Searching, Sorting and Complexity Analysis - 1 in Python

Algorithms

- Algorithms describe processes that run on real computers with finite resources.
- Processes consume two resources: processing time and space or memory.
- When run with the same problems or data sets, processes that consume less of these two resources are of higher quality than processes that consume more, and so are the corresponding algorithms

Measuring The Run Time of An Algorithm

- Benchmarking of profiling is One way to measure the time cost of an algorithm is to use the computer’s clock to obtain an actual run time.
- The process starts by determining the time for several different data sets of the same size and then calculates the average time.

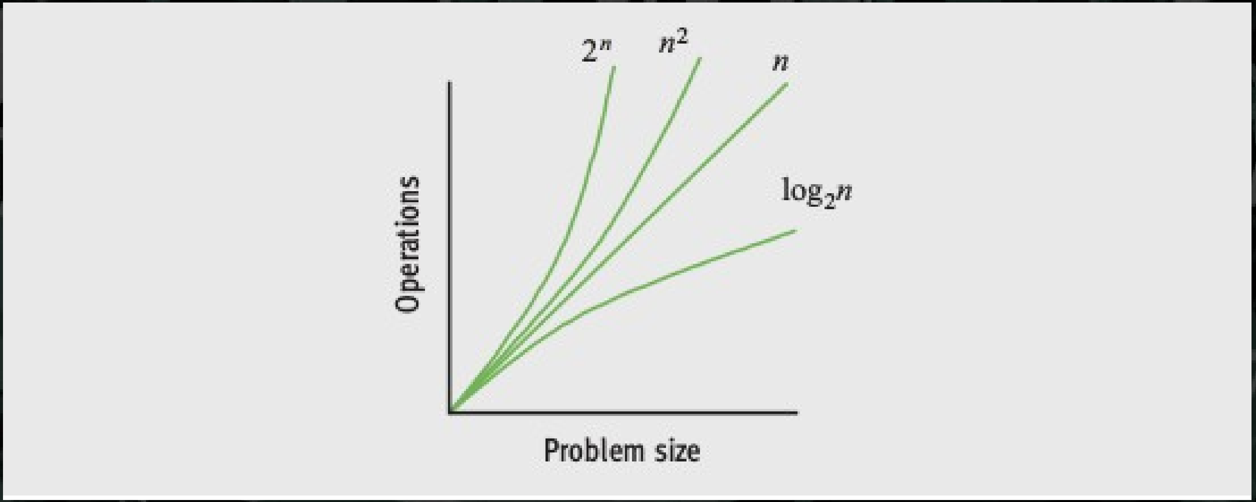
An Example Application for Measuring The Run Time of An Algorithm

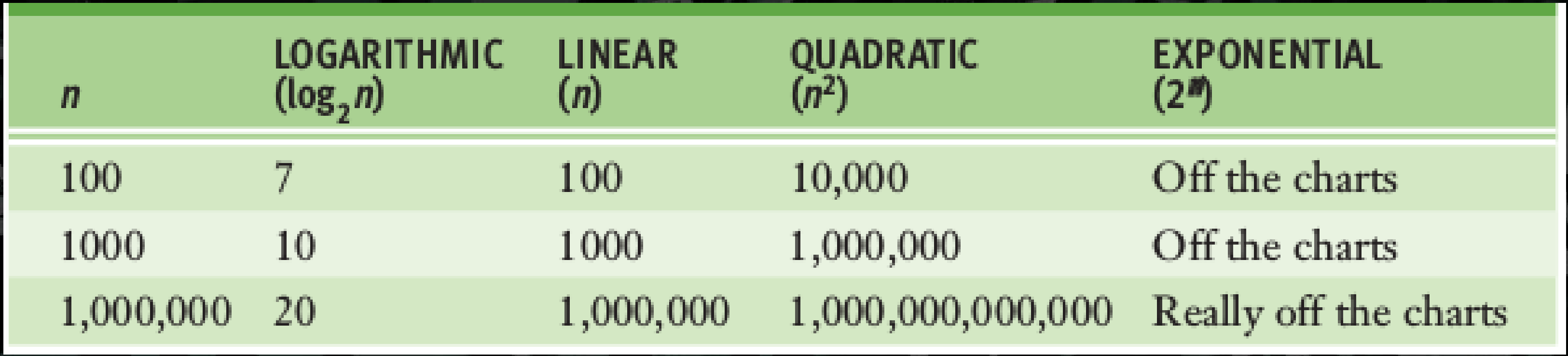
- The program that implements an algorithm that countsfrom 1 to a given number, Thus, the problem size is the number.
- The program start with the number 10,000,000, time the algorithm, and output the running time to the terminal window.
- The program doubles the size of this number and repeat this process until five iterations
- Source code of the program – look at timing1.py

Complexity Analysis

Complexity Analysis is a method of determining the efficiency of algorithms that allows us to rate them independently of platform-dependent timings or impractical instruction counts.

Order of Complexity Analysis





n	LOGARITHMIC ($\log_2 n$)	LINEAR (n)	QUADRATIC (n^2)	EXPONENTIAL (2^n)
100	7	100	10,000	Off the charts
1000	10	1000	1,000,000	Off the charts
1,000,000	20	1,000,000	1,000,000,000,000	Really off the charts

Big O Notation

- Focus on one term as dominant
- For example consider this polynomial :

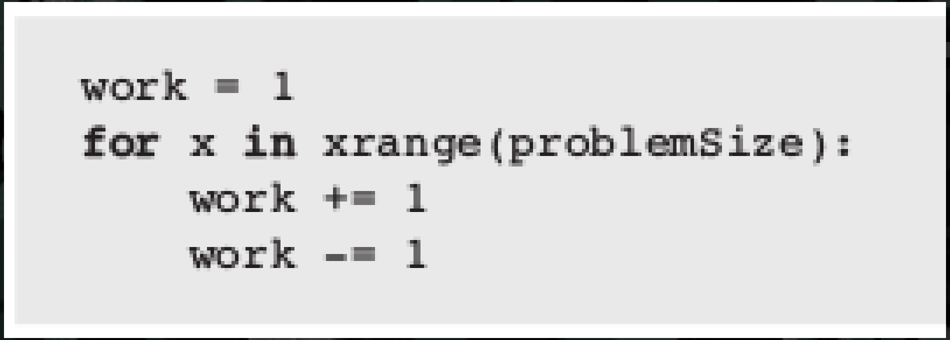
$\frac{1}{2} n^2 - \frac{1}{2} n \rightarrow$ Drop $\frac{1}{2}$ n.

The Role of the Constant of Proportionality

- The constant of proportionality involves the terms and coefficients that are usually ignored during big-O analysis.
- However, when these items are large, they may have an impact on the algorithm, particularly for small and medium-sizeddata sets. For example, no one can ignore the difference between n and n / 2,when n is $1,000,000.

The Role of the Constant of Proportionality

- The amount of abstract work performed by this algorithm is 3n + 1.



```
work = 1
for x in xrange(problemSize):
    work += 1
    work -= 1
```

Reference

Fundamentals of Python from First Program Through Data Structures Chapter 11