

LISTA 2

CHECKLIST

entregue ☐ fiz ☐ incompleto ☐ não entendi ☐

questão 1

a) ☐ b) ☐ c) ☐ d) ☐ e) ☐

questão 2 ☐

questão 3 ☐

questão 4 ☐

MAC0338 - ANÁLISE DE ALGORITMOS

LISTA 2

exercícios 1(a), 1(d), 3 e 4.

1. Resolva as recorrências abaixo.

(a) $T(n) = 2T(\lfloor n/2 \rfloor) + \Theta(n^2)$

$T(1) = 1$ e $T(n) = 2T(n/2) + n^2$ para $n \geq 2$ potência de 2

por expansão

$$T(n) = 2T(n/2) + n^2$$

$$= 2 \left(2T\left(\frac{n}{2^2}\right) + \frac{n^2}{2^2} \right) + n^2 = 2^2 T\left(\frac{n}{2^2}\right) + 2 \frac{n^2}{2^2} + n^2 = 2^2 T\left(\frac{n}{2^2}\right) + \frac{3n^2}{2}$$

$$= 2^2 \left(2T\left(\frac{n}{2^3}\right) + \frac{n^2}{2^4} \right) + \frac{3n^2}{2} = 2^3 T\left(\frac{n}{2^3}\right) + 2^2 \frac{n^2}{2^4} + \frac{3n^2}{2} = 2^3 T\left(\frac{n}{2^3}\right) + \frac{7n^2}{2^2}$$

$$= 2^3 \left(2T\left(\frac{n}{2^4}\right) + \frac{n^2}{2^6} \right) + \frac{7n^2}{2^2} = 2^4 T\left(\frac{n}{2^4}\right) + 2^3 \frac{n^2}{2^6} + \frac{7n^2}{2^2} = 2^4 T\left(\frac{n}{2^4}\right) + \frac{15n^2}{2^3}$$

$$= 2^4 \left(2T\left(\frac{n}{2^5}\right) + \frac{n^2}{2^8} \right) + \frac{15n^2}{2^3} = 2^5 T\left(\frac{n}{2^5}\right) + 2^4 \frac{n^2}{2^8} + \frac{15n^2}{2^3} = 2^5 T\left(\frac{n}{2^5}\right) + \frac{31n^2}{2^4}$$

$$= \dots = 2^k T\left(\frac{n}{2^k}\right) + (2^k - 1) \frac{n^2}{2^{k-1}} \text{ para } k = \lg n$$

$$= 2^{\lg n} T\left(\frac{n}{2^{\lg n}}\right) + (2^{\lg n} - 1) \frac{n^2}{2^{\lg n - 1}}$$

$$= n T\left(\frac{n}{n}\right) + (n-1) \frac{n^2}{2} = n + (n-1) \frac{2}{n} n^2$$

$$= n + (n-1) 2n = n + 2n^2 - 2n$$

$$= 2n^2 - n$$

$$= \Theta(n^2)$$

verificação

$$T(n) = 2T(n/2) + n^2 = n + n^2$$

por hipótese de indução

$$T(n) = 2T\left(\frac{n}{2}\right) + n^2$$

$$= 2 \left(2 \frac{n^2}{2^2} - \frac{n}{2} \right) + n^2 = n^2 - n + n^2$$

$$= 2n^2 - n$$

$$\text{logo, } T(n) = 2T\left(\frac{n}{2}\right) + n^2 = n + n^2 = \Theta(n^2)$$

(b) $T(n) = 8T(\lfloor n/2 \rfloor) + \Theta(n^2)$

$T(1) = 1$ e $T(n) = 8T\left(\frac{n}{2}\right) + n^2$ para $n \geq 2$ potência de 2

por expansão

$$T(n) = 8T\left(\frac{n}{2}\right) + n^2$$

$$= 8\left(8T\left(\frac{n}{2^2}\right) + \frac{n^2}{2^2}\right) + n^2 = 8^2 T\left(\frac{n}{2^2}\right) + 2^3 \frac{n^2}{2^2} + n^2 = 8^2 T\left(\frac{n}{2^2}\right) + 2n^2 + n^2 = 3n^2$$

