1)

b) $T(n) = 8T(\lfloor n/2 \rfloor) + c_2 n^2$

$$
\begin{aligned}
T(n) &= 2^3\, T(n/2) + c_2 n^2 \\
&= 2^6\, T(n/2^2) + 3 c_2 n^2 \\
&= 2^9\, T(n/2^3) + 7 c_2 n^2 \\
&= 2^{12}\, T(n/2^4) + 15 c_2 n^2 \\
&= 2^{3k}\, T(n/2^k) + (2^k - 1) c_2 n^2 \quad \text{se } n = 2^k
\end{aligned}
$$

$$\Rightarrow c\, n^3 + c_2 n^3 - c_2 n^2$$

$$
T(n) = \begin{cases} c_1 & , n = 1 \\ 8T(n/2) + c_2 n^2 & , n > 1 \end{cases}
$$

vale $T(n) = c\, n^3 + c_2 n^3 - c_2 n^2$ p/ $n = 2^k$

$K = 0 \Rightarrow T(1) = c + c_2 - c_2 \Rightarrow c = c$ //

$k > 1 \Rightarrow T(2^k) = 8T(2^{k-1}) + c_2 2^{2k}$ por H.I:

$$
\begin{aligned}
&= 8(c\, 2^{3k-3} + c_2 2^{3k-3} - c_2 2^{2k-2}) + c_2 2^{2k} \\
&= c\, 2^{3k} + c_2 2^{3k} - c_2 2^{2k+1} + c_2 2^{2k} \\
&= c\, 2^{3k} + c_2 2^{3k} - c_2 2^{2k} \\
&= c\, n^3 + c_2 n^3 - c_2 n^2 \text{ //}
\end{aligned}
$$

1 e)

$$T(n) = \begin{cases} c_1 & n = 1 \\ T(\lfloor 9n/10 \rfloor) + c_2 n & n > 1 \end{cases}$$

$$\begin{aligned} T(n) &= T(9n/10) + c_2 n \\ &= T(9^2 n/10^2) + c_2 \, 9n/10 + c_2 n \\ &= T(9^3 n/10^3) + c_2 n (9^2/10^2 + 9/10 + 1) \\ &= T(9^K n/10^K) + c_2 n (9^K/10^K + \dots + 1) \\ &= T(9^k n/10^k) - 10 c_2 \, n ((9/10)^K - 1) \\ &= c_1 + 10 c_2 n - 10 c_2 \quad \text{se } n = (10/9)^K \end{aligned}$$

$T(n)$ vale $c_1 + 10 c_2 n - 10$ p/ $n = \left(\frac{10}{9}\right)^K$

$K = 0 \Rightarrow T(1) = c_1 + 10c_2 - 10c_2 \Rightarrow c_1 = c_1$ //

$$\begin{aligned} K > 1 \Rightarrow T\left(\left(\tfrac{10}{9}\right)^K\right) &= T\left(\left(\tfrac{10}{9}\right)^{K-1}\right) + c_2 \left(\tfrac{10}{9}\right)^K \\ &= c_1 + 10 c_2 \left(\tfrac{10}{9}\right)^{K-1} - 10 c_2 + c_2 \left(\tfrac{10}{9}\right)^K \\ &= c_1 + 10 c_2 \left(\tfrac{10}{9}\right)^K - 10 c_2 \\ &= c_1 + 10 c_2 n - 10 c_2 \ // \end{aligned}$$

```
Mergesort(A, i, f)
inv <- 0
se i < f
    m <- ⌊(i+f)/2⌋
    Mergesort(A, i, m)
    Mergesort(A, m+1, f)
    inv += Intercala(A, i, m, f)
return inv
```

Intercala(A, i, m, f)

```
inv <- 0
B[i ... f]
para u <- i até m faça
    B[u] <- A[u]
para v <- m + 1 até f faça
    B[f + m + 1 - v] <- A[v]
u <- i
v <- f
para K <- i até f faça
    se B[u] ≤ B[v]
        A[K] <- B[u]
        u ++
    senão
        A[K] <- B[v]
        v --
        inv += m - u + 1
return inv
```

Analisando o Intercala(), vemos que seu consumo de tempo é $O(n)$ porque cada linha é executada, no máximo, n vezes se $n = f - i + 1$:

1 e 2: $O(1)$; 3 e 4: $O(n)$; 5 e 6: $O(n)$; 7 e 8: $O(1)$;
9: $O(n)$; 10-16: $O(n)$; 17: $O(1) = 6N + 5 = O(n)$

Analisando o Mergesort(), obtemos a recorrência:

$$T(n) = \begin{cases} c_1 & \text{se } n = 1 \\ T(\lfloor \frac{n}{2} \rfloor) + T(\lceil \frac{n}{2} \rceil) + O(n) & \text{c.c} \end{cases}$$

$$\begin{aligned} T(n) &= 2T(\tfrac{n}{2}) + c_2 n \\ &= 4T(\tfrac{n}{4}) + 2c_2 n \\ &= 8T(\tfrac{n}{8}) + 3c_2 n \\ &= 2^k T(\tfrac{n}{2^k}) + kc_2 n \\ &= 2^k c_1 + kc_2 n \quad \text{se } n = 2^k \\ &= c_1 n + c_2 n \log n \end{aligned}$$

$T(n)$ vale $c_1 n + c_2 n \log n$ para $n = 2^K$

$K = 0 \Rightarrow T(1) = c_1 \Rightarrow c_1 = c_1$ //

$$
\begin{aligned}
K > 1 \Rightarrow T(2^K) &= 2T(2^{K-1}) + c_2 2^K \\
&= 2c_1 2^{K-1} + 2c_2 2^{K-1} \log 2^{K-1} + c_2 2^K \\
&= c_1 2^K + c_2 2^K (K-1) + c_2 2^K \\
&= c_1 n + c_2 n \log n \text{ //}
\end{aligned}
$$