Solution 10

3.1

02. a) Lack finiteness.(The **while** loop lasts forerver.)

    b) Lack effectiveness.(When $n = 0$, $m := 1/n$ can not be computed.)

    c) Lack definiteness.(i is not be definited.)

    d) Lack definiteness.(It isn't determined.)

56. Each selection does not exceed the maximum number of current amounts.

    a) Three quarters, leaving 12 cents. One dime, leaving 2 cents. Two pennies.

    b) One quarter, leaving 24 cents. Two dimes, leaving 4 cents. Four pennies.

    c) Three quarters, leaving 24 cents. Two dimes, leaving 4 cents. Four pennies.

    d) One quarter, leaving 8 cents. One nickel, leaving 3 cents. Three pennies.

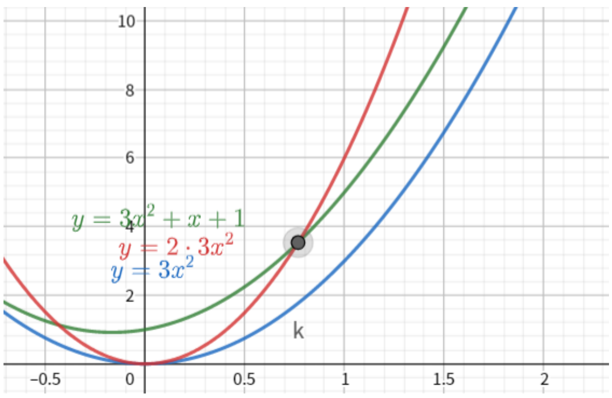
3.2

34. a)

$$
\begin{aligned}
&x > 0, 3x^2 \leqslant 3x^2 + x + 1 \\
&x > 1, 3x^2 + x + 1 \leqslant 2 \cdot 3x^2 \\
&1 \cdot 3x^2 \leqslant 3x^2 + x + 1 \leqslant 2 \cdot 3x^2
\end{aligned}
$$

    So, $C_1 = 1, C_2 = 2, k = 1$

    b) $k(\dfrac{\sqrt{13}+1}{6}, \dfrac{\sqrt{13}+7}{3})$



73.

$$
\begin{aligned}
&\because n \leqslant (n-k)(k+1) \\
&\therefore n^n \leqslant n((n-1)\cdot 2)\cdot((n-2)\cdot 3)\cdots(2\cdot(n-1))\cdot n = (n!)^2 \\
&\therefore \log n! \leqslant n\log n \leqslant 2\log n!
\end{aligned}
$$

So, $n\log n$ is $O(\log n!)$.

74. By 73., $n\log n$ is $O(\log n!)$.

$$
\begin{aligned}
&\because n! \leqslant n^n \\
&\therefore \log n! \leqslant n\log n
\end{aligned}
$$

So, $\log n!$ is $O(n\log n)$.
So, $\log n!$ is $\Theta(n\log n)$.

3.3

40. **procedure** change($c_1, c_2, \ldots, c_r$: coin denominations; n: a positive integer)
for $i := 1$ **to** r
  $d_i := 0$
  while $n \geqslant c_i$
    $d_i := d_i + 1$
    $n := n - c_i$

The algorithm uses comparison of denominations of coins which is positive integer. Any positive integer is $O(n)$. Each comparison takes some time, and there are at most n comparisons, since every comparison all decrease the number. Therefore, the greedy algorithm has $O(n)$ complexity measured in terms of comparison needed.