ICS 2020 Problem Sheet #2

Problem 2.1:

a)

The Big O notation (Landau notation) definition :

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

for $t_1(n)=5n^2 + 16$

$5n^2 + 16 \leq 5n^2 + 16$

=> $5n^2 + 16 \leq 5n^2 + 16n^2$ for $n_0 = 0$ and $n > no$

=> $5n^2 + 16 \leq 21n^2$ // $21 \neq 0$ // $n_0 = 0$ and $n > no$

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

for $t_2(n) = 6n^3 + n^2 + 18$

$6n^3 + n^2 + 18 \leq 6n^3 + n^2 + 18$

=> $6n^3 + n^2 + 18 \leq 6n^3 + n^3 + 18n^3$ for $n_0 = 0$ and $n > no$

=> $6n^3 + n^2 + 18 \leq 25n^3$ // $25 \neq 0$ // $n_0 = 0$ and $n > no$

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

b)

Logically, the entire program belongs to The big O set $O(n^3)$:

$O(1) \subset O(\log2 (n)) \subset O(n) \subset O(n \log2 (n)) \subset O(n^2) \subset O(n^k) \subset O(l^n)$

with $k>2$ and $i>1$

we conclude that:

$O(n^2) \subset O(n^k)$ => $O(n^2) \subset O(n^3)$

c)

We have that $f1 \in O(g1)$ and $f2 \in O(g2)$, then :

$\exists k_1 \in N.\ f_1 \leq_a k_1 \cdot g_1$ and $\exists k_2 \in N.\ f_2 \leq_a k_2 \cdot g_2$

=> $f_1 + f_2 \leq k_1 \cdot g_1 + k_2 \cdot g_2$

=> $f_1 + f_2 \leq k_1 \cdot \max\{g1, g2\} + k_2 \cdot \max\{g1, g2\}$

=> $f_1 + f_2 \leq (k_1 + k_2) \cdot \max\{g1, g2\}$ $(k_1 + k_2) \in N$

then $(f1 + f2) \in O(\max\{g1, g2\})$

Problem 2.2:

$$\text{let } \sum_{k=1}^{n} (2k - 1)^2 = Xn$$

In order to prove that $Xn = n(2n - 1)(2n + 1)/3$

We must show that the case is right for x_1 (x_1 here is our first element here) :

$x_1 = (2*1-1)^2 = 1$ and for $n = 1$: $n (2n - 1) (2n + 1) /3 = 1 * (2-1) * (2+1) /3 = 1$

Now we suppose that: $X_n = n(2n - 1)(2n + 1)/3$

and prove that $X_{n+1} = (n+1)(2n + 1)(2n + 3)/3$

$X_{n+1} = (n+1)(2n + 1)(2n + 3)/3$

(Here we got the element n+1 out of X_{n+1})

$\leftrightarrow X_n + (2(n+1)-1)^2 = (n+1)(2n + 1)(2n + 3)/3$

(here we replaced X_n by $n(2n - 1)(2n + 1)/3$

$\leftrightarrow n(2n - 1)(2n + 1)/3 + (2(n+1)-1)^2 = (n+1)(2n + 1)(2n + 3)/3$

$\leftrightarrow n(2n - 1)(2n + 1)/3 + (2n+1)^2 = (n+1)(2n + 1)(2n + 3)/3$

$\leftrightarrow (2n + 1)(n(2n - 1)/3 + (2n+1)) = (n+1)(2n + 1)(2n + 3)/3$ since $n \neq -1/2$

$\leftrightarrow n(2n - 1)/3 + 2n+1 = (n+1)(2n + 3)/3$

$\leftrightarrow n(2n - 1) + 6n+3 = (n+1)(2n + 3)$

$\leftrightarrow 2n^2 - n + 6n+3 = 2n^2+3n+2n+3$

$\leftrightarrow 2n^2 + 5n+3 = 2n^2+5n+3$ which is true

We can now conclude that $X_n = n(2n - 1)(2n + 1)/3$

<u>Problem 2.3:</u>

a)

a list comprehension that include all factors of 210 would be :

[x | x <- [1..210], 210 `mod` x == 0]

The program would then print :

[1,2,3,5,6,7,10,14,15,21,30,35,42,70,105,210]

b)

that list would be :

[(x,y,z) | x <- [1..100], y <- [1..100], z <- [1..100], x*x + y*y == z*z, if (x,y,z)==(y,x,z) then (x,y,z) 'rem' (y,x,z)]

When I tried it the program didn’t return anything so here is a list that would do the job without removing the duplicate elements :

[(x,y,z) | x <- [1..100], y <- [1..100], z <- [1..100], x*x + y*y == z*z]