Q 2.1a answer) $t_1(n) \in O(n^2)$ and $t_2(n) \in O(n^3)$

Proof for $t_1(n) \in O(n^2)$:

Def^n : $O(g) = \{f \mid \exists k \in N. f \leq_a k \cdot g\}$

To prove:

$t_1(n) \in O(n^2)$

From definition:

$\exists$ k $\in$ N and $n_0 \in$ N such that $t_1(n) \leq kn^2$ for n > n0.

Taking k = 6,

So, $t_1(n) = 5n^2 + 16 \geq 6n^2$ for $n \in \{0, 1, 2, 3, 4\}$, but $t_1(n) \leq 6n^2$ for $n > 4 = n_0$.

Hence, $t_1(n)$ is in $O(n^2)$.

Proof for $t_1(n) \in O(n^2)$:

Def^n : $O(g) = \{f \mid \exists k \in N. f \leq_a k \cdot g\}$

To prove:

$t_2(n) \in O(n^3)$

From definition:

$\exists$ k $\in$ N and $n_0 \in$ N such that $t_1(n) \leq kn^2$ for $n > n_0$.

Taking k = 7,

So, $t_2(n) = 6n^3+n^2+18 \geq 7n^3$ for $n \in \{0, 1, 2, 3\}$, but $t_2(n) \leq 7n^3$ for $n > 4 = n_0$.

Hence, $t_2(n)$ is in $O(n^3)$.

Q 2.1b answer) As the program runs sequentially the total time is just the addition of the time taken by $t_1(n)$ and t_2, the program belongs to the time complexity for the function $t_1(n) + t_2(n)$.

So, $t_1(n) + t_2(n) \in O(n^3)$

Proof for $t_1(n) + t_2(n) \in O(n^3)$:

Def^n : $O(g) = \{f \mid \exists k \in N.f \leq_a k \cdot g\}$

To prove:

$t_1(n) + t_2(n) \in O(n^3)$

From definition:

$\exists$ k $\in$ N and $n_0 \in$ N such that $t_1(n)+t_2(n) \leq kn^2$ for $n > n_0$.

$t_1(n) + t_2(n) = 6n^3+6n^2+34$

Taking k = 7,

So, $t_1(n) + t_2(n) = 6n^3+6n^2+34 \geq 7n^3$ for $n \in \{0, 1,\ldots 6\}$, but $t_1(n)+t_2(n) \leq 7n^3$ for $n > 6 = n_0$.

Hence, $t_1(n) + t_2(n)$ is in $O(n^3)$.

QN 2.1 c)

Here,

$f_1 \in O(g_1)$ and $f_2 \in O(g_2)$

g_1 and g_2 are some canonical functions.

From defn

$\exists k_1$ s.t $f_1(n) \leq k_1 g_1(n)$

Similarly

$\exists k_2$ s.t $f_2(n) \leq k_2 g_2(n)$

Proof

$f_1(n) + f_2(n) \leq k_1 g_1(n) + k_2 g_2(n)$

$\Rightarrow f_1(n) + f_2(n) \leq \max(k_1, k_2) \{g_1(n) + g_2(n)\}$

$\left[\because \max(k_1, k_2)\{g_1(n) + g_2(n)\} \geq k_1 g_1(n) + k_2 g_2(n)\right.$

$\Rightarrow f_1(n) + f_2(n) \leq \max(k_1, k_2)\ 2 \times \max(g_1(n), g_2(n))$

$\left[\because 2 \times \max(g_1(n), g_2(n)) \geq \cancel{\max}\{g_1(n) + g_2(n)\}\right]$

$$\Rightarrow \underbrace{f_1(n) + f_2(n)}_{f'} \leq \underbrace{2\max(k_1, k_2)}_{k'} \underbrace{\max(g_1(n), g_2(n))}_{g'}$$

Hence,

$f' \in O(g')$

$(f_1 + f_2) \in O(\max\{g_1, g_2\})$

proved

Problem 2.2) Soln

Proof by Induction on n

Basic step: $n = 1$ $(\because n \geq 1)$

$$1^2 = \sum_{K=1}^{n} (2K-1)^2 = (2 \times 1 - 1)^2 = 1$$

For $n = 2$

$$1^2 + 3^2 = \sum_{K=1}^{n} (2K-1)^2 = (2 \times 1 - 1)^2 + (2 \times 2 - 1)^2 = 10$$

Inductive step: Assume that $n = m$

Inductive Hypothesis:

$$1^2 + 3^2 + \ldots + (2m-1)^2 = \sum_{K=1}^{M} (2K-1)^2 = \frac{2m(2m-1)(2m+1)}{6}$$

$\forall m \geq 1$

To prove:

$$1^2 + 3^2 + 5^2 + \ldots + (2m-1)^2 + (2(m+1)-1)^2 = \sum_{K=1}^{m+1} (2K-1)^2$$

$$= \frac{2(m+1)(2(m+1)-1)(2(m+1)+1)}{6}$$

$$= \frac{2(m+1)(2m+1)(2m+3)}{6}$$

$$= \frac{(m+1)(2m+1)(2m+3)}{3}$$

Taking L.H.S :

$$1^2+3^2+5^2+\ldots+(2m-1)^2+(2(m+1)-1)^2$$

$$= \frac{2m(2m-1)(2m+1)}{6}+(2(m+1)-1)^2$$

[$\because$ from our induction hypothesis]

$$= \frac{2m(4m^2-1)}{6}+\frac{4(m+1)^2-4(m+1)+1}{1}$$

$$= \frac{8m^3-2m+24(m^2+2m+1)-24m-24+6}{6}$$

$$= \frac{8m^3+24m^2+22m+6}{6}$$

$$= \frac{4m^3+12m^2+11m+3}{3}$$

$$= \frac{4m^3+4m^2+8m^2+8m+3m+3}{3}$$

$$= \frac{(m+1)4m^2+(m+1)8m+(m+1)3}{3}$$

$$= \frac{(m+1)(4m^2+8m+3)}{3}$$

$$= \frac{(m+1)(4m^2+6m+2m+3)}{3}$$

$$= \frac{(m+1)(2m+1)(2m+3)}{3} = R.H.S$$

Taking M.H.S

$$\sum_{k=1}^{m+1} (2k-1)^2$$

$$= \sum_{k=1}^{m} (2k-1)^2 + (2(m+1)-1)^2$$

$$= 1^2 + 3^2 + 5^2 + \dots + (2m-1)^2 + (2(m+1)-1)^2$$

[$\because$ from our induction hypothesis $\sum_{k=1}^{m} (2k-1)^2 = 1^2 + 3^2 \dots + (2m-1)^2$]

$= L.H.S$

$= R.H.S$

Hence,

We can conclude inductively that

$$1^2 + 3^2 + \dots + (2m-1)^2 = \sum_{k=1}^{m} (2k-1)^2 = \frac{2m(2m-1)(2m+1)}{6}$$

Problem 2.3 solution

(a)

```
factors :: Integral a => a -> [a]

factors n = [x | x <- [1..n], (n mod x) == 0]
```

(b)

```
pythagoras :: (Num a, Eq a, Enum a) => a -> [(a,a,a)]

pythagoras x = [(a,b,c) | a <- [1..x], b <- [1..a], c <- [1..x], (a)^2 + (b)^2 == (c)^2]
```