

## 1.1/Merge.h

```

1  /*----- File: Merge.c -----+
2  |Merge Sort                      |
3  |                               |
4  |                               |
5  | Implementado por Guilherme C. Pena em 08/11/2023 |
6  +-----+ */
7  #ifndef MERGE_H
8  #define MERGE_H
9
10
11 #include <stdio.h>
12 #include <stdlib.h>
13 #include <time.h>
14
15 //Medidas de Complexidade
16 int comp; //Num. de comparacoes
17 int mov; //Num. de movimentacoes
18
19 int* copiaVetorMerge(int* v, int n){
20     int i;
21     int *v2;
22     v2 = (int*) malloc (n*sizeof(int));
23     for(i=0; i<n; i++) v2[i] = v[i];
24     return v2;
25 }
26 void imprimeVetorMerge(int* v, int n){
27     int i, prim = 1;
28     printf("[");
29     for(i=0; i<n; i++)
30         if(prim){ printf("%d", v[i]); prim = 0; }
31         else printf(", %d", v[i]);
32     printf("]\n");
33 }
34
35 void preencheAleatorioMerge(int* v, int n, int ini, int fim){
36     int i;
37     for(i=0; i<n; i++)
38         v[i] = ini + rand() % (fim-ini + 1);
39 }
40
41 void trocaMerge(int* a, int *b){
42     int aux = *a;
43     *a = *b;
44     *b = aux;
45 }
46
47 void Merge(int *v, int ini, int meio, int fim){
48     int tam = fim-ini+1;
49     //Vetor Auxiliar - A
50     int *A = (int*) malloc (tam*sizeof(int));
51     int i = ini, j = meio+1, k = 0;
52     while (i<=meio && j<=fim) {
53         if (v[i] < v[j]){ A[k] = v[i]; i++; }
54         else { A[k] = v[j]; j++; }
55         k++;
56     }
57     while (i<=meio) { A[k] = v[i]; i++; k++; }

```

```
58 |     while (j<=fim) { A[k] = v[j]; j++; k++; }
59 |     for(i = ini, k=0; i<=fim; i++, k++) v[i] = A[k];
60 |     free(A);
61 | }
62 |
63 | void MergeSort(int *v, int ini, int fim){
64 |     if(ini < fim ){
65 |         int meio = (ini + fim)/2;
66 |         MergeSort(v, ini, meio);
67 |         MergeSort(v, meio+1, fim);
68 |         Merge(v, ini, meio, fim);
69 |     }
70 | }
71 |
72 |
73 | #endif
```