

1. Implemente os algoritmos de ordenação para ordenar um vetor de elementos inteiros: insertionsort, selectionsort, mergesort e quicksort (com pivô sendo o elemento do meio do vetor). Os algoritmos devem ser implementados através de funções conforme abaixo:

- void insertionsort(int *v, int size);
- void selectionsort(int *v, int size);
- void mergesort(int *v, int esquerda, int direita);
- void quicksort(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));  
}
```

3. Altere o algoritmo quicksort para usar como pivô a mediana entre dos elementos presentes nos extremos e no meio do vetor (mediana dos 3 elementos). Refaça a simulação da questão 2 com esta versão do quicksort.

4. Crie um algoritmo de ordenação quicksort para ordenar o pequeno banco de dados abaixo, usando como chave o Nome do funcionário.

Idade	Nome	Salario	Departamento
40	Joao	100.43	Matriz
42	Maria	200.32	Filial
35	Amalia	50.54	Matriz
30	Joao	150.73	Filial
32	Mario	250.22	Matriz
25	Amauri	60.14	Matriz