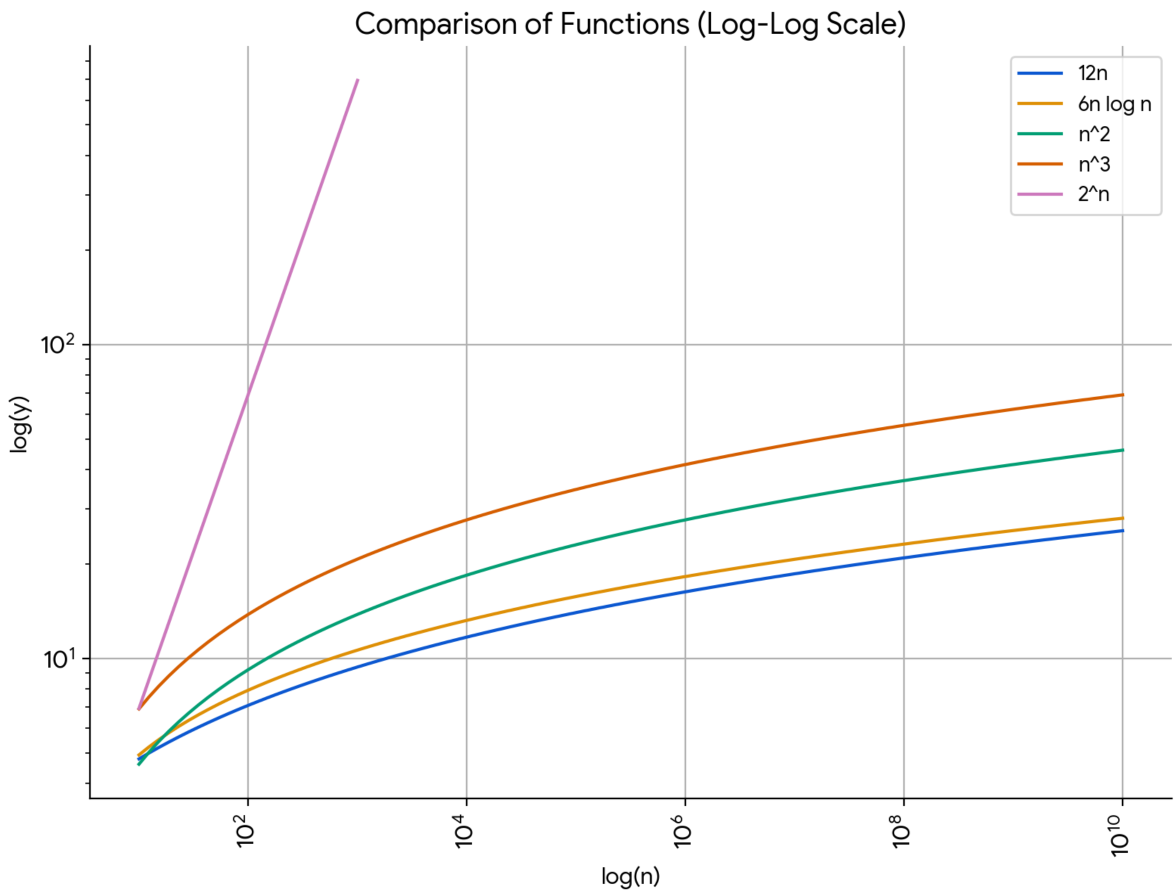
R-1.1



R1.2

To find the value n_0 where Algorithm A becomes more efficient than Algorithm B, we need to solve the inequality:

$10n \log n < n^2$

This inequality is difficult to solve analytically. However, we can use numerical methods or plotting techniques to find an approximate value for n_0.

By plotting the two functions $10n \log n$ and n^2 on a graph, we can visually determine the intersection point, which will give us an approximate value for n_0.

R1.6

To order functions by their big-O notation, we need to compare their growth rates as n approaches infinity. Here's the ordered list:

1. **Constant functions:** 1/n, 2
2. **Logarithmic functions:** log log n, 4 log n
3. **Polynomial functions:** n, n log n, n^2 log n, n^3
4. **Exponential functions:** 2^n, 4^n, 2^n log n

R-1.10 Give a big-O characterization, in terms of n, of the running time of the Loop1 method below:

```
Algorithm Loop1(n)

s = 0

for i = 1 to n do

  s = s + i
```

The loop iterates n times. In each iteration, a constant-time operation is performed. Therefore, the total running time is **O(n)**.

R-1.14 Perform a similar analysis for method Loop5 below:

```
Algorithm Loop5(n)

s = 0

for i = 1 to n do

  for j = 1 to i do

    s = s + i
```

The outer loop iterates n times. For each iteration of the outer loop, the inner loop iterates i times. The total number of iterations is:

1 + 2 + 3 + ... + n = n(n+1)/2

This is approximately **O(n^2)**.

Prove: log b(x^a) = a log b(x)

We can use the properties of logarithms to prove this:

Let $y = \log b(x^a)$. Then, by the definition of logarithms:

$b^y = x^a$

Taking the logarithm of both sides with base b:

$\text{Log } b(b^y) = \log b(x^a)$

Using the property $\log b(b^x) = x$:

$y = a \log b(x)$

Therefore, $\log b(x^a) = a \log b(x)$.