"Programmers waste enormous amounts of time thinking about, or worrying about, the speed of noncritical parts of their programs, and these attempts at efficiency actually have a strong negative impact when debugging and maintenance are considered. We should forget about small efficiencies, say about 97% of the time: **premature optimization is the root of all evil**. Yet we should not pass up our opportunities in that critical 3%."

—Don Knuth

Write the tabulation template for fib



make-change, infinite coins

Given a value and a set of coins, what is the minimum number of coins required to sum to the value, assuming we have an infinite number of each coin?

longest-common substring (LCS)

How similar are these strings?

The longest-common substring of **s1** and **s2** is the longest string that is a *non-consecutive* substring of both **s1** and **s2**.

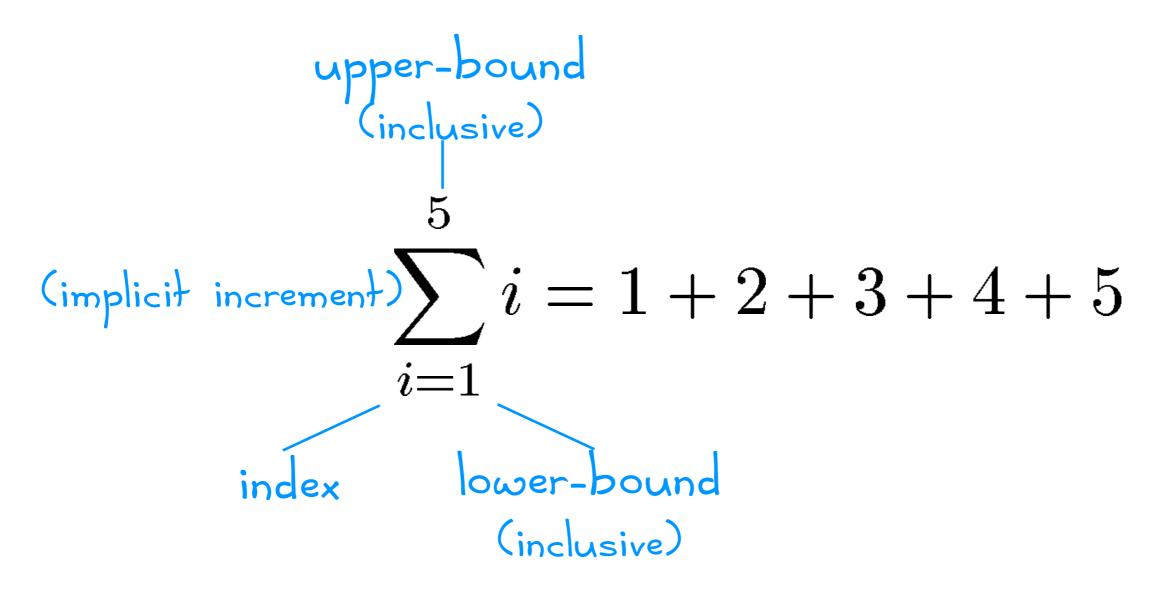
lcs('x', 'y') == 0 lcs('car', 'cat') == 2
lcs('x', '') == 0 lcs('human', 'chimpanzee') == 4
lcs('', 'x') == 0

How many times does the platypus quack?

platypus.quack()

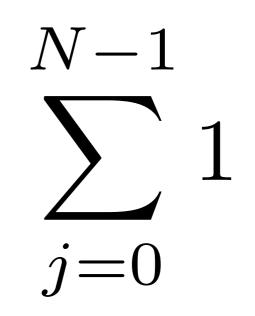
How many times does the platypus quack?