$$= 8^2 \left(8T\left(\frac{n}{2^3}\right) + \frac{n^2}{2^4}\right) + 2n^2 + n^2 = 8^3 T\left(\frac{n}{2^3}\right) + 2^6 \frac{n^2}{2^4} + 2n^2 + n^2 = 8^3 T\left(\frac{n}{2^3}\right) + 4n^2 + 2n^2 + n^2 = 7n^2$$

$$= 8^3 \left(8T\left(\frac{n}{2^4}\right) + \frac{n^2}{2^6}\right) + 4n^2 + 2n^2 + n^2 = 8^4 T\left(\frac{n}{2^4}\right) + 2^9 \frac{n^2}{2^6} + 4n^2 + 2n^2 + n^2 = 8^4 T\left(\frac{n}{2^4}\right) + 8n^2 + 4n^2 + 2n^2 + n^2 = 15n^2$$

$$= \dots = 8^k T\left(\frac{n}{2^k}\right) + \sum_{i=0}^{k-1} 2^i n^2 \text{ para } k = \lg n$$

$$= 8^{\lg n} T\left(\frac{n}{2^{\lg n}}\right) + \left(\sum_{i=0}^{\lg n} 2^i\right) n^2 = (2^3)^{\lg n} T\left(\frac{n}{n}\right) + \left(\frac{1 \cdot (2^{\lg n} - 1)}{2 - 1}\right) n^2$$

$$= n^3 + (n - 1) n^2 = n^3 + n^3 - n^2 = 2n^3 - n^2$$

$$= \Theta(n^3)$$

verificação

por hipótese de indução

• verificação:

• pegar o $T(n)$ original e depois colocar o $T(n)$ encontrado no original

• tem que resultar no $T(n)$ encontrado

$$T(n) = 8T\left(\frac{n}{2}\right) + n^2$$

$$= 8 \left(2 \frac{n^3}{2^3} - \frac{n^2}{2^2} \right) + n^2$$

$$= 2^4 \frac{n^3}{2^3} - 2^3 \frac{n^2}{2^2} + n^2$$

$$= 2n^3 - 2n^2 + n^2$$

$$= 2n^3 + n^2$$

$$\text{logo, } T(n) = 8T\left(\frac{n}{2}\right) + n^2 = 2n^3 + n^2 = \Theta(n^3)$$

(c) $T(n) = 2T(\lfloor n/2 \rfloor) + \Theta(n^3)$

$T(1) = 1$ e $T(n) = 2T(\frac{n}{2}) + n^3$ para $n \geq 2$ potência de 2.

por expansão.

$$T(n) = 2T\left(\frac{n}{2}\right) + n^3$$

$$= 2\left(2T\left(\frac{n}{2^2}\right) + \frac{n^3}{2^3}\right) + n^3 = 2^2 T\left(\frac{n}{2^2}\right) + \frac{n^3}{2^3} + n^3$$

$$= 2^2\left(2T\left(\frac{n}{2^3}\right) + \frac{n^3}{2^6}\right) + \frac{n^3}{2^3} + n^3 = 2^3 T\left(\frac{n}{2^3}\right) + \frac{n^3}{2^6} + \frac{n^3}{2^3} + n^3$$

$$= 2^3\left(2T\left(\frac{n}{2^4}\right) + \frac{n^3}{2^9}\right) + \frac{n^3}{2^6} + \frac{n^3}{2^3} + n^3 = 2^4 T\left(\frac{n}{2^4}\right) + \frac{n^3}{2^9} + \frac{n^3}{2^6} + \frac{n^3}{2^3} + n^3$$

$$= \dots = 2^k T\left(\frac{n}{2^k}\right) + \left(\sum_{i=0}^{k-1} \frac{1}{2^i}\right) n^3 \quad \text{para } k = \lg n$$

$$= 2^{\lg n} T\left(\frac{n}{2^{\lg n}}\right) + \left(\sum_{i=0}^{\lg n} \frac{1}{2^{3^i}}\right) n^3 = n + 1 \left(\frac{\left(\frac{1}{2^3}\right)^{\lg n} - 1}{\frac{1}{2^3} - 1}\right) n^3$$

$$= n + \frac{\frac{1}{n^3} - 1}{\frac{1}{2^3} - 1} n^3 = n + \left(\frac{1}{n^3} - 1\right) \left(-\frac{8}{7}\right) n^3 = n - \frac{8}{7} + \frac{8}{7} n^3 = \Theta(n^3)$$

