Kevin Sekuj
1/11/2022
CS325: Analysis of Algorithms
Homework 1

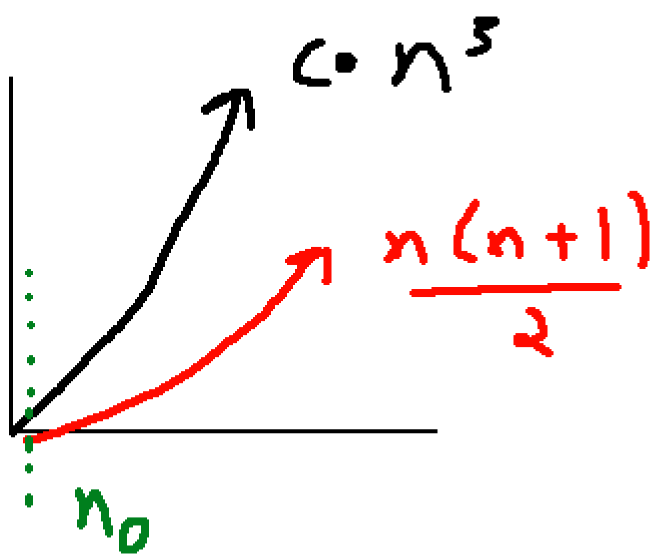
1. Identify and compare the order of growth

a. n(n+1)/2 ∈ O(n3)

True

$$\lim_{n\to\infty}\frac{\left(\frac{n(n+1)}{2}\right)}{n^3} \to \lim_{n\to\infty}\frac{\left(\frac{n^2+n}{2}\right)}{n^3} \to \lim_{n\to\infty}\frac{\left(\frac{n^2}{2}+\frac{n}{2}\right)}{n^3} \to \lim_{n\to\infty}\frac{\left(\frac{n^2}{2}+\frac{n}{2}\right)}{n^3} \to \lim_{n\to\infty}\frac{(n^2+n)}{2n^3}$$

$$\lim_{n\to\infty}\frac{(n^2+n)}{2n^3} \to \lim_{n\to\infty}\frac{n^2}{2n^3}+\frac{n}{2n^3} \to \lim_{n\to\infty}\frac{1}{2n}+\frac{1}{2n^2} = \boxed{0+0\ since\ \lim_{n\to\infty}\frac{1}{n} = 0}$$

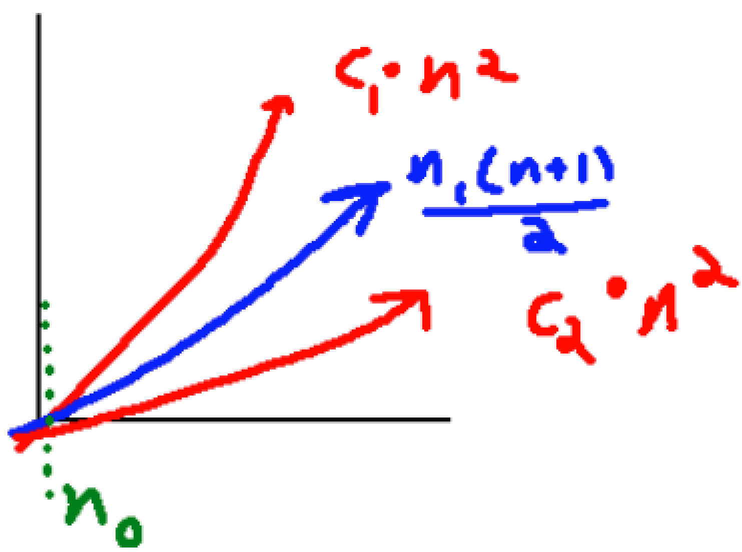


b. n(n+1)/2 ∈ Θ(n2)

True

$$\lim_{n\to\infty}\frac{\left(\frac{n(n+1)}{2}\right)}{n^2} \to \lim_{n\to\infty}\frac{\left(\frac{n^2+n}{2}\right)}{n^2} \to \lim_{n\to\infty}\frac{\left(\frac{n^2}{2}+\frac{n}{2}\right)}{n^2} \to \lim_{n\to\infty}\left(\frac{n^2+n}{2n^2}\right) \to \lim_{n\to\infty}\left(\frac{n^2}{2n^2}+\frac{n}{2n^2}\right)$$

$$\to \lim_{n\to\infty}\left(\frac{1}{2}+\frac{1}{2n}\right) = \frac{1}{2}+0 = \boxed{\frac{1}{2}, as \lim_{n\to\infty}\left(\frac{1}{n}\right) = 0}$$



c. 10n-6 ∈ Ω(78n + 2020)

The limit equates to a positive constant, so the functions have the same order of growth. Thus:

