1. let $f(n) = n^3 - 2n^2 + n$ and $g(n) - 2$ show wether $f(n) = \Omega(g(n))$ is true (or) false justify your answer.

A.) * express the growth rates:

$f(n) = n^3 - 2n^2 + n$

$g(n) = -n^2.$

* dominat terms:

for large n, the dominat terms in $f(n)$ is n^3 because it grows faster than the terms.

* the dominate term $g(n)$ is $-n^2$.

compare $f(n)$ and $g(n)$:

* As n grows large, n^3 will dominate $-2n^2 + n$

* thus $f(n) \approx n^3$ to large n.

* we need to show that $f(n) > c \cdot (-n^2)$ for some $c > 0$ and sufficiently large n.

choose a suitable constant c:

* let's choose $c = 1$. then the inequality becomes:

$n^3 - 2n^2 + n \geq -n^2$

simplify the inequality:

* for large n, n^3 will dominate $-n^2 + n$.

* Hence, $n^3 - n^2 + n$ is positively for sufficiently large n.

$f(n) = \Omega\, g(n)$

$f(n) = n^3 - 2n^2 + n$ is indeeded $\Omega\, g(n) = -n^2$.

2. Determine wether $h(n) = n\log n + n$ is in $\Theta(n\log n)$ prove a rigrous proof for your conclusion.

express the functions:

$h(n) = n \log n + n$

$g(n) = n \log n$.

Dominant terms:

* for large n, $n \log n$ dominates n
* Hence, $h(n) \approx n \log n$ for large n.

form the inequalities:

* we need to show their exist positive constants c_1, c_2 and n_0 such that for all $n \geq n_0$: $c_1 \cdot n \log n \leq n \log n + n \leq c_2 \cdot n \log n$.

prove the lower bound:

* $n \log n$ is always less than or equal to $n \log n + n$, because n is positive for $n > 0$.
* therefore, $c_1 = 1$ works and we have

$$n \log n \leq n \log n + n.$$

prove the upper bound:

* To show $n \log n + n \leq c_2 \cdot n \log n$ for large n, we write the inequality as: $n \log n + n \leq c_2 \cdot n \log n$.
* Divide both sides by n:

$$\log n + 1 \leq c_2 \log n.$$

* we can choose $c_2 = 2$. then for large n:

$$\log n + 1 \leq 2 \log n.$$

combine the bounds:

we have show that

$$n \log n \leq n \log n + n \leq 2 n \log n.$$

this holds for all $n \geq n_0$.

$\therefore h(n) = n \log n + n = \Theta(n \log n)$.

3. solve the following recurrence relation and find the order of growth for solutions.

$$T(n) = 4T(n/2) + n^2, \quad T(1) = 1.$$

master's theorem:

* $f(n) = O(n^c)$ where $c < \log_b a$ then $T(n) = \Theta(n^{\log_b a})$

* $f(n) = \Theta(n^{\log_b a})$, then $T(n) = \Theta(n^{\log_b a} \log n)$.

* $f(n) = \Omega(n^c)$ where $c > \log_b a$, and if $a f(\frac{n}{b}) \le K f(n)$ for some $K<1$ and sufficiently large n, then $T(n) = \Theta f(n)$.

Apply:

* calculate $\log_b a$:

$$\log_b a = \log_2 4 = 2.$$

* compare $f(n)$ with $n^{\log_b a}$.

$$f(n) = n^2$$

$$n^{\log_b a} = n^2.$$

since $f(n) = n^2$ and $n^2 = n^{\log_b a}$, we are in case 2 of the master theorem, which states:

if $f(n) = \Theta(n^{\log_b a})$, then $T(n) = \Theta(n^{\log_b a} \log n)$.

$T(n) = 4T(n/2) + n^2$ is $T(n) = \Theta(n^2 \log n)$.

4. Given an array of [4, -2, 5, 3, 10, -5, 2, 8, -3, 6, 7] integers. find the maximum and minimum product that can be obtained by multiplying two integers from the array.

A.) sort the array:

* sorting help us easily find the largest and smallest elements.

* sorted array

[-9, -8, -~~6~~, -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]

maximum product:

* two largest positive numbers in the sorted array 10 and 11

$10 \times 11 = 110.$

* the two most negative numbers are -9 and -8

$(-9) \times (-8) = 72.$

* so, the maximum product is 110.

minimum product:

* the two smallest numbers are -9 and -8.

$(-9) \times (-8) = 72$.

* the most negative product can also be obtained by multiplying the smallest negative number -9 and the largest number 11.

$(-9) \times 11 = -99$.

* so, the minimum product is -99.

conclusion:

maximum product: 110

minimum product: -99.

5. Demonstrates binary search method to search Key = 23, from the array arr[] = {2, 5, 8, 12, 16, 23, 38, 56, 72, 9}

Intital state:

low = 0

high = 9.

* first iteration:

* calculate mid = $\left[\frac{0+9}{2}\right] = 4$.

* compare arr[mid] with key:

arr[4] = 16

since 16 < 23, set low = 5.

calculate mid = $\frac{5+9}{2} = 7$.

* compare arr[mid] with key.

arr[7] = 56.

since 56 > 23, set high = 6.

third iteration:

mid = $\frac{5+6}{2} = 5$.

* compare arr[mid] with key:

arr[5] = 23.

since 23 == 23.

* key element is found at index 5.