1. (a) 
$$6n^2 - 4(n + 2 + 60(n^2))$$

$$6n^2 - 4(n + 2 + 6n^2 + 2n^2)$$

$$= 8n^2$$

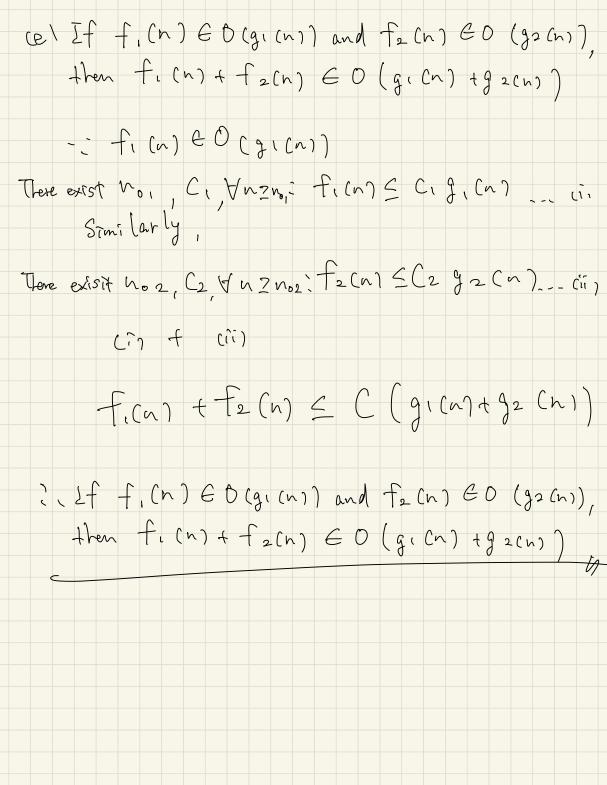
$$= 8n^2$$

$$\frac{1}{2} + \frac{1}{2} +$$

(c) \( \a 7 \righta 7 \righta \) (\log 2 n) \( \in \nagger \) \( \log 2 n \right) \) log2h = logah loga2 e logan = loga2 g (n) . There exist C = loga 2 and no = 1 such that Vn zno, azi: logzn & clogan i. Va71:0(log2n60(logan)

(d) Va 71: a ant ( & O (a an ) If  $\forall a > 1$ ; A = 0 (A = 0)

then there exist C and D = 0 such that  $\forall k > 1, k > k > 0$   $\Delta^{an+1} \leq \Delta^{an}$  Cileft hand side: take loga, then anti right hand side:
take loga, than (i) contradict (IV) i. Va 712 a anti & O (a a )



(b) for i in the range [1, n] {

(c) x = pow(2, n); i = 1;while i <= x { for j in range [1, i] { <some-constant number of atomic/elementary operations> i = i \* 2 Assume that pow(2, n) (i.e., 2<sup>n</sup>) is computed magically in constant time. The inside loop takes O(n) because it runs [1,i] The ouside while loop takes O(logn) because the upper bound, oc is constant but i Increases exponentially 0 (n (8 g n)

Extra credit function myf(integer a, integer n) { integer a1; while n >= 0 { if n==0 then return 1; -> O(() a1 = myf(a, n/2); \_\_\_\_\_ -9 O (logn) if n is even then { return a1 \* a1; -9 O (1) else { return a \* a1 \* a1; -7 0 (1) n = n/2;
} Assume that n/2 = 0, when n < 2. X call the function itself recursively with decreasing n by halt o ((og n) The while loop takes O(log h) because everytime it gets into the loop it decrease h by half while the bound stays the same i Inside of 1900 is O((ogh) and the (oop is O(logh)  $O(((\circ g(N))^{2})$