verificação

$$T(n) = 2T\left(\frac{n}{2}\right) + n^3$$

$$(d) T(n) = 7T(\lfloor n/3 \rfloor) + \Theta(n^2)$$

$$T(1) = 1 \text{ e } T(n) = 7T\left(\frac{n}{3}\right) + n^2 \text{ para } n \geq 3 \text{ potência de 3}$$

por expansão

$$T(n) = 7T\left(\frac{n}{3}\right) + n^2$$

$$= 7\left(7T\left(\frac{n}{3^2}\right) + \frac{n^2}{3^2}\right) + n^2 = 7^2 T\left(\frac{n}{3^2}\right) + 7\frac{n^2}{3^2} + n^2$$

$$= 7^2\left(7T\left(\frac{n}{3^3}\right) + \frac{n^2}{3^4}\right) + 7\frac{n^2}{3^2} + n^2 = 7^3 T\left(\frac{n}{3^3}\right) + 7^2\frac{n^2}{3^4} + 7\frac{n^2}{3^2} + n^2$$

$$= 7^3\left(7T\left(\frac{n}{3^4}\right) + \frac{n^2}{3^6}\right) + 7^2\frac{n^2}{3^4} + 7\frac{n^2}{3^2} + n^2 = 7^4 T\left(\frac{n}{3^4}\right) + 7^3\frac{n^2}{3^6} + 7^2\frac{n^2}{3^4} + 7\frac{n^2}{3^2} + n^2$$

$$= 7^4\left(7T\left(\frac{n}{3^5}\right) + \frac{n^2}{3^8}\right) + 7^3\frac{n^2}{3^6} + 7^2\frac{n^2}{3^4} + 7\frac{n^2}{3^2} + n^2 = 7^5 T\left(\frac{n}{3^5}\right) + 7^4\frac{n^2}{3^8} + 7^3\frac{n^2}{3^6} + 7^2\frac{n^2}{3^4} + 7\frac{n^2}{3^2} + n^2$$

$$= \dots = 7^K T\left(\frac{n}{3^K}\right) + \left(\sum_{i=0}^K \frac{7^i}{3^{2i}}\right) n^2 \text{ para } K = \log_3 n$$

$$= 7^{\log_3 n} T\left(\frac{n}{n}\right) + \left(\sum_{i=0}^{\log_3 n} \frac{7^i}{3^{2i}}\right) n^2$$

$$= 7^{\log_3 n} + \left(\frac{\left(\frac{7}{3^2}\right)^{\log_3 n} - 1}{\frac{7}{3^2} - 1}\right) n^2 = 7^{\log_3 n} + \left(\frac{7^{\log_3 n} - 1}{\frac{7}{n^2} - 1}\right) \cdot \left(-\frac{9}{2}\right) n^2$$

$$= 7^{\log_3 n} + (7^{\log_3 n} - n^2) \cdot \left(-\frac{3^2}{2}\right)$$

$$= \Theta(n^2)$$

verificação

$$T(n) = 7T\left(\frac{n}{3}\right) + n^2$$

por hipótese de indução

$$T(n) = 7T\left(\frac{n}{3}\right) + n^2$$

$$= 7\left(7^{\log_3 \frac{n}{3}} + \left(7^{\log_3 \frac{n}{3}} - \frac{n^2}{3^2}\right) \left(-\frac{3^2}{2}\right)\right) + n^2$$

$$= 7\left(7^{\log_3 n - 1} + \frac{3^2}{2} 7^{\log_3 \frac{n}{3}} - \frac{3^2}{2} \frac{n^2}{3^2}\right) + n^2$$

$$= 7\left(7^{\log_3 n} \cdot \frac{1}{7} - \frac{3^2}{2} 7^{\log_3 \frac{n}{3}} + \frac{n^2}{2}\right) + n^2$$

$$= 7 \cdot 7^{\log_3 n} - 7 \cdot \frac{3^2}{2} 7^{\log_3 \frac{n}{3}} + \frac{3^2}{2} n^2$$

$$= 7^{1+\log_3 n} - \frac{3^2}{2} \cdot 7^{1+\log_3 n-1} + \frac{3^2}{2} n^2$$

$$= 7^{\log_3 n} - \frac{3^2}{2} 7^{\log_3 n} + \frac{3^2}{2} n^2 = 7^{\log_3 n} + (7^{\log_3 n} - n^2) \left(-\frac{3^2}{2}\right)$$

