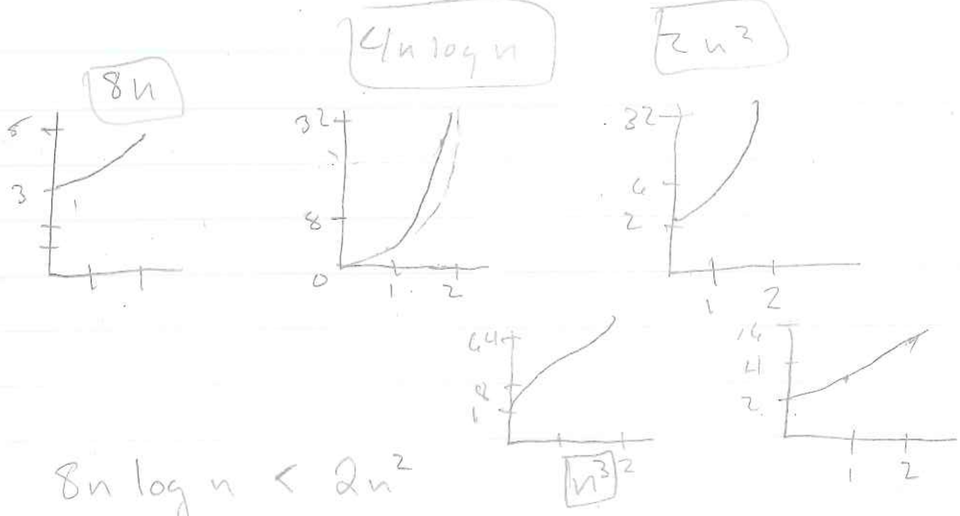
3.1)



3.2) $8n \log n < 2n^2$

$8 \log n < 2n$

$4 \log n < n$

$n = 16$

3.8) 2^{10}, $3n + 100 \log n$, $4n$, $n \log n$, $4n \log n + 2n$

$n^2 + 10n$, n^3, $2^{\log n}$, 2^n

3.10) $d(n) \leq c_1 \cdot f(n)$ $n \geq n_1$ $c_3 = c_1 c_2$ $d(n)e(n) \leq c_3 f(n) g(n)$

$e(n) \leq c_2 \cdot g(n)$ $n \geq n_2$ $n_3 = n_1 n_2$

3.14) Show than $O(\max\{f(n), g(n)\}) = O(f(n) + g(n))$

• Since we are adding. The higest term of the given functions will always be its big O this is because the lower term does not affect the big O

3.17) as $(n+1)^5$ gets higher the +1 becomes less and less important there for n^5 appraches $(n+1)^5$ as n gets biggr

3.18) as n gets larger the +1 matters less and less. this means 2^n approaches 2^{n+1} so $O(2^{n+1}) = O(2^n)$

3.20.) $n^2 \geq n \log n$

$n \geq \log n$

While $n \geq 2$ $n^2 \geq n \log n$

So $n^2 \in \Omega(n \log n)$

3.27.) the Big-O is $O(n^3)$

3.35.) lets say n numbers are stored inside a list

We can take the 3 lists and make 1 big sorted list and check if the number is repeated 3 times if so. they are not disjoint

3.49.) $F(n) \geq c(\frac{3}{2})^n$

Assume $F(k) \geq c(\frac{3}{2})^k$ for $k \geq n_0$

Need to show $F(k+1) \geq (\frac{3}{2})^{k+1}$

$F(k+1) = F(k) + F(k-1)$

$\geq c(\frac{3}{2})^k + c(\frac{3}{2})^{k-1}$

$= c(\frac{3}{2}+1)(\frac{3}{2})^{k-1}$

$\frac{3}{2} + 1 = \frac{5}{2}$

$c(\frac{5}{2})(\frac{3}{2})^{k-1}) > (c(\frac{3}{2})^2 (\frac{3}{2})^{k-1}$

therefor $f(n) \in \Omega(c(\frac{3}{2})^n)$

4.3R) power(2,18) = power(2,9) · power(2·9)
power(2,9) = power(2,4) · power(2,4) · 2
power(2,4) = power(2,2) · power(2,2)
power(2,2) = power(2,1) · power(2,1)
power(2,1) = power(2,0) · power(2,0) · 2
power(2,0) = 1

power(2,1) = 1 · 1 · 2 = 2 = 2
power(2,2) = 2 · 2 = 4
power(2,4) = 4 · 4 = 16
power(2,9) = 16 · 16 · 2 = 512
power(2,18) = 512 · 512 = $\boxed{262,144}$

4.5) s=empty U = {a, b, c, d}
PS(3, (), {A, b, c, D})

pS(2, (A), {B, C, D}) pS(2, (B), {A, C, D}) pS(2, (C), {A, B, D}) pS(2, (D), {A, B})
(1, (AB), {C, D}) (1, (A, C), {BD}) (1, (AD), {B, C}) = abCD, ACBD, ADBC
BACD, BCAD, BDAC = (1, (BA), {C, D}) (1, (BC), {A, D}) (1, (BD), {AC})
CABD, CBAD, CDAB = (1, (CA), {BD}) (1, (CB), {AD}) (1, (CD) {AB})

4.7) the function will take 2 aruments:
an accumulator and string the accumulator
will be for the exponenet of 10^{acc}
the recusive call will move down
len(arg) - 1 and the acc moves up by 1
all whie multiplying the $10^{acc} f(len)$ + recussive call

4.11) Compare the first element in the rest of
the list, if it contains it, return falals, else
recur with a new first elem and a new rest of
the list