Chapter 2, Problem 2.3: (a)

\[
T(n) = 3T(n/2) + O(n)
\]

\[
\leq 3T(n/2) + cn
\]

\[
\leq 3[3T(n/4) + cn/2] + cn = 3^2T(n/4) + 3/2cn + cn
\]

\[
\leq 3^2[3T(n/8) + cn/4] + 3/2cn + cn = 3^3T(n/8) + (3/2)^2cn + 3/2cn + cn
\]

\[
\ldots
\]

\[
\leq 3^kT(n/2^k) + [(3/2)^k + (3/2)^{k-1} + \ldots + 1]cn
\]

\[
= 3^kT(n/2^k) + 2[(3/2)^k - 1]cn
\]

Plugging in \( k = \log_2 n \), we have

\[
T(n) \leq 3^\log_2 n T(1) + 2[3^\log_2 n/n - 1]cn = O(\log_2 n^3).
\]

Chapter 2, Problem 2.3: (b)

\[
T(n) = T(n - 1) + O(1)
\]

\[
\leq T(n - 1) + c
\]

\[
\leq [T(n - 2) + c] + c = T(n - 2) + 2c
\]

\[
\ldots
\]

\[
\leq T(1) + (n - 1)c
\]

\[
= O(n).
\]

Chapter 2, Problem 2.12: Let \( L(n) \) be the number lines printed. We have

\[
L(n) = 2L(n/2) + 1
\]

\[
= 2[2L(n/4) + 1] + 1 = 4L(n/4) + 2 + 1
\]

\[
= 4[2L(n/8) + 1] + 2 + 1 = 8L(n/8) + 4 + 2 + 1
\]

\[
\ldots
\]

\[
= 2^kL(n/2^k) + [2^{k-1} + 2^{k-2} + \ldots + 1] = 2^kL(n/2^k) + 2^k - 1
\]

\[
= nL(1) + n - 1
\]

\[
= n - 1 = \Theta(n).
\]
Chapter 2, Problem 2.17: Algorithm:

function search(a[1\cdots n])
    i = a[n/2]
    if a[i] > i
        search(a[1\cdots (i - 1)])
    else if a[i] < i
        search(a[(i + 1)\cdots n])
    else
        return i

Time analysis: \( T(n) = T(n/2) + O(1) = O(\log n) \).

Problem 2: Let \( E \) be the expected number of iteration.
With probability 1/2, it takes edge \( a \to b \), which takes 1 iteration to reach \( b \).
With probability 1/2, it takes edge \( a \to c \), which will take 3 iterations to get back to \( a \), and then \( E \) iterations to reach \( b \).
Thus \( E = 1/2 \cdot 1 + 1/2 \cdot (3 + E) \), which works out to \( E = 4 \).