$$\text{logo, } T(n) = 7T\left(\frac{n}{3}\right) + n^2 = 7^{\log_3 n} + (7^{\log_3 n} - n^2) \left(-\frac{3^2}{2}\right) = \Theta(n^2)$$

$$(e) T(n) = T(\lfloor 9n/10 \rfloor) + \Theta(n)$$

$$T(1) = 1 \text{ e } T(n) = T\left(\frac{9n}{10}\right) + n \text{ para } n \geq 10 \text{ potência de } \frac{10}{9}$$

$$T(n) = T\left(\frac{9n}{10}\right) + n$$

$$= \left(T\left(\frac{9^2 n}{10^2}\right) + \frac{9n}{10} \right) + n$$

$$= T\left(\frac{9^3 n}{10^3}\right) + \frac{9^2 n}{10^2} + \frac{9n}{10} + n$$

$$= T\left(\frac{9^4 n}{10^4}\right) + \frac{9^3 n}{10^3} + \frac{9^2 n}{10^2} + \frac{9n}{10} + n$$

$$= \dots = T\left(\frac{9^k n}{10^k}\right) + \left(\sum_{i=0}^{k-1} \frac{9^i}{10^i} \right) n$$

como queremos o caso base

$$T\left(\left(\frac{9}{10}\right)^k n\right) = T(1) \Rightarrow \left(\frac{9}{10}\right)^k n = 1 \Rightarrow n = \left(\frac{10}{9}\right)^k$$

$$\text{logo, } k = \log_{\frac{10}{9}} n$$

$$= 1 + n \left(\frac{\frac{9}{10} \log_{\frac{10}{9}} n - 1}{\frac{9}{10} - 1} \right) = 1 + (-10)(n^{-1} - 1)n$$

$$= 1 + (-10n^{-1} + 10)n = 1 - 10 + 10n = 10n - 9 = O(n)$$

verificação por indução

$$T(n) = T\left(\frac{9}{10}n\right) + n$$

$$= T\left(\frac{9}{10} \cdot 10n - 9\right) + n$$

$$= 9n - 9 + n = 10n - 9$$

$$\text{logo, } T(n) = T\left(\frac{9}{10}n\right) + n = 10n - 9 = O(n)$$

2. Escreva um algoritmo que ordena uma lista de n itens dividindo-a em três sublistas de aproximadamente $n/3$ itens, ordenando cada sublista recursivamente e intercalando as três sublistas ordenadas. Analise seu algoritmo concluindo qual é o seu consumo de tempo.

ORDENA (A, p, r)

1 se $p < r$ e $r \geq 3$

2 então ORDENA ($A, p, \lceil \frac{r}{3} \rceil$)

3 ORDENA ($A, \lceil \frac{r}{3} \rceil + 1, 2 \cdot \lceil \frac{r}{3} \rceil$)

4 ORDENA ($A, 2 \cdot \lceil \frac{r}{3} \rceil + 1, r$)

5 $q \leftarrow \lfloor (L(p + (2 \cdot \lceil \frac{r}{3} \rceil)) / 2) \rfloor$

6 INTERCALA ($A, p, q, (2 \cdot \lceil \frac{r}{3} \rceil)$)

7 $q \leftarrow \lfloor (L(p + r) / 2) \rfloor$

8 INTERCALA (A, p, q, r)

linha consumo de todas as execuções da linha

1 $O(1)$

2 $T(\lceil n/2 \rceil)$

3 $T(\lceil n/2 \rceil)$

4 $T(\lceil n/2 \rceil)$

5 $O(1)$

6 $O(n)$

7 $O(1)$

8 $O(n)$

$$T(n) = 3T(\lceil n/2 \rceil) + 2O(n)$$

análise de recorrência por expansão

$$T(n) = 3T(n/2) + 2n$$

$$= 3\left(3T\left(\frac{n}{2^2}\right) + 2n\right) + 2n = 3^2T\left(\frac{n}{2^2}\right) + 6n + 2n$$

$$= 3^2\left(3T\left(\frac{n}{2^3}\right) + 2n\right) + 6n + 2n = 3^3T\left(\frac{n}{2^3}\right) + 18n + 6n + 2n$$

$$= 3^3\left(3T\left(\frac{n}{2^4}\right) + 2n\right) + 18n + 6n + 2n = 3^4T\left(\frac{n}{2^4}\right) + 54n + 18n + 6n + 2n$$

