Exercício 1

(a). $\boldsymbol{T(n) := 2T(n/2) + Cn^2,\ \forall n \geq 2,\ S := \{2^k : k \in \mathbb{N}\}}$

Vamos desenvolver a expressão dada, buscando uma fórmula não recursiva equivalente a ela.

$$\begin{aligned}
T(n) &= 2\,T(n/2) + Cn^2 \\
&= 2 \cdot \Big(2\,T(n/4) + C(n/2)^2\Big) + Cn^2 \\
&= 4\,T(n/4) + 2C(n/2)^2 + Cn^2 \\
&= 4 \cdot \Big(2\,T(n/8) + C(n/4)^2\Big) + 2C(n/2)^2 + Cn^2 \\
&= 8\,T(n/8) + 4C(n/4)^2 + 2C(n/2)^2 + Cn^2 \\
&\vdots \\
&= 2^k\,T(n/2^k) + 2^{k-1}C(n/2^{k-1})^2 + \cdots + 2C(n/2)^2 + Cn^2 \\
&= 2^k \cdot 1 + Cn^2 \cdot \Big(\frac{1}{2^{k-1}} + \frac{1}{2^{k-2}} + \cdots + \frac{1}{2} + 1\Big) \\
&= 2^k + Cn^2 \cdot \Big(2 \cdot (1 - \frac{1}{2^k})\Big) \qquad \text{[soma geométrica]} \\
&= n + 2Cn^2 \cdot (1 - \frac{1}{n}). \\
&= 2Cn^2 - 2Cn + n.
\end{aligned}$$

Agora, precisamos mostrar que a fórmula que encontramos é correta. Fazemos isso mostrando, por indução sobre k, que ela vale para todos os valores do conjunto S.

Para $k = 1$ (ou seja, $n = 2$), temos $T(1) = 2T(2/2) + 4C = 4C + 2$. De fato, nossa fórmula nos dá $2C \cdot 4 - 2C \cdot 2 + 2 = 8C - 4 + 2 = 4C + 2$.

Agora, vamos supor que $T(2^k) = 2C \cdot (2^k)^2 - 2C \cdot 2^k + 2^k$.

Para $T(2^{k+1})$, vale: $T(2^{k+1}) = 2T(2^k) + C(2^{k+1})^2$. Utilizando nossa hipótese da indução, temos:

$$\begin{aligned}
T(2^{k+1}) &= 2\Big(2C \cdot (2^k)^2 - 2C \cdot 2^k + 2^k\Big) + C(2^{k+1})^2 \\
&= C \cdot (2^{k+1})^2 - 2C \cdot 2^{k+1} + 2^{k+1} + C(2^{k+1})^2 \\
&= 2C \cdot (2^{k+1})^2 - 2C \cdot 2^{k+1} + 2^{k+1},
\end{aligned}$$

como queríamos demonstrar.

(b). $\boldsymbol{T(n) := 8T(n/2) + Cn^2,\ \forall n \geq 2,\ S := \{2^k : k \in \mathbb{N}\}}$

Vamos desenvolver a expressão dada, buscando uma fórmula não recursiva equivalente a ela.

$$\begin{aligned}
T(n) &= 8\,T(n/2) + Cn^2 \\
&= 8 \cdot \Big(8\,T(n/4) + C(n/2)^2\Big) + Cn^2 \\
&= 64\,T(n/4) + 8C(n/2)^2 + Cn^2 \\
&= 64 \cdot \Big(8\,T(n/8) + C(n/4)^2\Big) + 8C(n/2)^2 + Cn^2 \\
&= 512\,T(n/8) + 64C(n/4)^2 + 8C(n/2)^2 + Cn^2 \\
&\vdots \\
&= 8^k\,T(n/2^k) + 8^{k-1}C(n/2^{k-1})^2 + \cdots + 8C(n/2)^2 + Cn^2.
\end{aligned}$$

Substituindo $T(1) = 1$ e desenvolvendo a soma geométrica, temos:

$$\begin{aligned}
T(n) &= 8^k \cdot 1 + Cn^2 \cdot \Big(\frac{8^{k-1}}{2^{2(k-1)}} + \cdots + \frac{8}{2^2} + 1\Big) \\
&= 8^k + Cn^2 \cdot \sum_{i=0}^{k-1} 2^i \\
&= 8^k + Cn^2 \cdot (2^k - 1) \\
&= n^3 + Cn^2(n-1) \\
&= (C+1)n^3 - Cn^2.
\end{aligned}$$

Agora, vamos mostrar que a fórmula é correta por indução sobre k, $\forall k \in S$.