10n-6 ∈ Θ(78n + 2020)

which implies

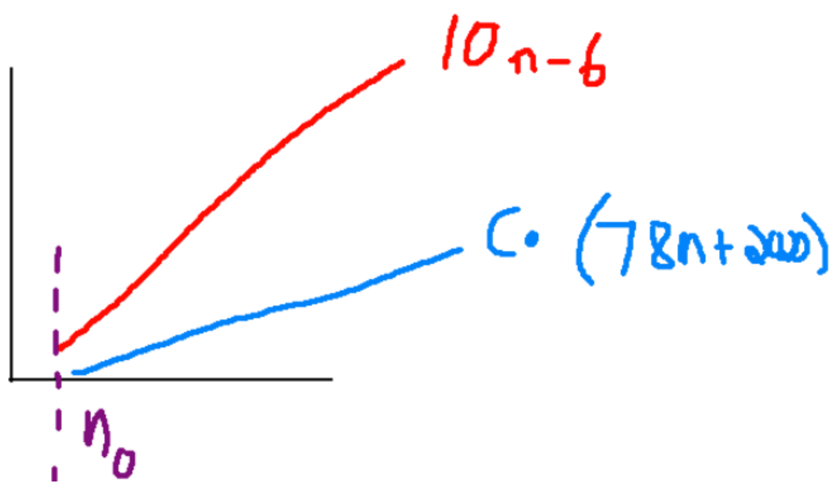
10n-6 ∈ Ω(78n + 2020)

and

10n-6 ∈ O(78n + 2020)

$$\lim_{n\to\infty}\left(\frac{(10n-6)'}{(78n+2020)'}\right) \rightarrow \lim_{n\to\infty}\left(\frac{10n'-6'}{78n'+2020'}\right) = \boxed{\frac{10-0}{78+0} = \frac{10}{78}}$$

(LH's formula)



d. n! ∈ Ω (0.00001n)

True

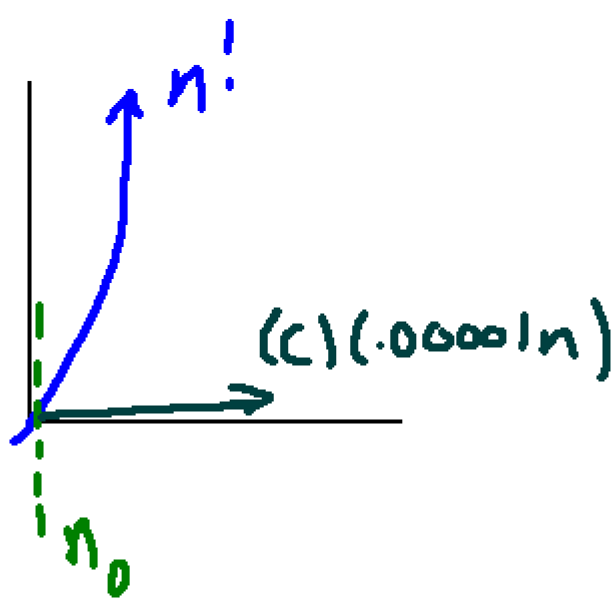
$$\lim_{n\to\infty} \frac{\sqrt{2\pi n}\frac{n}{e}^{n}}{.00001n} \to \lim_{n\to\infty} \frac{(100{,}000)(\sqrt{2\pi n}(\frac{n}{e}^{n}))}{n} \to \lim_{n\to\infty} \frac{(100000)\left(\sqrt{2\pi n}\right)\left(\frac{n}{e}\right)^{n}}{n} \to$$

$$\lim_{n\to\infty} \frac{(100000)\left(\sqrt{2\pi n}\right)\left(\frac{n}{e}\right)\left(\frac{n}{e}\right)^{n-1}}{n} \to$$

$$\boxed{\lim_{n\to\infty} (100000)\left(\sqrt{2\pi n}\right)\left(\frac{1}{e}\right)\left(\frac{n}{e}\right)^{n-1} = \infty}$$

(Stirling's formula)

hence n! ∈ Ω (0.00001n



2.

a.

The algorithm computes the difference between the array's maximum and minimum values.

b.

The basic operation are the two statements below the *for*-loop which compare the current element of the array to the stored minimum and maximum values.

```
if A[i] < minval:
    minval = A[i]
if A[i] > maxval:
    maxval = A[i]
```

c.

The basic operation executes *n* times for an array of length *n*.

d.

The time complexity of the algorithm is Θ(n) – this is because the algorithm executes n times in both the worst and best case.

3. a.

Loop invariant

At each iteration i of the while-loop on line 4, the subarray $A[i \dots j]$ consists of the elements originally in subarray $A[i \dots j]$, but in reversed order.

b.

Initialization

The loop invariant states: "…the subarray $A[i \dots j]$ consists of the elements originally in subarray $A[i \dots j]$, but in reversed order." Indeed, at the start of the first iteration, the array elements at indices i and j are swapped, or in other words, reversed. Thus, the subarray A[i...j] is reversed.

Maintenance

Assume that the loop invariant holds true at the start of iteration i. During this iteration, the elements at these indices have been swapped, and the pointers increment or decrement respectively. This means that at iteration i+1, there must exist a subarray *A*[i-1…j+1] such that the elements in that subarray have been reversed - which is what we needed to prove.

Termination

When the while-loop terminates, i = (n − 1) + 1 = n. Now, the loop invariant states: "the subarray $\mathrm{A}[\mathrm{i} \dots \mathrm{j}]$ consists of the elements originally in subarray $\mathrm{A}[\mathrm{i} \dots \mathrm{j}]$, but in reversed order." This is exactly what the algorithm should output, which it then outputs. So, the algorithm is correct.