

Programming Workshop: More Algorithm Analysis, Python Timelt

Intermediate Workshop #7
31 October 2016

Outline

1. Speed Exercises
2. Quick review of last week's examples
3. Python timeit and examples

Speed Exercises

1:

Make a function that accepts two arrays and outputs the product of $A[i]$ and $B[i]$ as an array.

Analyze the run time.

240 seconds

Speed Exercises

2:

Make another function that accepts two arrays and outputs the product of $A[i]$ and $B[i]$ as an array.

Make it run in a *different* asymptotic time (preferably better).

240 seconds

“Big Oh” Notation - Example 1

Insertion Sort pseudocode:

```
for i ← 1 to length(A)-1
  j ← i
  while j > 0 and A[j-1] > A[j]
    swap A[j] and A[j-1]
    j ← j - 1
  end while
end for
```

Time to Run	Times Run
Constant	n total
Constant	1 per loop
Constant	1 per loop
Constant	$\leq j$ per while
Constant	$\leq j$ per while

Total: n total iterations through “for” loop, up to n total iterations through “while” $\rightarrow n*n = O(n^2)$

“Big Oh” Notation - Example 2

Merge Sort pseudocode:

(see Python file `merge_sort.py` -- too long to put in slides)

This is $O(n \lg(n))$

“Big Oh” Notation - Example 3

Counting Sort Pseudocode:

(see Python file `counting_sort.py` -- too long to put in slides)

This is $O(n + k)$ for fast array implementations.

Python Timeit

- Used to give average runtime of code over some range of runs
- Can be used within Python script or simply in terminal

Python Timeit - Example

- Sometimes the coefficients (usually truncated off largest term in asymptotic notations) matter.
- Take example 1: (see source)