

Exercise walk-through

5.1

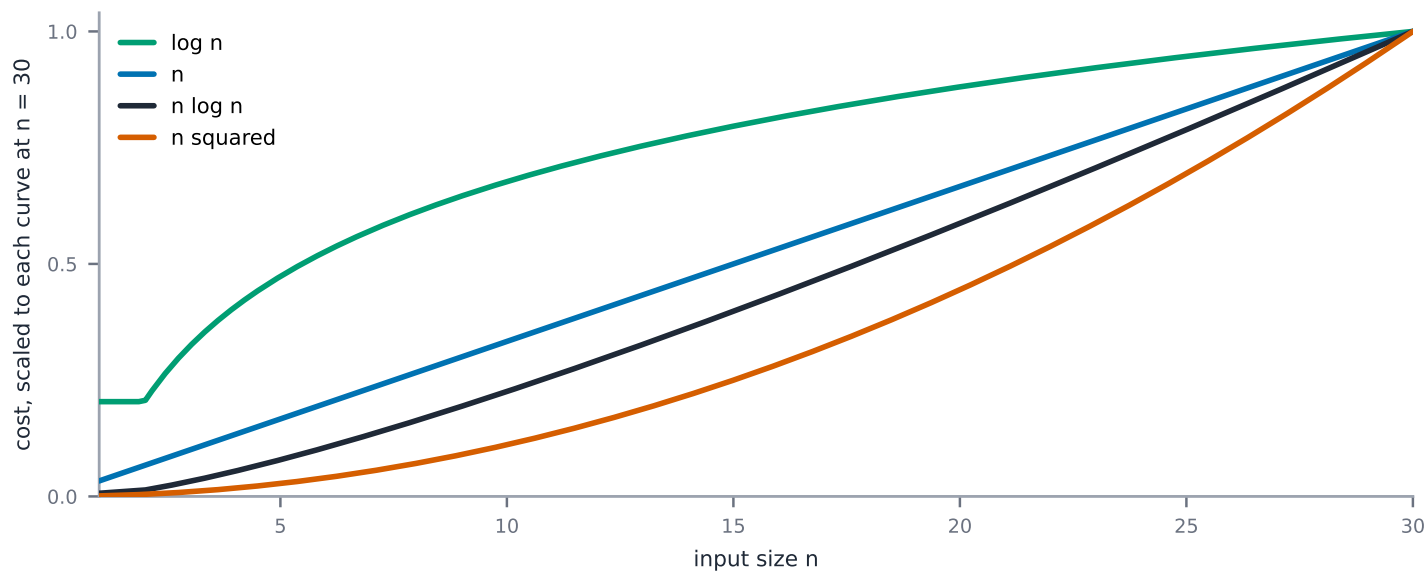
Asymptotic analysis

University Computer Science

Asymptotic analysis: read the exponent off the data

- 1 **Classes** At $n = 1000$: $n \log n$ is about 9 966 and n squared is a million - a factor of 100; log of a million is about 20.
- 2 **Logs** Halving each step touches $\log n$ levels: 16 items resolve in 4 probes because 2 to the 4th is 16.
- 3 **Doubling** Nested loops at $n = 100$ run 10 000 bodies; at $n = 200$ they run 40 000 - the ratio 4 is the exponent.

Complexity curves



Growth of the common classes over one axis: logarithmic stays flat, quadratic bends the page.

Exercises 2.1-2.3

2.1 State the class of $7n \log_2 n + 2n + 40$, naming the rule you applied. [3]

2.2 A halving loop starts at $n = 1024$. State how many iterations to reach 1. [3]

2.3 Give the $\log_2$ values for $n = 10, 100, 1000$, rounded to integers. [3]

Exercises 2.1·2.3

2.1 State the class of $7n \log_2 n + 2n + 40$, naming the rule you applied. [3]

2.1

- $\Theta(n \log n)$ ✓.
 - Dominant term wins; constants and $2n$ dropped ✓.
-

2.2 A halving loop starts at $n = 1024$. State how many iterations to reach 1. [3]

2.2

- 10 iterations: $2^{10} = 1024$ ✓.
-

2.3 Give the $\log_2$ values for $n = 10, 100, 1000$, rounded to integers. [3]

2.3

- 3, 7, 10 ✓.

Exercise 3.1

3.1 Two programs run on the same machine. Program A performs $100n$ operations; program B performs $2n^2$ operations.

State the size n at which B first overtakes A (solve $2n^2 = 100n$). Classify each program, justifying by the dominant term. State which program you run at $n = 10$ and which at $n = 1000$, with the operation counts at both sizes for each program. [8]

Exercise 3.1

3.1 Two programs run on the same machine. Program A performs $100n$ operations; program B performs $2n^2$ operations.

State the size n at which B first overtakes A (solve $2n^2 = 100n$). Classify each program, justifying by the dominant term. State which program you run at $n = 10$ and which at $n = 1000$, with the operation counts at both sizes for each program. [8]

3.1

- $2n^2 = 100n$ gives $n = 50$: B first overtakes A there ✓.
- A is $\Theta(n)$ (linear); B is $\Theta(n^2)$ – the dominant terms decide ✓.
- At $n = 10$: A 1 000, B 200 – run B ✓.
- At $n = 1000$: A 100 000, B 2 000 000 – run A ✓.

Exercise 3.2

3.2 A loop runs `while i > 1: i = i // 2` and inside it a second loop runs n times.

State what each loop contributes and multiply them into one class. Compare its cost at $n = 1024$ with a pure n^2 loop at the same n , computing both values. Explain in one line why halving produces a logarithm. [7]

Exercise 3.2

3.2 A loop runs `while i > 1: i = i // 2` and inside it a second loop runs n times.

State what each loop contributes and multiply them into one class. Compare its cost at $n = 1024$ with a pure n^2 loop at the same n , computing both values. Explain in one line why halving produces a logarithm. [7]

3.2

- The halving loop contributes $\log_2 n$ iterations; the inner loop n each time ✓.
- Product: $\Theta(n \log n)$ ✓.
- At $n = 1024$: $n \log_2 n = 1024 \times 10 = 10\,240$ against $n^2 = 1024^2 = 1\,048\,576$ – about 102 times more ✓.
- Each iteration halves the input, so the count of steps to reach 1 is the number of halvings – the logarithm ✓.