Exercise 2.3-1

\[
\begin{array}{cccc}
3 & 9 & 26 & 38 \\
26 & 41 & 52 & 49 \\
3 & 41 & 52 & 26 \\
3 & 41 & 52 & 26 \\
3 & 41 & 52 & 26 \\
\end{array}
\]

Exercise D.

C(0) = 0 = d·0, so the result holds when \( n = 0 \).

Let \( n \geq 1 \), and assume the result holds for all \( i \) with \( i < n \). \( C(n) = d + C(k) + C(n-1-k) \) where \( 0 \leq k \leq n-1 \). Note \( n-k-1 \leq n-1 \). By the inductive hypothesis,

\[
C(n) = d + dk + d(n-k-1) = d(1 + k + (n-k-1)) = dn,
\]

so the result also holds for \( n \). By induction it holds for all nonnegative integers.

Exercise E.

Let \( L, M, \) and \( R \) be sorted arrays of length \( n/3 \) (possibly \( \lfloor n/3 \rfloor \) or \( \lceil n/3 \rceil \), so the sum of the lengths is \( n \)). For simplicity, assume \( n \) is a power of 3, so always \( L, M, \) and \( R \) have length \( n/3 \). Assume that each array has an extra element \( \infty \) at the end. We can merge \( L, M, \) and \( R \) into a single sorted array \( A \) of length \( n \) using the algorithm below. Here \( i, j, \) and \( k \) represent the positions of the current elements in \( L, M, \) and \( R \) respectively; and \( x \) represents the smallest element not yet merged from \( M \) or \( R \), provided \( xValid \) is true. As usual, indentation indicates nesting of blocks.
$i = 1; \; \; j = 1; \; \; k = 1;$
$xValid = \text{false};$
\begin{align*}
\text{for ( } q = 1, 2, \ldots, n \text{ )} \quad & \\
& \text{if ( not } xValid \text{ )} \\
& \quad \text{if ( } M[j] \leq R[k] \text{ )} \quad (\ast) \\
& \quad \quad x = M[j]; \\
& \quad \quad j = j + 1; \\
& \quad \text{else} \\
& \quad \quad x = R[k]; \\
& \quad \quad k = k + 1; \\
& \quad xValid = \text{true}; \\
& \text{if ( } L[i] \leq x \text{ )} \quad (\ast\ast) \\
& \quad A[q] = L[i]; \\
& \quad i = i + 1; \\
& \text{else} \\
& \quad A[q] = x; \\
& \quad xValid = \text{false};
\end{align*}

Comparisons are performed in the lines (\ast) and (\ast\ast). The comparison in line (\ast\ast) is performed on each pass through the loop — a total of $n$ times. The comparison on line (\ast) is always performed on the first pass ($q = 1$). On the remaining passes, it is performed if the element merged to $A$ on the previous pass came from $M$ or $R$, but not if it came from $L$. Thus the total number of comparisons in line (\ast) is

\[ n - (\text{number of elements merged from } L \text{ on the first } n-1 \text{ passes}) \]
\[ = n - (n/3 \text{ or } n/3-1) \]
\[ = 2/3 \; n \text{ or } 2/3 \; n + 1 \]

times. The total number of comparisons performed by the algorithm is $5/3 \; n$ or $5/3 \; n + 1$.

Exercise F.

$C(n) = 3C(n/3) + 5/3n$, $C(1) = 0$. We assume $n = 3^k$, so $k = \log_3(n)$.

\[
C(n) = 3C(n/3) + 5/3n \\
= 3(3C(n/3^2) + (5/3)(n/3)) + 5/3n \\
= 3^2C(n/3^2) + 2(5/3n) \\
= 3^2(3C(n/3^3) + (5/3)(n/3^2)) + 2(5/3n) \\
= 3^3C(n/3^3) + 3(5/3n) \\
\vdots \\
= 3^kC(n/3^k) + k(5/3n) \\
= nC(1) + 5/3 \; n \log_3(n) \\
= 5/3 \; n \log_3(n)
\]

The exact solution when $n$ is a power of 3 is $C(n) = 5/3 \; n \log_3(n) \approx 1.052 \; n \lg(n)$.

By contrast, ordinary (2-way) mergesort uses approximately $n \lg(n)$ comparisons.
Exercise G  In each part, we assume \( n = 2^k \), so \( k = \log(n) \).

a)  \( C(n) = C(n/2) + 2n + 3 \)
    = \( (C(n/2^2) + 2(n/2) + 3) + 2n + 3 \)
    = \( C(n/2^2) + 2n/2 + n + 2 \cdot 3 \)
    = \( C(n/2^3) + 2(n/2^2) + 3) + 2(n/2 + n) + 2 \cdot 3 \)
    = \( C(n/2^3) + 2(n/2^2 + n/2 + n) + 3 \cdot 3 \)
    
    .
    
    = \( C(n/2^k) + 2(n/2^k−1 + \ldots + n/2^2 + n/2 + n) + k \cdot 3 \)
    = \( C(1) + 2n(1/2^k−1 + \ldots + 1/2^2 + 1/2 + 1) + k \cdot 3 \)
    = \( 1 + 2n(2−1/2^k−1) + 3 \log(n) \)
    = \( 1 + 4n - 4 + 3 \log(n) \)  
    (since \( n/2^k−1 = 2/2^k−1 = 2 \))
    = \( 4n - 3 + 3 \log(n) \)

b)  \( C(n) = 2C(n/2) + n \log(n) \)
    = \( 2(2C(n/2^2) + n/2 \cdot \log(n/2)) + n \log(n) \)
    = \( 2^2 C(n/2^2) + n(\log(n) - 1) + n \log(n) \)
    = \( 2^2 C(n/2^2) + 2n \log(n) - n \)
    = \( 2^3 (2C(n/2^3) + n^2 \cdot \log(n/2^2)) + 2n \log(n) - n \)
    = \( 2^3 C(n/2^3) + n(\log(n) - 2) + 2n \log(n) - n \)
    = \( 2^3 C(n/2^3) + 3n \log(n) - n(1+2) \)
    
    .
    
    = \( 2^k C(n/2^k) + k n \log(n) - n(1+2+\ldots+k−1) \)
    = \( nC(1) + n(\log(n))^2 - nk(k−1)/2 \)
    = \( n(\log(n))^2 - n \log(n)(\log(n)−1)/2 \)
    = \( n(\log(n))^2/2 + n(\log(n))/2 \)

c)  \( C(n) = 4C(n/2) + 3n^2 \)
    = \( 4(4C(n/2^2) + 3(n/2)^2) + 3n^2 \)
    = \( 4^2 C(n/2^2) + 4 \cdot 3(n/2)^2 + n^2 \)
    = \( 4^2 C(n/2^2) + 3(2n^2) \)
    = \( 4^2 (4C(n/2^3) + 3(n/2^2)^2) + 3(2n^2) \)
    = \( 4^3 C(n/2^3) + 4^2 \cdot 3(n/2^2)^2 + 3(2n^2) \)
    = \( 4^3 C(n/2^3) + 3(3n^2) \)
    
    .
    
    = \( 4^k C(n/2^k) + 3(kn^2) \)
    following pattern on lines (*) (**) (***)
    = \( 4^k C(1) + 3 \log(n)n^2 \)
    = \( 4^k 0 + 3n^2 \log(n) = 3n^2 \log(n) \)