Para $k = 1$ (ou seja, $n = 2$) temos

$$T(2) = 8T(1) + C \cdot 2^2 = 8 \cdot 1 + 4C = 8 + 4C.$$

Tal valor também é devolvido por nossa fórmula:

$$T(2) = (C+1) \cdot 2^3 - C \cdot 2^2 = 8 + 4C.$$

Agora, consideremos a hipótese $T(2^k) = (C+1)2^{3k} - C2^{2k}$. Nesse caso, para $T(2^{k+1})$, podemos desenvolver a fórmula recursiva da seguinte maneira:

$$\begin{aligned}
T(2^{k+1}) &= 8T(2^k) + C(2^{k+1})^2 \\
&= 8\big((C+1)2^{3k} - C2^{2k}\big) + C \cdot 2^{2(k+1)} \\
&= (C+1)2^{3(k+1)} - C2^{2(k+1)},
\end{aligned}$$

e assim concluir nossa prova indutiva.

(c). $\boldsymbol{T(n) := 7T(n/3) + Cn^2,\ \forall n \geq 2,\ S := \{3^k : k \in \mathbb{N}\}}$

Vamos desenvolver a expressão dada, buscando uma fórmula não recursiva equivalente a ela.

$$\begin{aligned}
T(n) &= 7\,T(n/3) + Cn^2 \\
&= 7 \cdot \left(7\,T(n/3^2) + C(n/3)^2\right) + Cn^2 \\
&= 7^2\,T(n/3^2) + 7C(n/3)^2 + Cn^2 \\
&= 7^2 \cdot \left(7\,T(n/3^3) + C(n/3^2)^2\right) + 7C(n/3)^2 + Cn^2 \\
&= 7^3\,T(n/3^3) + 7^2C(n/3^2)^2 + 7C(n/3)^2 + Cn^2 \\
&\ \vdots \\
&= 7^k\,T(n/3^k) + Cn^2 \cdot \sum_{i=0}^{k-1} 7^i \left(\frac{1}{3^i}\right)^2.
\end{aligned}$$

Substituindo $T(1) = 1$ e simplificando a soma, obtemos

$$\begin{aligned}
T(n) &= 7^k + Cn^2 \sum_{i=0}^{k-1} \frac{7^i}{3^{2i}} \\
&= 7^k + Cn^2 \sum_{i=0}^{k-1} \left(\frac{7}{9}\right)^i \\
&= 7^k + Cn^2 \cdot \frac{1 - (7/9)^k}{1 - 7/9} \\
&= 7^k + Cn^2 \cdot \frac{1 - (7/9)^k}{2/9} \\
&= 7^k + \frac{9C}{2}\,n^2 \left(1 - \left(\frac{7}{9}\right)^k\right).
\end{aligned}$$

Agora usamos $n = 3^k$, de modo que $(7/9)^k = \dfrac{7^k}{3^{2k}} = \dfrac{7^k}{n^2}$ e escrevemos a forma fechada em termos de n:

$$\begin{aligned}
T(n) &= 7^k + \frac{9C}{2}\,n^2 - \frac{9C}{2}\,7^k \\
&= \left(1 - \frac{9C}{2}\right) 7^k + \frac{9C}{2}\,n^2 \\
&= \left(1 - \frac{9C}{2}\right) n^{\log_3 7} + \frac{9C}{2}\,n^2.
\end{aligned}$$

Agora, provamos que a fórmula é correta por indução sobre k, no conjunto S.

Para $k = 1$ (ou seja, $n = 3$):

$$T(3) = 7T(1) + C \cdot 3^2 = 7 \cdot 1 + 9C = 7 + 9C.$$

Com a fórmula que desenvolvemos, obtemos o mesmo resultado

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

Agora, suponha que para k valha

$$T(3^k) = \left(1 - \frac{9C}{2}\right) 7^k + \frac{9C}{2} \cdot 3^{2k}.$$

Para $T(3^{k+1})$, temos

$$\begin{aligned}
T(3^{k+1}) &= 7T(3^k) + C(3^{k+1})^2 \\
&= 7\left[\left(1 - \frac{9C}{2}\right) 7^k + \frac{9C}{2} \cdot 3^{2k}\right] + C \cdot 3^{2(k+1)} \\
&= \left(1 - \frac{9C}{2}\right) 7^{k+1} + \frac{63C}{2} \cdot 3^{2k} + 9C \cdot 3^{2k} \\
&= \left(1 - \frac{9C}{2}\right) 7^{k+1} + \frac{9C}{2} \cdot 3^{2(k+1)},
\end{aligned}$$

isto é, exatamente o valor de nossa fórmula para $k+1$, como queríamos mostrar.

