

### 1.3/desempenho.h

```
1  #ifndef DESEMPENHO_H
2  #define DESEMPENHO_H
3
4  #include "Heap.h"
5  #include "Merge.h"
6  #include "Quick.h"
7  #include "Shell.h"
8  #include <stdio.h>
9  #include <stdlib.h>
10 #include <string.h>
11 #include <time.h>
12
13 void medirDesempenho(int *valores, int tamanho, char *tipoOrdenacao) {
14     int *copiaValores = (int *)malloc(tamanho * sizeof(int));
15     for (int i = 0; i < tamanho; ++i) {
16         copiaValores[i] = valores[i];
17     }
18
19     clock_t inicio, fim;
20     double tempoExecucao;
21     long comp = 0, mov = 0;
22
23     inicio = clock();
24     if (strcmp(tipoOrdenacao, "Heap") == 0) {
25         HeapSort(copiaValores, tamanho, &comp, &mov);
26     } else if (strcmp(tipoOrdenacao, "Merge") == 0) {
27         MergeSort(copiaValores, 0, tamanho-1, &comp, &mov);
28     } else if (strcmp(tipoOrdenacao, "Quick") == 0) {
29         QuickSort(copiaValores, 0, tamanho-1, tamanho, &comp, &mov);
30     } else if (strcmp(tipoOrdenacao, "Shell") == 0) {
31         ShellSort(copiaValores, tamanho, &comp, &mov);
32     }
33
34     else {
35         printf("Algoritmo de ordenação não está disponível\n");
36         free(copiaValores);
37         return;
38     }
39     fim = clock();
40
41     tempoExecucao = ((double)(fim - inicio)) / CLOCKS_PER_SEC;
42
43     printf("Tipo de ordenacao: %s\n", tipoOrdenacao);
44     printf("Tempo de execucao: %f segundos\n", tempoExecucao);
45     printf("Numero de comparações: %ld\n", comp);
46     printf("Numero de movimentações: %ld\n\n", mov);
47
48     free(copiaValores);
49 }
50
51
52 #endif
```