MAT2440, Quiz8, Spring2025

ID:______________________________ Name:______________________________

1. Arrange the functions $\sqrt{n}$, $1000\log(n)$, $n\log(n)$, $2n!$, 2^n, 3^n, and $n^2/10000$ in a list so that each function is Big-O of the next function.

Sol:

$1000\log(n)$, $\sqrt{n}$, $n\log(n)$, $\frac{n^2}{10000}$, 2^n, 3^n, $2n!$

2. Give a big-O estimate for the number of times for the Cashier's algorithm for making change for $\boldsymbol{n}$ cents using quarters, dimes, nickels, and pennies.

procedure *change*($c_1, c_2, \cdots, c_r$: values of denominations of coins, $c_1, > c_2 > \cdots > c_r$,
n: a positive integer)
$r :=$ the length of $\{c_i\}$
for $i := 1$ **to** r
  $d_i := 0\{d_i$ counts the coins of denomination c_i used$\}$
  while $(n \geq c_i)$ → 1 comparison
    $d_i := d_i + 1$ → 1 addition
    $n := n - c_i$ → 1 subtration
3 operations in a while loop
return $(d_1, d_2, \cdots, d_r)$ $\{d_i$ counts the coins of denomination c_i used$\}$

Sol: For the big-O estimate, we need to find a worse-case time complexity function f(n) of this algorithm.

Given n cents. We have

# of quarters	# of dime(s)	# of nickel(s)	# of penny
$\left\lfloor \frac{n}{25} \right\rfloor$	$\left\lfloor \frac{24}{10} \right\rfloor = 2$ ← the worse case	$\left\lfloor \frac{9}{5} \right\rfloor = 1$ ← the worse case	$\left\lfloor \frac{4}{1} \right\rfloor = 4$

Since the "while loop" runs once when it gets a coin, then

$$f(n) = 3\left\lfloor \frac{n}{25} \right\rfloor + 3\cdot 2 + 3\cdot 1 + 3\cdot 4 = 3\left\lfloor \frac{n}{25} \right\rfloor + 21 < n \quad \text{when } n > 22$$

$\Rightarrow$ The worse-case time complexity of the cash's algorithm is $O(n)$.