

Todos os programas devem ser escritos em linguagem C.

Entregar até 9/10. A entrega deverá ser um arquivo .zip com o seguinte formato: nome_lista3.zip.

1. Determine a complexidade dos fragmentos de código abaixo, em função de N :

(a)

```
int soma = 0;
for (int i = N; i > 0; i /= 2)
    for (int j = 0; j < i; j++)
        soma++;
```

(b)

```
int soma = 0;
for (int i = 1; i < N; i *= 2)
    for (int j = 0; j < i; j++)
        soma++;
```

(c)

```
int soma = 0;
for (int i = 1; i < N; i *= 2)
    for (int j = 0; j < N; j++)
        soma++;
```

2. Sabe-se que o tempo de execução de certo algoritmo com uma entrada de tamanho N_0 é T_k . Vários experimentos mostraram que, dobrando-se N_0 sucessivamente, o fator de incremento é 2^b . Ou seja, $T(2N_0) = 2^b T_k$, e assim por diante. Determine $T(N)$, i.e., a expressão que permite calcular o tempo estimado de execução para uma entrada genérica N .
3. Crie um arquivo `ordenacao.c` e, nele:
 - (a) implemente uma função que realize a ordenação por seleção. Teste-a na `main`.
 - (b) implemente uma função que realize a ordenação por inserção. Teste-a na `main`.
 - (c) compare os dois algoritmos, num mesmo vetor aleatorizado, em termos das quantidades de comparações e de trocas. (Dica: utilize

variáveis globais para armazenar essas quantidades para cada algoritmo. Dessa forma, elas podem ser acessadas tanto nas funções quanto na `main`.)