(d). $\boldsymbol{T(n) := 2T(n/2) + Cn^3,\ \forall n \geq 2,\ S := \{2^k : k \in \mathbb{N}\}}$

Vamos desenvolver a expressão dada, buscando uma fórmula não recursiva equivalente a ela.

$$\begin{aligned}
T(n) &= 2\,T(n/2) + Cn^3 \\
&= 2 \cdot \left(2\,T(n/4) + C(n/2)^3\right) + Cn^3 \\
&= 4\,T(n/4) + 2C(n/2)^3 + Cn^3 \\
&= 4 \cdot \left(2\,T(n/8) + C(n/4)^3\right) + 2C(n/2)^3 + Cn^3 \\
&= 8\,T(n/8) + 4C(n/4)^3 + 2C(n/2)^3 + Cn^3 \\
&\vdots \\
&= 2^k\,T(n/2^k) + \sum_{i=0}^{k-1} 2^i C \left(\frac{n}{2^i}\right)^3.
\end{aligned}$$

Substituindo $T(1) = 1$ e simplificando a soma:

$$\begin{aligned}
T(n) &= 2^k + C\sum_{i=0}^{k-1} \frac{2^i n^3}{2^{3i}} \\
&= 2^k + Cn^3 \sum_{i=0}^{k-1} 2^{-2i} \\
&= 2^k + Cn^3 \cdot \frac{1 - 2^{-2k}}{1 - 2^{-2}} \\
&= 2^k + \frac{4C}{3} n^3 \left(1 - \frac{1}{4^k}\right).
\end{aligned}$$

Como $n = 2^k$, temos $1/4^k = 1/n^2$, então:

$$\begin{aligned}
T(n) &= n + \frac{4C}{3} n^3 \left(1 - \frac{1}{n^2}\right) \\
&= n + \frac{4C}{3}(n^3 - n) \\
&= \frac{4C}{3} n^3 + \left(1 - \frac{4C}{3}\right) n.
\end{aligned}$$

Agora, provamos que a fórmula é correta por indução sobre k, no conjunto S.

Para $k = 1$ (ou $n = 2$), temos

$$T(2) = 2T(1) + C \cdot 2^3 = 2 \cdot 1 + 8C = 2 + 8C.$$

E pela fórmula que desenvolvemos:

$$T(2) = \frac{4C}{3} \cdot 8 + \left(1 - \frac{4C}{3}\right) \cdot 2 = \frac{32C}{3} + 2 - \frac{8C}{3} = 2 + 8C.$$

Agora, suponha que $T(2^k) = \frac{4C}{3}2^{3k} + \left(1 - \frac{4C}{3}\right)2^k$. E então, para $T(2^{k+1})$:

$$\begin{aligned}
T(2^{k+1}) &= 2T(2^k) + C(2^{k+1})^3 \\
&= 2\left[\frac{4C}{3}2^{3k} + \left(1 - \frac{4C}{3}\right)2^k\right] + C \cdot 2^{3(k+1)} \\
&= \frac{8C}{3}2^{3k} + 2\left(1 - \frac{4C}{3}\right)2^k + 8C2^{3k} \\
&= \frac{32C}{3}2^{3k} + \left(1 - \frac{4C}{3}\right)2^{k+1} \\
&= \frac{4C}{3}2^{3(k+1)} + \left(1 - \frac{4C}{3}\right)2^{k+1},
\end{aligned}$$

concluindo o passo indutivo e a demonstração.

(e). $T(n) := 9T(n/10) + Cn,\ \forall n \geq 2,\ S := \{10^k : k \in \mathbb{N}\}$

Vamos desenvolver a expressão dada, buscando uma fórmula não recursiva equivalente a ela.

$$\begin{aligned}
T(n) &= 9\,T(n/10) + Cn \\
&= 9 \cdot \Big(9\,T(n/10^2) + C(n/10)\Big) + Cn \\
&= 9^2\,T(n/10^2) + 9C(n/10) + Cn \\
&= 9^2 \cdot \Big(9\,T(n/10^3) + C(n/10^2)\Big) + 9C(n/10) + Cn \\
&= 9^3\,T(n/10^3) + 9^2C(n/10^2) + 9C(n/10) + Cn \\
&\vdots \\
&= 9^k\,T(n/10^k) + Cn \cdot \sum_{i=0}^{k-1} \frac{9^i}{10^i}.
\end{aligned}$$