$$= \dots = 3^k T\left(\frac{n}{2^k}\right) + \left(\sum_{i=0}^k 3^i\right) 2n \quad \text{para } k = \lg n$$

$$= 3^{\lg n} T\left(\frac{n}{2^{\lg n}}\right) + \left(\sum_{i=0}^{\lg n} 3^i\right) 2n = 3^{\lg n} + \left(\frac{1(3^{\lg n} - 1)}{3 - 1}\right) 2n$$

$$= 3^{\lg n} + (3^{\lg n} - 1)n = 3^{\lg n} + 3n^{\lg n} - n = \Theta(n 3^{\lg n})$$

verificação

$$T(n) = 3T\left(\frac{n}{2}\right) + 2n$$

$$= 3T\left(3^{\lg \frac{n}{2}} + \frac{3}{2}n^{\lg \frac{n}{2}} - \frac{n}{2}\right)$$

3. Seja $X[1..n]$ um vetor de inteiros e i e j dois índices distintos de X , ou seja, i e j são inteiros entre 1 e n . Dizemos que o par (i, j) é uma *inversão* de X se $i < j$ e $X[i] > X[j]$. Escreva um algoritmo $O(n \lg n)$ que devolva o número de inversões em um vetor X , onde n é o número de elementos em X .

Podemos construir o algoritmo adaptando o algoritmo de ordenação *mergesort*

seja um vetor $X[a..b]$

INVERSOES (X, a, b)

1 se $a < b$

2 então $q \leftarrow \lfloor (a+b)/2 \rfloor$ q é o pivô

3 $c \leftarrow \text{INVERSOES}(X, a, q) +$

4 $\text{INVERSOES}(X, q+1, b) +$

5 $\text{INTERCALA}(X, a, q, b)$

6 retorna c

7 senão retorna 0

INTERCALA (X, a, q, b)

0 $B[a..b]$ é um vetor auxiliar

1 para $i \leftarrow a$ até q faça

2 $B[i] \leftarrow X[i]$

3 para $j \leftarrow q+1$ até b faça

4 $B[b+q+1-j] \leftarrow X[j]$

5 $i \leftarrow a$

6 $j \leftarrow b$

7 contador $\leftarrow 0$

8 para $k \leftarrow a$ até b faça

9 se $B[i] \leq B[j]$

10 então $X[k] \leftarrow B[i]$

11 $i \leftarrow i + 1$

12 senão $X[k] \leftarrow B[j]$

13 $j \leftarrow j - 1$

14 contador $\leftarrow \text{contador} + (q - i + 1)$

15 retorna contador

4. Descreva um algoritmo que, dados inteiros n e k , juntamente com k listas ordenadas que em conjunto tenham n registros, produza uma única lista ordenada contendo todos os registros dessas listas (isto é, faça uma *intercalação*). O seu algoritmo deve ter complexidade $O(n \lg k)$. Note que isto se transforma em $O(n \lg n)$ no caso de n listas de 1 elemento, e em $O(n)$ se só houver duas listas (no total com n elementos).

Esse algoritmo pode ser construído de forma que recursivamente o número de vetores é dividido por dois e em cada recursão dois vetores são intercalados. Como o número de vetores é dividido por dois serão $\Theta(\lg k)$ iterações. É possível adaptar o algoritmo *mergesort* e utilizar a função *intercala* (mostrada em aula) também adaptada.

```

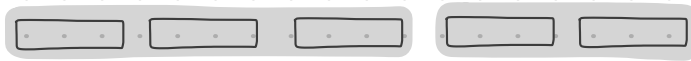
INTERCALA (A, p, q, r)
0  ▷ B[p..r] é um vetor auxiliar
1  para i ← p até q faça
2      B[i] ← A[i]
3  para j ← q + 1 até r faça
4      B[r + q + 1 - j] ← A[j]
5  i ← p
6  j ← r
7  para k ← p até r faça
8      se B[i] ≤ B[j] e i ≤ q      ▷ alteração
9          então A[k] ← B[i]
10         i ← i + 1
11     senão A[k] ← B[j]
12         j ← j - 1

```

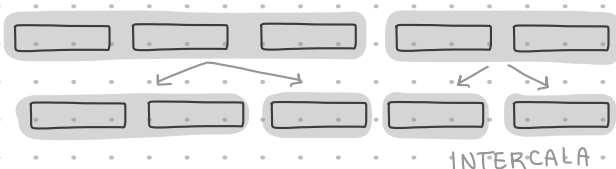
Assim, como a função *intercala* consome tempo $\Theta(n)$ e as recursões consomem $\Theta(\lg k)$, o algoritmo completo terá complexidade $O(n \lg k)$.

