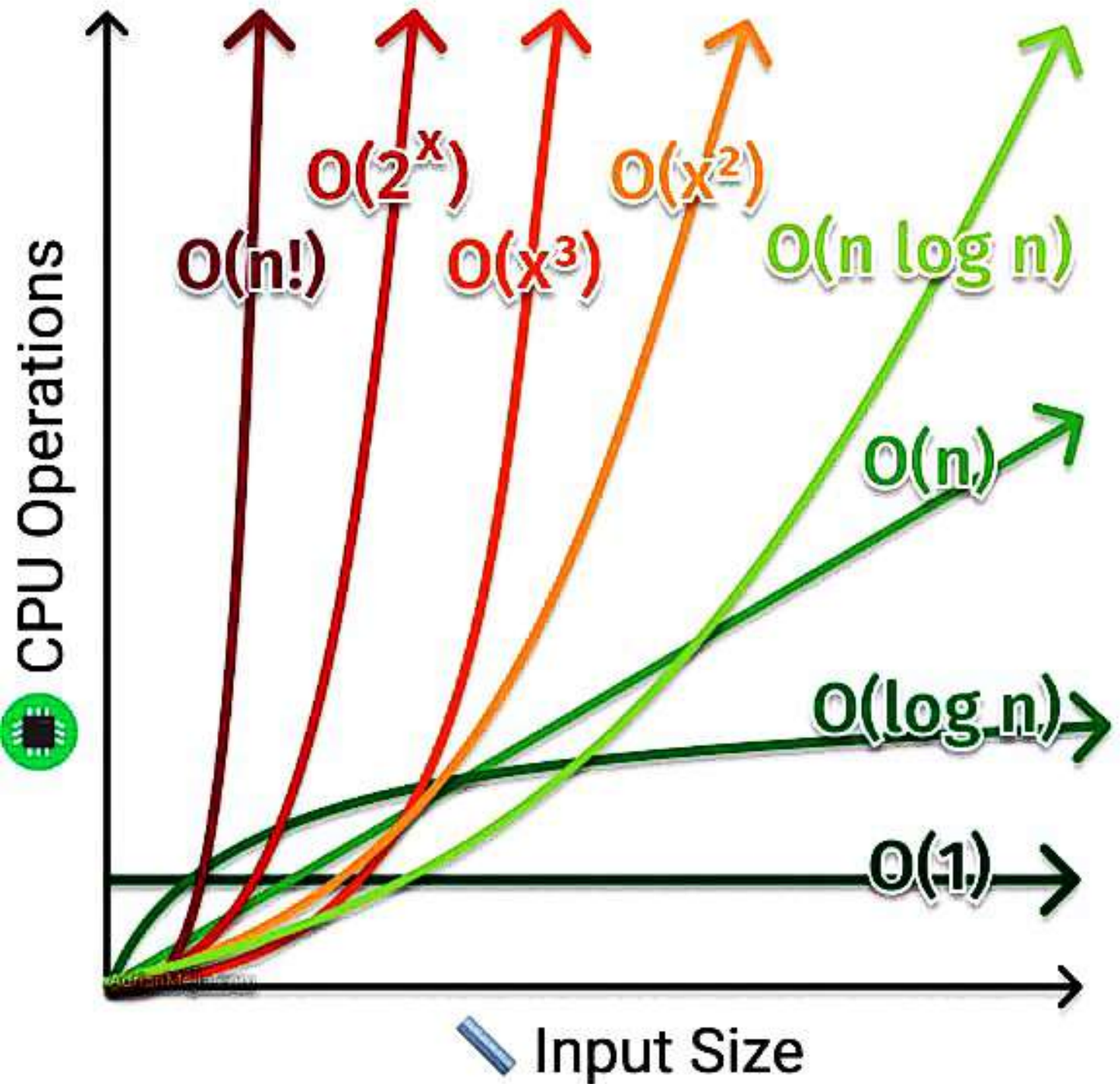| Function Name	Big-O Notation
Constant	O(c)
Linear	O(n)
Quadratic	O(n^2)
Cubic	O(n^3)
Exponential	O(2^n)
Logarithmic	O(log(n))
Log Linear	O(nlog(n))

Algorithm	Best Time Complexity	Average Time Complexity	Worst Time Complexity	Worst Space Complexity
Linear Search	O(1)	O(n)	O(n)	O(1)
Binary Search	O(1)	O(log n)	O(log n)	O(1)
Bubble Sort	O(n)	O(n^2)	O(n^2)	O(1)
Selection Sort	O(n^2)	O(n^2)	O(n^2)	O(1)
Insertion Sort	O(n)	O(n^2)	O(n^2)	O(1)
Merge Sort	O(nlogn)	O(nlogn)	O(nlogn)	O(n)
Quick Sort	O(nlogn)	O(nlogn)	O(n^2)	O(log n)
Heap Sort	O(nlogn)	O(nlogn)	O(nlogn)	O(n)
Bucket Sort	O(n+k)	O(n+k)	O(n^2)	O(n)
Radix Sort	O(nk)	O(nk)	O(nk)	O(n+k)
Tim Sort	O(n)	O(nlogn)	O(nlogn)	O(n)
Shell Sort	O(n)	O((nlog(n))^2)	O((nlog(n))^2)	O(1)

⏱ Time Complexity

CPU Operations

O(n!)　O(2^x)　O(x^3)　O(x^2)　O(n log n)

O(n)

O(log n)

O(1)

Input Size

Python complexities

Lists [a, b, c, ...]	
`mylist.append(val)`	O(1)
`mylist[i]`	O(1)
`val in mylist`	O(N)
`for val in mylist:`	O(N)
`mylist.sort()`	O(N log N)

Dicts {k:v, ...}	
`mydict[key] = val`	O(1)
`mydict[key]`	O(1)
`key in mydict`	O(1)
`for key in mydict:`	O(N)

Sets {a, b, c, ...}	
`myset.add(val)`	O(1)
`val in myset`	O(1)
`for val in myset:`	O(N)

☝ Pro-tip: replace lists with sets

@ nedbat