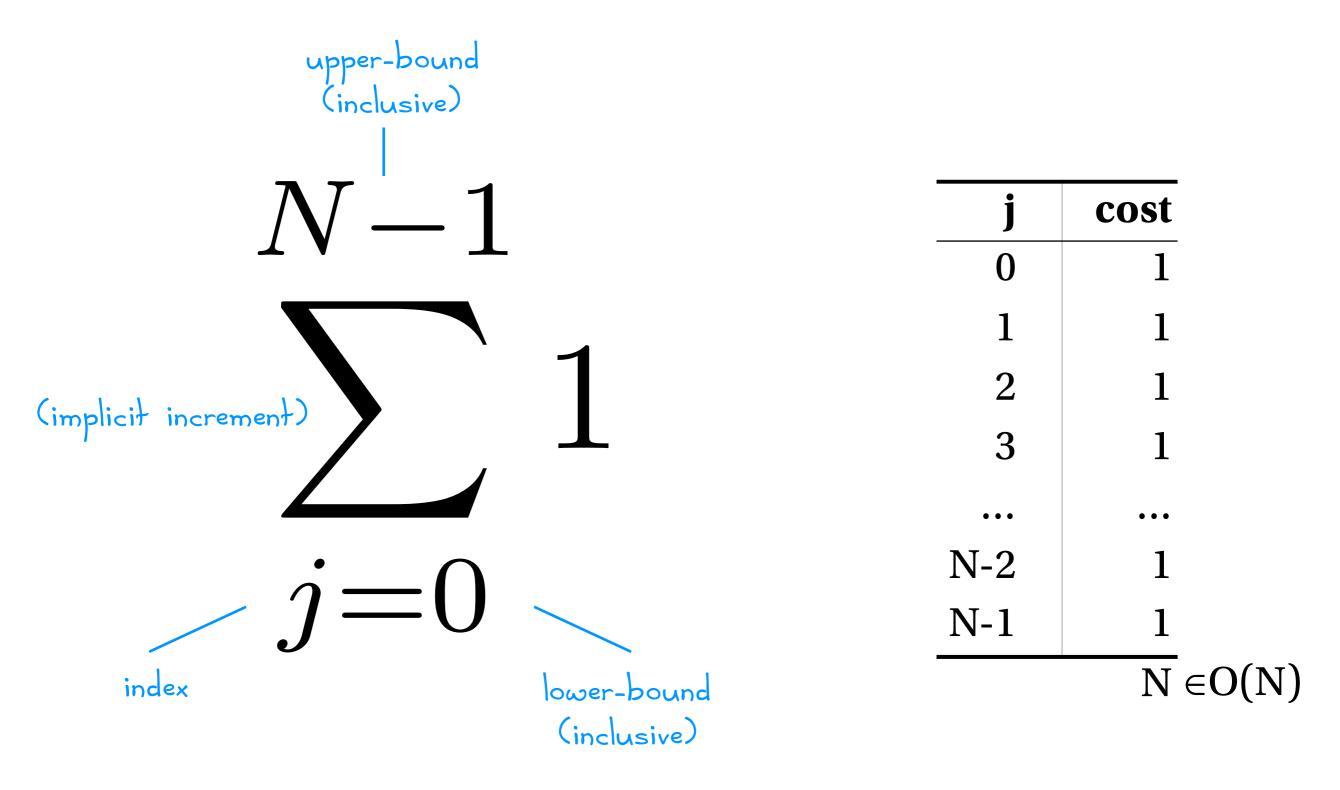
for j in range(N):
 platypus.quack()

