(2.2-1)

$\frac{n^3}{1000} - 100n^2 - 100n + 3$ em termos da notação Θ

$$c_1 \cdot n^2 \leq \frac{n^3}{1000} - 100n^2 - 100n + 3 \leq c_2 \cdot n^3$$

(2.2-2)

n em $A = \langle n \rangle$
↳ números

```
for i = 1 to A.comprimento          c1 n
    Chave = A[i]                    c2 n-1
    for j = 2 to A.comprimento      c3 Σ_{i=2}^{n} t_i
        if Chave > A[j]             c4 Σ_{i=2}^{n} (t_i - 1)
            A[i] = A[j]             c5 Σ_{i=2}^{n} (t_i - 1)
```

Inicialização: na primeira interação o algoritmo é verdadeiro pois A[i] = A[j]. As posições do arranjo se mantém corretas.

Manutenção: Após entrar no segundo for as posições do arranjo se mantém a mesma pois o valor de i e j apontam para posições do arranjo isso persiste até próxima interação enquanto o valor da posição i for maior que a posição em j.

Término: O código finaliza suas interações após percorrer todo o arranjo para as posições de i, ou seja quando $i = n$.

b) Por que para a posição inicial a condicional de permuta não ocorre.

c) $T(n) = c_1 n + c_2(n-1) + c_3(n-1) + c_4(n-1) + c_5(n-1)$ melhor caso

$$T(n) = c_1 n + c_2(n-1) + c_3\left(\frac{n(n+1)}{2} - 1\right) + c_4\left(\frac{n(n-1)}{2}\right) + c_5\left(\frac{n(n-1)}{2}\right)$$

(2-2-3)

$T(n) = C_1 n + C_2 (n-1)$ médio

$n-1$ elementos

$T(n) = C_1 n + C_2 n$ pior

n elementos

(2.2-4)

Modificar o algoritmo para encontrar uma resposta que já é conhecida.

Digitalizado com CamScanner