Simulação do algoritmo proposto para $k = 5$.

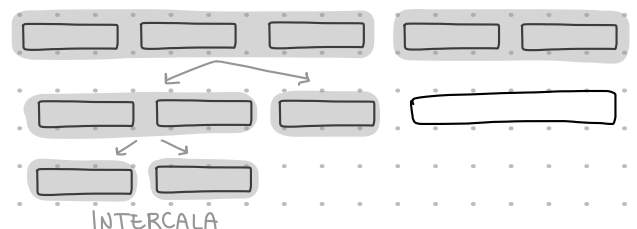
1. Inicialmente dividimos as 5 listas com o pivô sendo igual a $\lceil k/2 \rceil$



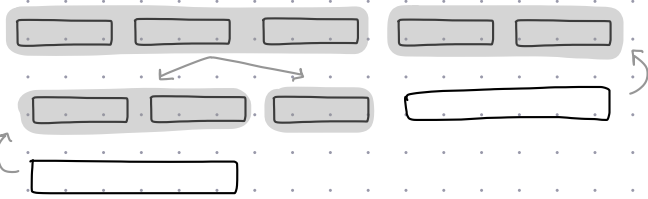
2. recursivamente temos



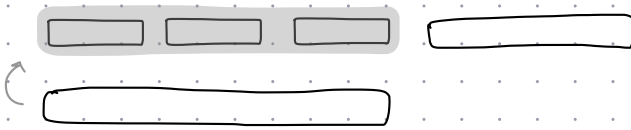
3. como na esquerda chegamos no momento da recursão no qual não há mais que 1 lista no grupo a função *intercala* é chamada, e na direita a recursão continua.



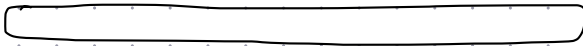
4. seguindo a lógica apresentada



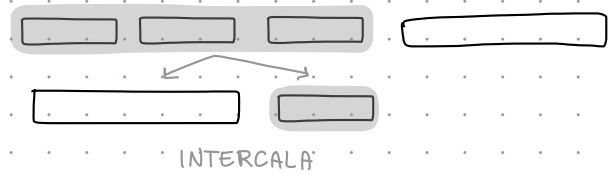
6.



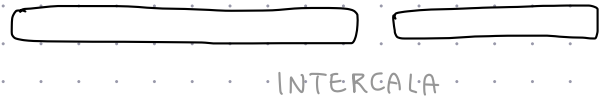
8. e recursivamente temos o vetor final



5.



7.



Pseudo código:

1 INTERCALACAO (n, K, L_1, \dots, L_i) (lista L_i com i sendo o n° de listas em cada recursão indo de 1 a K)

2 se o número de listas for maior que 1

3 $K \leftarrow n^\circ$ de listas atual

4 então $pivô \leftarrow \lceil K/2 \rceil$

(dividi-se a listas em dois grupos de acordo com um pivô arredondando para cima)

5 $A \leftarrow \text{INTERCALACAO}(n, K, L_1, \dots, L_{pivô})$

6 $B \leftarrow \text{INTERCALACAO}(n, L_{pivô+1}, \dots, L_i)$

7 $D \leftarrow \text{INTERCALA}(n, A, B)$

8 retorna D

1 INTERCALA (n, A, B)

2 para $i \leftarrow 0$ até $n-1$ faça

3 $C[i] \leftarrow A[i]$

4 $C[2n-1-i] \leftarrow B[i]$ o vetor C é junção de A com o inverso de B

5 $i \leftarrow 0$

6 $j \leftarrow 2n-1$

7 para $k \leftarrow 0$ até $2n-1$ faça

8 se $C[i] \leq C[j]$

$D[0, \dots, 2n-1]$ é um vetor auxiliar

9 então $D[k] \leftarrow C[i]$

10 $i \leftarrow i+1$

11 senão $D[k] \leftarrow C[j]$

12 $j \leftarrow j-1$

13 retorna D