Lab Exercises 4

Example. Take the following list of functions and arrange them in ascending order of growth rate. That is, if function $g(n)$ immediately follows function $f(n)$ in your list, then it should be the case that $f(n)$ is $O(g(n))$.

1. $f_1(n) = 10^n$
2. $f_2(n) = n^{1/3}$
3. $f_3(n) = n^n$
4. $f_4(n) = \log_2 n$
5. $f_5(n) = 2^{\sqrt{\log_2 n}}$

Theorem 1: For every $b > 1$ and every $x > 0$, we have $\log_b n = O(n^x)$

Proof: Recall that $\log_b n = \frac{\ln n}{\ln b}$, where ln is the natural logarithm. Thus, we have:

$$\log_b n = \frac{\ln n}{\ln b}$$

We know that for any $x > 0$, the function n^x grows faster than $\ln n$ as n becomes large. Specifically, the limit:

$$\lim_{n\to\infty} \frac{\ln n}{n^x} = 0$$

This means that for sufficiently large n, $\ln n$ is dominated by n^x.

Since $\lim_{n\to\infty} \frac{\ln n}{n^x} = 0$, there exists a constant $C' > 0$ and an n_0 such that for all $n \geq n_0$:

$$\frac{\ln n}{n^x} \leq C'$$

Multiplying both sides by n^x and then by $\frac{1}{\ln b}$ (which is positive since $b > 1$), we obtain:

$$\log_b n = \frac{\ln n}{\ln b} \leq \frac{C' \cdot n^x}{\ln b}$$

Let $C = \frac{C'}{\ln b}$. Then:

$$\log_b n \leq C \cdot n^x$$

We have shown that there exist constants $C > 0$ and n_0 such that for all $n \geq n_0$, $\log_b n \leq C \cdot n^x$. Therefore, $\log_b n = O(n^x)$.

Theorem 2: For every $r > 1$ and every $d > 0$, we have $n^d = O(r^n)$.

Proof: For $r > 1$ and $d > 0$, the exponential function r^n grows faster than the polynomial function n^d as n becomes large. Specifically, the limit:

$$\lim_{n\to\infty} \frac{n^d}{r^n} = 0$$

This means that for sufficiently large n, n^d is dominated by r^n.

Since $\lim_{n\to\infty} \frac{n^d}{r^n} = 0$, there exists a constant $C > 0$ and an n_0 such that for all $n \geq n_0$:

$$\frac{n^d}{r^n} \leq C$$

Multiplying both sides by r^n, we obtain:

$$n^d \leq C \cdot r^n$$

We have shown that there exist constants $C > 0$ and n_0 such that for all $n \geq n_0$, $n^d \leq C \cdot r^n$. Therefore, $n^d = O(r^n)$.

Solution: We can deal with functions f_1, f_2, and f_4 very easily, since they belong to the basic families of exponentials, polynomials, and logarithms. In particular, by Theorem 1, we have $f_4(n) = O(f_2(n))$; and by Theorem 2, we have $f_2(n) = O(f_1(n))$.

Now, the function f_3 isn't so hard to deal with. It starts out smaller than 10^n, but once $n \geq 10$, then clearly $10^n \leq n^n$. This is exactly what we need for the definition of $O(\cdot)$ notation: for all $n \geq 10$, we have $10^n \leq c \cdot n^n$, where in this case c = 1, and so $10^n = O(n^n)$.

Finally, we come to function f_5, which is admittedly kind of strangelooking. A useful rule of thumb in such situations is to try taking logarithms to see whether this makes things clearer. In this case, $\log_2 f_5(n) = \sqrt{\log_2 n} = (\log_2 n)^{1/2}$. $\log_2 f_4(n) = \log_2(\log_2 n)$, while $\log_2 f_2(n) = \frac{1}{3}\log_2 n$. All of these can be viewed as functions of $\log_2 n$, and so using the notation $z = \log_2 n$, we can write

$$\log_2 f_2(n) = \frac{1}{2}z$$
$$\log_2 f_4(n) = \log_2 z$$
$$\log_2 f_5(n) = z^{1/2}$$

Now it's easier to see what's going on. First, for $z \geq 16$, we have $log_2 z \leq z^{1/2}$. But the condition $z \geq 16$ is the same as $n \geq 2^{16} = 65536$; thus once $n \geq 2^{16}$ we have $\log_2 f_4(n) \leq \log_2 f_5(n)$, and so $f_4(n) \leq f_5(n)$. Thus we can write $f_4(n) = O(f_5(n))$.

Similarly we have $z^{1/2} \leq \frac{1}{3}z$ once $z \geq 9$ (once $n \geq 2^9 = 512$). For n above this bound we have $\log_2 f_5(n) \leq \log_2 f_2(n)$ and hence $f_5(n) \leq f_2(n)$, and so we can write $f_5(n) = O(f_2(n))$. Essentially, we have discovered that $2^{\sqrt{\log_2 n}}$ is a function whose growth rate lies somewhere between that of logarithms and polynomials.

The final list: $f_4 \leq f_5 \leq f_2 \leq f_1 \leq f_3$

Problem 1 (10 points). For each row i in Table 1, determine whether A_i belongs to $O(B_i)$, $\Omega(B_i)$, or $\Theta(B_i)$. Place a checkmark ($\checkmark$) in the appropriate column(s). The first two rows are provided as examples. **No explanation is required**.

Table 1: Big-O, Omega, and Theta Classification

A	B	O	Ω	Θ
$5n$	n	$\checkmark$	$\checkmark$	$\checkmark$
5	n	$\checkmark$		
n^3	$n^2 \log n$			
2^n	3^n			
$\log(n!)$	$n \log n$			
$\sqrt{n}$	$n^{0.5}$			
$(\frac{1}{2})^n$	1			
$n^{1.5}$	$n(\log n)^2$			
$\log_2 n$	$\ln n$			
4^n	2^{2n}			
n	$\log(n^n)$			
$100n + 50$	n			

Problem 2 (10 points). Consider the following list of functions. Arrange them in ascending order of growth rate. If function $g(n)$ immediately follows function $f(n)$ in your list, then it should be the case that $f(n) = O(g(n))$.

1. $f_1(n) = n^{2.5}$
2. $f_2(n) = \sqrt{2n}$
3. $f_3(n) = n + 10$
4. $f_4(n) = 10^n$
5. $f_5(n) = 100^n$
6. $f_6(n) = n^2 \log n$

Requirements:

- Provide your final ordered list
- For each consecutive pair (f_i, f_j), briefly justify why $f_i = O(f_j)$
- Consider asymptotic behavior as $n \to \infty$

Problem 3 (10 points). Given an array of integers nums and an integer target, return indices of the two numbers such that they add up to target. You may assume that each input would have exactly one solution, and you may not use the same element twice. You can return the answer in any order.

Example 1:

Input: nums = [2,7,11,15], target = 9 Output: [0,1] Explanation: Because nums[0] + nums[1] == 9, we return [0, 1] or [1, 0].

Example 2:

Input: nums = [3,2,4], target = 6 Output: [1,2]

Example 3:

Input: nums = [3,3], target = 6 Output: [0,1]

Requirements:

- Provide your Python code.
- Analyze the time complexity (Big-O) of your algorithm.

Problem 4 (10 points). True or False? **No explanation is required**.

1. Is $2^{n+1} = O(2^n)$
2. Is $2^{2n} = O(2^n)$

Problem 5 (10 points). Prove that $n^2 = O(2^n)$