

1. Implemente os algoritmos de ordenação para ordenar um vetor de elementos inteiros: insertion sort, selection sort e merge sort. Os algoritmos devem ser implementados através de funções conforme abaixo:

- void insertion sort(int \*v, int size);
- void selection sort(int \*v, int size);
- void merge sort(int \*v, int esquerda, int direita);

2. Crie um programa que inicializa dinamicamente em memória e preenche com valores aleatórios 4 vetores de valores inteiros contendo respectivamente 100, 1000, 10000, 100000 elementos e compare o tempo gasto pelos algoritmos implementados na questão 1 para ordenar estes vetores. Siga as dicas abaixo:

- É necessário salvar uma cópia do vetor para usá-lo com os diferentes algoritmos;
- Para gerar números aleatórios:

```
#include <stdlib.h>
```

```
int *preenchevetor(int size){
    v = (int *)malloc(sizeof(int)*size);
    // inicializa a semente
    srand(time(NULL));
    int i;
    for (i = 0; i < size; i++){
        v[i] = rand();
    }
    return v;
}
```

- Para medir o tempo gasto por uma parte do programa:

```
#include <sys/time.h>
```

```
int main(){
    struct timeval inicio, fim;

    // marca o momento do inicio
    gettimeofday(&inicio, NULL);

    // executa o objeto de teste

    // marca o momento do termino
    gettimeofday(&fim, NULL);

    // calcula o tempo gasto
    printf ("Tempo gasto = %f segundos\n",
            (double) (fim.tv_usec - inicio.tv_usec) / 1000000 +
            (double) (fim.tv_sec - inicio.tv_sec));
}
```