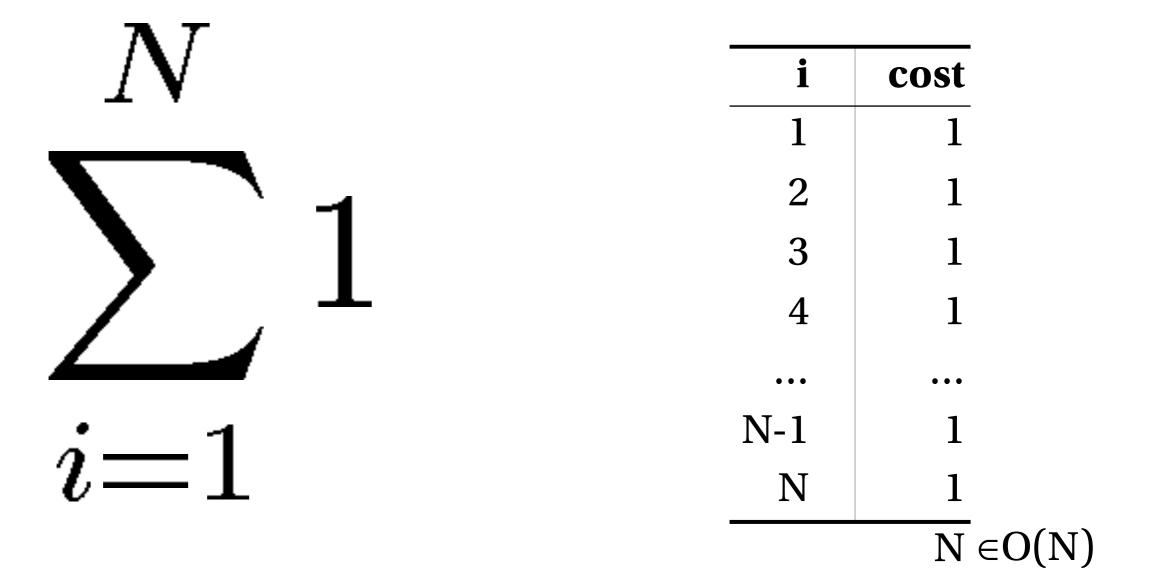


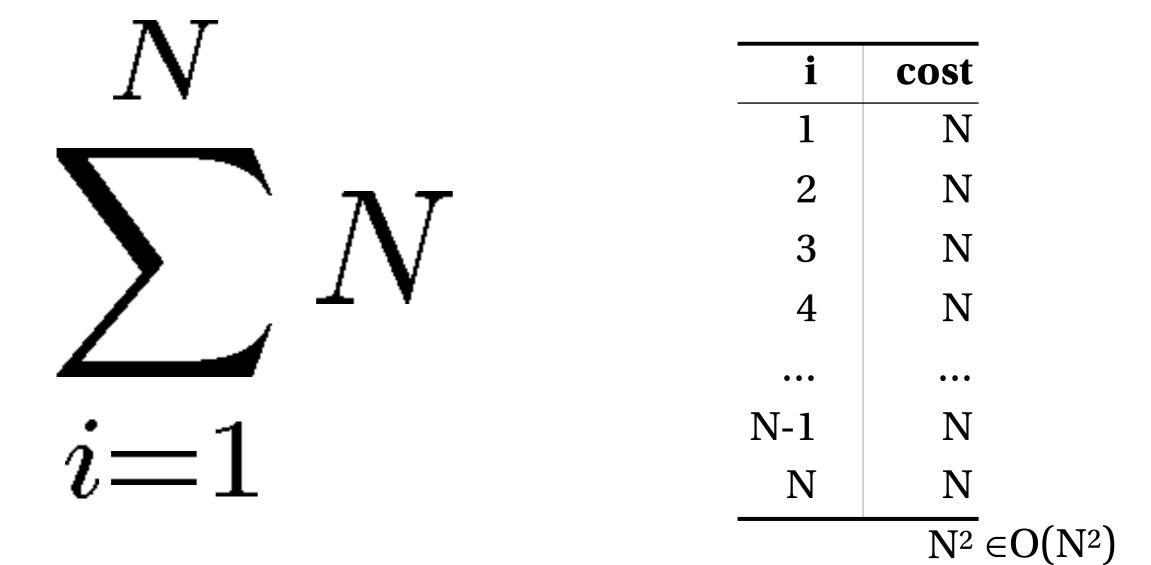
summations

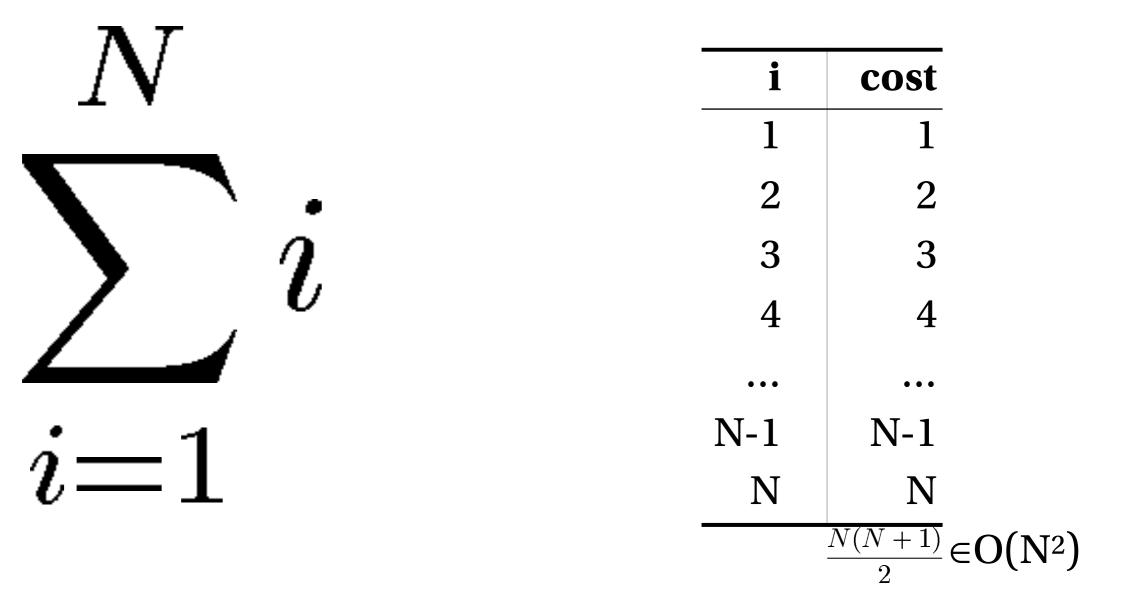
for j in range(N):
 platypus.quack()











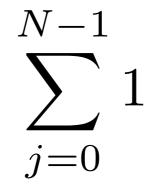
How many times does the platypus quack?

code

math

Wolfram Alpha

for j in range(N):
 platypus.quack()



sum 1,j=0 to N-1

 \mathcal{N}

closed form

asymptotic notation

O(N)

How many times does the platypus quack?

for i in range(N):
 for j in range(N):
 platypus.quack()

How many times does the platypus quack?

code

math

for i in range(N):
 for j in range(N):
 platypus.quack()

 $\sum_{i=0}^{N-1} \sum_{j=0}^{N-1} 1$

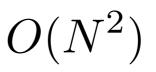
Wolfram Alpha

sum (sum 1,j=0 to N-1),i=0 to N-1

closed form

 N^2

asymptotic notation



How many times does the platypus quack?

for i in range(N):
 for j in range(i, N):
 platypus.quack()

How many times does the platypus quack?

for i in range(N):
 for j in range(i, N):
 platypus.quack()

 $\sum_{i=0}^{N-1} \sum_{j=i}^{N-1} 1$

Wolfram Alpha

code

math

sum (sum 1,j=i to N-1),i=0 to N-1

closed form

asymptotic notation

$$\frac{N(N+1)}{2}$$

$$O(N^2)$$