87          //printf("Parts: (%d, %d) e (%d, %d): ", ini, q, q+1, fim);
88          //imprimeVetor(v, n);
89          QuickSort(v, ini, q, n);
90          QuickSort(v, q+1, fim, n);
91      }
92  }
93
94  #endif
```