Problem 1.

a) \( T(n) = T(9n/10) + n \)

Case 3 of Master theorem: \( a = 1, b = 10/9, n^{\log_b a} = n^{\log_{10/9} 1} = n^0 = 1, f(n) = n \)

i) \( f(n) = \Omega(n^{\log_b a + \epsilon}) \) holds for \( \epsilon = 0.9 \).

ii) \( a f(n/b) = c f(n) \) holds for \( c = 0.99 \).

Thus \( T(n) = \Theta(n) \).

b) \( T(n) = 7T(n/3) + n^2 \)

Case 3 of Master theorem: \( a = 7, b = 3, n^{\log_b a} = n^{\log_3 7} = n^{1.7}, f(n) = n^2 \)

i) \( f(n) = \Omega(n^{\log_b a + \epsilon}) \) holds for \( \epsilon = 0.1 \).

ii) \( a f(n/b) = c f(n) \) holds for \( 7/9 \leq c < 1 \).

Thus \( T(n) = \Theta(n^2) \).

Problem 2.

Base case: for \( n=2 \) there is only one term in the summation:
\[
\sum_{k=1}^{2-1} \frac{1}{k(k+1)} = \frac{1}{1(1+1)} = \frac{1}{2} = 1 - \frac{1}{2} = 1 - 2/2.
\]

Inductive step: Assume (I.H.) that the proposition: \( \sum_{k=1}^{n-1} \frac{1}{k(k+1)} = 1 - \frac{1}{n} \) holds for \( n \).

Show that the proposition holds for \( n+1 \):
\[
\sum_{k=1}^{n+1-1} \frac{1}{k(k+1)} = \sum_{k=1}^{n-1} \frac{1}{k(k+1)} + \frac{1}{n(n+1)}
= 1 - \frac{1}{n} + \frac{1}{n(n+1)} \quad \text{I.H.}
= n(n+1) - (n+1) + 1
= \frac{n(n+1) - (n+1) + 1}{n(n+1)}
= \frac{n^2 + n - n - 1 + 1}{n(n+1)}
= \frac{n^2}{n(n+1)} = \frac{n}{n+1}
= 1 - \frac{1}{n+1} \quad Q.E.D.
\]
Problem 3.

Pseudo code for the MERGE procedure can be found on page 29 of CLRS.

Problem 4.

To solve the problem we will modify MERGE-SORT to return the number of inversion. We define the number of inversion in an array $A$ recursively as:

$$\text{# of inv in } A = \text{# of inv in } L + \text{# of inv in } R + \text{# of inv induced by merging } L \text{ and } R,$$

where $L$ and $R$ are sub-arrays of $A$.

$$\text{INV}(A, p, r)$$

$$\begin{align*}
\text{inv} & \leftarrow 0 \\
q & \leftarrow \text{floor}((p+r)/2) \\
\text{inv} & \leftarrow \text{inv} + \text{INV}(A, p, q) \\
\text{inv} & \leftarrow \text{inv} + \text{INV}(A, q+1, r) \\
\text{inv} & \leftarrow \text{inv} + \text{MERGE-INV}(A, p, q, r)
\end{align*}$$

We can compute the number of inversion induced by the merging of $L$ and $R$ by adding the following two lines to the pseudo code presented on page 29 of CLRS.

$$\begin{align*}
0 & \quad \text{inv} \leftarrow 0 \\
18 & \quad \text{inv} \leftarrow \text{inv} + n_{1-i+1}
\end{align*}$$

Problem 5.

Assuming we have $n$ bills and $m$ checks:

1. Sort the bills by the telephone numbers. This takes $O(n \log n)$ time. Assuming that the phone numbers are say 10 digit long, we can even apply radix sort and sort the bills in $O(n)$ time.
2. Sort the checks by the telephone numbers. This takes $O(m \log m)$ time. Assuming that the phone numbers are say 10 digit long, we can even apply radix sort and sort the checks in $O(m)$ time.
3. Compare elements in the two sorted arrays as follows:

$$\begin{align*}
i & \leftarrow 1; j \leftarrow 1 \\
\text{while } (i \leq n \text{ and } j \leq m) \\
\quad \text{if } \text{BILLS}[i] \neq \text{CHECKS}[j] \\
\quad \quad \text{print BILLS}[i]; i \leftarrow i + 1 \\
\quad \text{else} \\
\quad \quad i \leftarrow i + 1; j \leftarrow j + 1
\end{align*}$$

print the remaining bills if any

Depending on the sort applied, the worst-case runtime complexity is either $\max \{\Theta(n \log n), \Theta(m \log m)\}$ or $\max \{\Theta(n), \Theta(m)\}$. 