

H

a-) $f(n) = n^2 + 7n$ $g(n) = n^2 + 7$

• Before we can comment on the solution, we need to take limit to infinity.

$$\lim_{n \rightarrow \infty} \left(\frac{f(n)}{g(n)} \right) = \lim_{n \rightarrow \infty} \left(\frac{n^2 + 7n}{n^2 + 7} \right) \xrightarrow{\text{L'Hospital}} \lim_{n \rightarrow \infty} \left(\frac{2n + 7}{2n} \right) = \lim_{n \rightarrow \infty} \left(\frac{2}{2} \right) = \lim_{n \rightarrow \infty} \left(\frac{1}{1} \right) = 1$$

• The final solution approaches to "1".

• So the answer is $f(n) = O(g(n))$

b-) $f(n) = 12n + \log_2(n^2)$ $g(n) = n^2 + 6n$

$$\lim_{n \rightarrow \infty} \left(\frac{12n + \log_2(n^2)}{n^2 + 6n} \right) = \lim_{n \rightarrow \infty} \left(\frac{\frac{12}{n} + \frac{\log_2(n^2)}{n^2}}{1 + \frac{6}{n}} \right) = \lim_{n \rightarrow \infty} \left(\frac{\frac{12}{n} + \frac{\log_2(n^2)}{n^2}}{1 + \frac{6}{n}} \right) = 0$$

• Answer is $f(n) = O(g(n))$

c-) $f(n) = n \log_2(3n)$ $g(n) = n + \log_2(8n^3)$

$$\lim_{n \rightarrow \infty} \left(\frac{n \log_2(3n)}{n + \log_2(8n^3)} \right) \xrightarrow{\text{dividing}} \lim_{n \rightarrow \infty} \left(\frac{\log_2(3n)}{1 + \frac{\log_2(8n^3)}{n}} \right) = \lim_{n \rightarrow \infty} \left(\frac{\log_2(3n)}{1 + \frac{3}{n}} \right) = \infty$$

$$\lim_{n \rightarrow \infty} \frac{\log_2(8n^3)}{n} = \lim_{n \rightarrow \infty} \frac{24n^2}{8n^3 \ln 2} = \lim_{n \rightarrow \infty} \frac{3}{n \ln 2} = 0$$

• Solution is $f(n) = \Omega(g(n))$

d-) $f(n) = n^n + 5n$ $g(n) = 3 \cdot 2^n$

$$\lim_{n \rightarrow \infty} \left(\frac{n^n + 5n}{3 \cdot 2^n} \right) = \frac{1}{3} \left(\lim_{n \rightarrow \infty} \left(\frac{n^n}{2^n} \right) + \lim_{n \rightarrow \infty} \left(\frac{5n}{2^n} \right) \right) = \infty$$

• Solution approaches infinity so the answer is $f(n) = \Omega(g(n))$

e-) $f(n) = \sqrt[3]{2n}$ and $g(n) = \sqrt{3n}$

$$\lim_{n \rightarrow \infty} \left(\frac{\sqrt[3]{2n}}{\sqrt{3n}} \right) = \frac{1}{\sqrt{6}} \lim_{n \rightarrow \infty} \left(\frac{1}{n^{1/6}} \right) = 0$$

• The solution approaches zero so the answer is $\Rightarrow f(n) = O(g(n))$

2) a) worst-case time complexity of the method is $O(n)$.
• Because of code block iterations repeat it itself as long as input size.

b) Method B calls method A. Method A worst-case time comp. is $O(n)$
also method B calls method A functions n times, so $O(n) * O(n) = O(n^2)$

c) It is impossible to calculate, because of loop never ends.

d) worst-case is all input is lower than k so the time comp is $O(n)$

3) There is no difference between two method in terms of time complexities.
• But withLoop function is more efficient because it has a simpler implementation and does the same task in fewer lines of code.
• Also another advantage is when we change the input size, we have to change to the code of withLoop function but in withLoop function can handle to what extent.

4) No, there is no solution, to find exact number in a constant time.
We have to search all array to it includes our target number or not.
and it takes $O(n)$ time.

pseudo-code:

```
function find-target(array, target):  
    int idx := 0  
    while idx < array.length do  
        if array[idx] == target:  
            return true  
        idx++  
    return false
```


5)

• My solution approach:

- 1) Find min1 and max1 values in array A. $O(n)$
- 2) Find max2 and min2 values in array B. $O(n)$
- 3) Compare $\text{min1} \times \text{max2}$ and $\text{max1} \times \text{min2}$. $O(1)$
- 4) return the smallest.

pseudo code:

function find-min-product(A, B, n, m)

 int idx := 0

 int min1 := infinity

 int max1 := -infinity

 int min2 := min

 int max2 := max

 while idx < n do

 if A[idx] > max1:

 max1 = A[idx]

 if A[idx] < min1:

 min1 = A[idx]

 idx++

 idx := 0

 while idx < m do

 if B[idx] > max2:

 max2 = B[idx]

 if B[idx] < min2:

 min2 = B[idx]

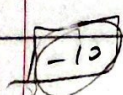
 idx++

 if $\text{min2} \times \text{max1} < \text{min1} \times \text{max2}$ do

 return $\text{min2} \times \text{max1}$

 return $\text{max2} \times \text{min1}$

Time Complexity = $O(n) + O(n) = 2 \times (O(n)) = O(n)$



mb