

1.2/Merge.h

```
1
2 #ifndef MERGE_H
3 #define MERGE_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* copiaVetorMerge(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 imprimeVetorMerge(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 preencheAleatorioMerge(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 }
35
36 void trocaMerge(int* a, int *b){
37     int aux = *a;
38     *a = *b;
39     *b = aux;
40 }
41
42 void Merge(int *v, int ini, int meio, int fim){
43     int tam = fim - ini + 1;
44     int *A = (int*) malloc(tam * sizeof(int));
45     int i = ini, j = meio + 1, k = 0;
46
47     while (i <= meio && j <= fim) {
48         if (v[i] > v[j]) {
49             A[k] = v[i];
50             i++;
51         } else {
52             A[k] = v[j];
53             j++;
54         }
55         k++;
56     }
57 }
```

```
58     while (i <= meio) {
59         A[k] = v[i];
60         i++;
61         k++;
62     }
63
64     while (j <= fim) {
65         A[k] = v[j];
66         j++;
67         k++;
68     }
69
70     for (i = ini, k = 0; i <= fim; i++, k++) {
71         v[i] = A[k];
72     }
73
74     free(A);
75 }
76
77
78 void MergeSort(int *v, int ini, int fim){
79     if(ini < fim ){
80         int meio = (ini + fim)/2;
81         MergeSort(v, ini, meio);
82         MergeSort(v, meio+1, fim);
83         Merge(v, ini, meio, fim);
84     }
85 }
86
87
88 #endif
```