1/1 point

1. Introduction and Learning Outcomes

The goal of this assignment is to practice with big-O notation.

Recall that we write $f(n) = O(g(n))$ to express the fact that $f(n)$ grows no faster than $g(n)$: there exist constants N and $c > 0$ so that for all $n \geq N$, $f(n) \leq c \cdot g(n)$.

Is it true that $\log_2 n = O(n^2)$?

- (selected) Yes

  Correct
  A logarithmic function grows slower than a polynomial function.

- No

1/1 point

2. $n \log_2 n = O(n)$

- Yes
- (selected) No

  Correct
  To compare these two functions, one first cancels n. What is left is $\log_2 n$ versus 1. Clearly, $\log_2 n$ grows faster than 1.

1/1 point

3. $n^2 = O(n^3)$

- (selected) Yes

  Correct
  n^a grows slower than n^b for constants $a < b$.

- No

1/1 point

4. $n = O(\sqrt{n})$

- Yes
- (selected) No

  Correct
  $\sqrt{n} = n^{1/2}$ grows slower than $n = n^1$ as $1/2 < 1$.

1/1 point

5. $5^{\log_2 n} = O(n^2)$

- Yes
- (selected) No

  Correct
  Recall that $a^{\log_b c} = c^{\log_b a}$ so $5^{\log_2 n} = n^{\log_2 5}$. This grows faster than n^2 since $\log_2 5 = 2.321\ldots > 2$.

1/1 point

6. $n^5 = O(2^{3 \log_2 n})$

- Yes
- (selected) No

  Correct
  $2^{3 \log_2 n} = (2^{\log_2 n})^3 = n^3$ and n^3 grows slower than n^5.

1/1 point

7. $2^n = O(2^{n+1})$

- (selected) Yes

  Correct
  $2^{n+1} = 2 \cdot 2^n$, that is, 2^n and 2^{n+1} have the same growth rate and hence $2^n = \Theta(2^{n+1})$.

- No