Substituindo $T(1) = 1$ e simplificando a soma geométrica:

$$\begin{aligned}
T(n) &= 9^k + Cn \sum_{i=0}^{k-1} \left(\frac{9}{10}\right)^i \\
&= 9^k + Cn \cdot \frac{1 - (9/10)^k}{1 - 9/10} \\
&= 9^k + 10Cn \left(1 - \left(\frac{9}{10}\right)^k\right).
\end{aligned}$$

Como $n = 10^k$, temos $(9/10)^k = 9^k/10^k = 9^k/n$, então:

$$\begin{aligned}
T(n) &= 9^k + 10Cn\left(1 - \frac{9^k}{n}\right) \\
&= 9^k + 10Cn - 10C \cdot 9^k \\
&= (1 - 10C)9^k + 10Cn \\
&= (1 - 10C)n^{\log_{10} 9} + 10Cn.
\end{aligned}$$

Agora, provamos que a fórmula é correta por indução sobre k, no conjunto S.

Para $k = 1$, $n = 10$:

$$T(10) = 9T(1) + C \cdot 10 = 9 \cdot 1 + 10C = 9 + 10C.$$

Pela fórmula:

$$T(10) = (1 - 10C) \cdot 9 + 10C \cdot 10 = 9 - 90C + 100C = 9 + 10C.$$

Agora, suponha que $T(10^k) = (1 - 10C)9^k + 10C \cdot 10^k$. Para $T(10^{k+1})$, então:

$$\begin{aligned}
T(10^{k+1}) &= 9T(10^k) + C \cdot 10^{k+1} \\
&= 9\big((1 - 10C)9^k + 10C \cdot 10^k\big) + 10C \cdot 10^k \\
&= (1 - 10C)9^{k+1} + 90C \cdot 10^k + 10C \cdot 10^k - 90C \cdot 10^k? \\
&= (1 - 10C)9^{k+1} + 10C \cdot 10^{k+1},
\end{aligned}$$

concluindo o passo indutivo e a demonstração.

Exercício 2

O algoritmo abaixo funciona de maneira muito semelhante ao MERGE-SORT original, com a sutil diferença de que nosso MERGE-COUNT faz, além de intercalar e ordenar os sub-arrays, armazena a quantidade de inversões presentes entre eles incrementando uma variável sempre que uma inversão é encontrada.

A entrada do algoritmo é o vetor A, uma cópia do vetor $X[1..n]$ recebido, com os parâmetros $A, 1, A$.length passados para a função recursiva COUNT-INVERSIONS, e a saída é o número de inversões presentes em A.

```
COUNT-INVERSIONS(A, p, r)
if p ≥ r then
    return 0
end if
q ← ⌊(p + r)/2⌋
leftInv ← COUNT-INVERSIONS(A, p, q)
rightInv ← COUNT-INVERSIONS(A, q + 1, r)
crossInv ← MERGE-COUNT(A, p, q, r)
return leftInv + rightInv + crossInv
```

```
MERGE-COUNT(A, p, q, r)
n1 ← q − p + 1
n2 ← r − q
L[1..n1] ← A[p..q]
R[1..n2] ← A[q + 1..r]
i ← 1, j ← 1, inv ← 0
for k = p to r do
    if i ≤ n1 and (j > n2 or L[i] ≤ R[j]) then
        A[k] ← L[i]
        i ← i + 1
    else
        A[k] ← R[j]
        inv ← inv +(n1 − i + 1)
        j ← j + 1
    end if
end for
return inv
```

Vamos mostrar a corretude do algoritmo usando indução forte sobre o tamanho do vetor.

Se o vetor tem um único elemento, o algoritmo retorna zero, o que é correto, já que um vetor de um elemento não tem inversões.

Agora, suponha que o algoritmo funciona para qualquer vetor de tamanho menor que n, e então, considere um vetor de tamanho $n = r - p + 1 > 1$. O algoritmo divide A em dois, e realiza uma chamada recursiva em cada metade. Por hipótese, ele devolve o número correto de inversões em cada metade.

Em seguida, o algoritmo faz a contagem de inversões cruzadas, isto é, inversões entre as duas metades. É importante lembrar que tais metades estão ordenadas, o que também faz parte da hipótese. Então, durante a intercalação dessas metades, cada vez que um elemento do sub-array da direita é menor que um elemento do sub-array da esquerda, significa que todos os elementos restantes na esquerda também são maiores que esse elemento da direita, assim, a variável `inv` é incrementada pelo número de elementos no sub-array da direita.

Assim, como o algoritmo devolve o número de inversões cruzadas, somado ao número de inversões contidas nos sub-arrays, as quais, por hipótese, são corretas, concluímos sua corretude.