

# More on Asymptotic Analysis

December 8, 2021

# Administrative Notes

- Project 3 due next Monday, December 13, before midnight
  - Submit the .ipynb file on gl.umbc.edu just as you always have
    - submit cmisc201 PROJECT 3 project3.ipynb
  - Jupyter notebook files do NOT end in .py; they end in .ipynb
    - It used to be called “iPython notebooks” before Project Jupyter took it over
- Final exam the following Monday: December 20, 3:30 - 5:30 pm, in the regular classroom
- I'll definitely be in Colorado next Monday - I'll send out a notice about when I'll be on-line live
  - I am available via Discord, e-mail,... if you have questions
- There are both a sample exam and a file with the answers available on the class GitHub as of last night

# First a correction

Talk about an oversight:

$\sum i$  from 1 to  $n = n*(n+1)/2$  ; NOT  $n*(n-1)/2$

It doesn't change the central point - when talking about big-O notation, we throw away all but the largest polynomial term, but it's an embarrassing mistake to make. My first calculus teacher is waiting to talk to me about this.

# Asymptotic run-time

From:

<https://xlinux.nist.gov/dads/HTML/asymptoticTimeComplexity.html#:~:text=Definition%3A%20The%20limiting%20behavior%20of,denoted%20in%20big%2DO%20notation.>

**Definition:** The limiting behavior of the execution time of an [algorithm](#) when the size of the problem goes to infinity. This is usually denoted in [big-O notation](#).

Asymptotic literally means “the value of an expression as the value tends toward infinity.”

A common mathematical concept

# HUMONGOUS Data Sets

- About 500 million tweets per day
- 30,000 hours of video uploaded to YouTube per hour; 720,000 hours of video per day
- 95 million Instagram posts per day

... you get the picture. There are some big data sets

We really need algorithms that run fast on these big data sets

# Comparison of algorithms

There are two ways to compare sorting and searching algorithms: by the number of comparisons and swaps they involve; or by the time they take

We'll look at our algorithms both ways

- Comparisons and swaps
- time