Intro and Algorithmic Timing

N. Lutz
MIT Media Lab
Object Based Media

Algorithms Are For Everyone

What is an algorithm

A process or set of rules to be followed in calculations or other problem-solving operations, especially by a computer.

Essentially, how does a computer do xyz.

Who am I?

Grad student here at OBM.

Have TAed and taken various courses in Computer Science, specifically in computational for not computer science and algorithms

MIT gave me a B.Sc in Computer Science.

What an algorithm is NOT

Code - The algorithm is the instructions, not the software. However, some are often implemented in code and all can be written in **pseudocode.**

Math - some algorithmic paths can be considered a branch of mathematics but in reality they are **logic.**

Restricted to Computer Science — many disciplines have them.

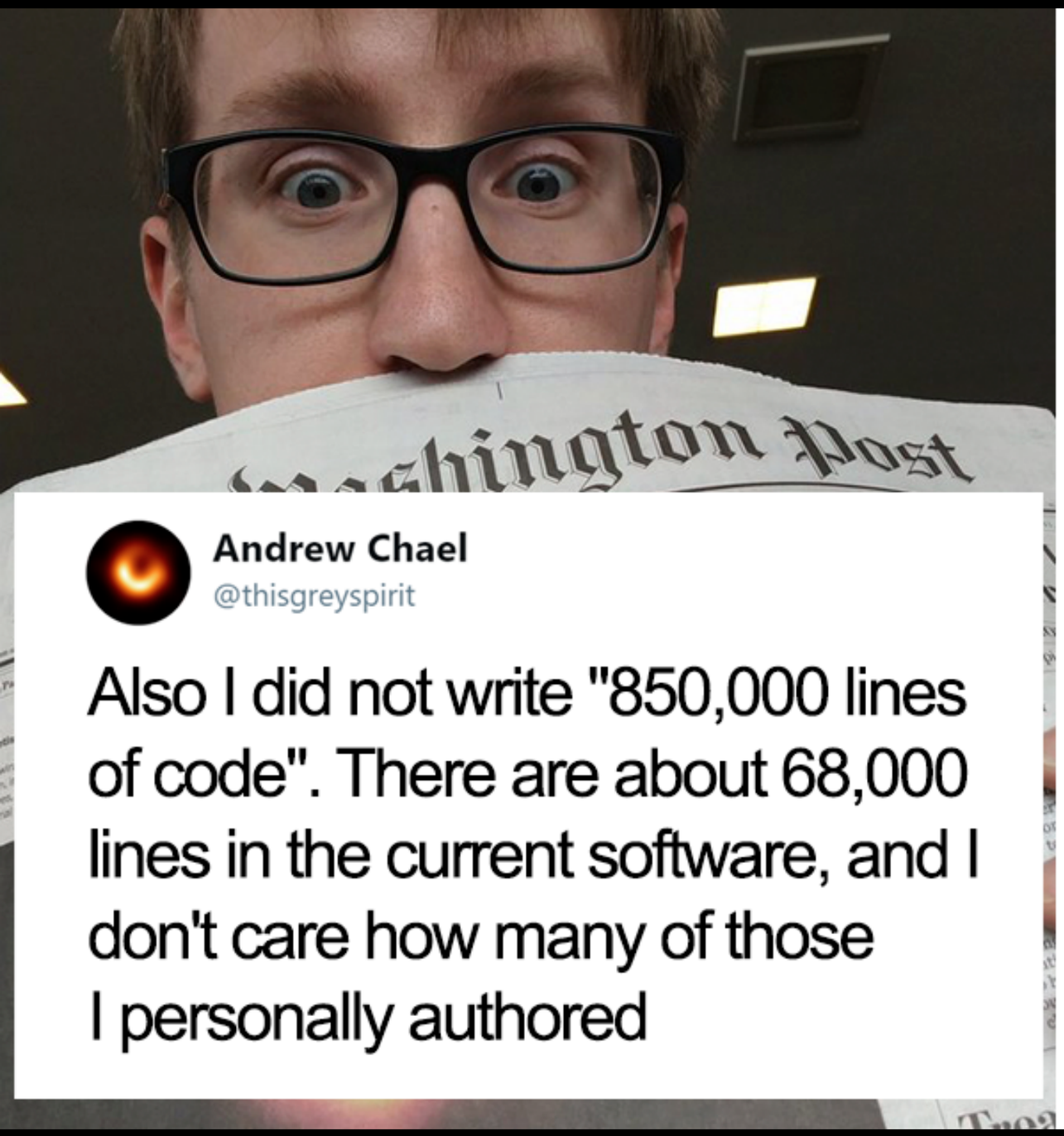
Computer Science vs Software Engineering

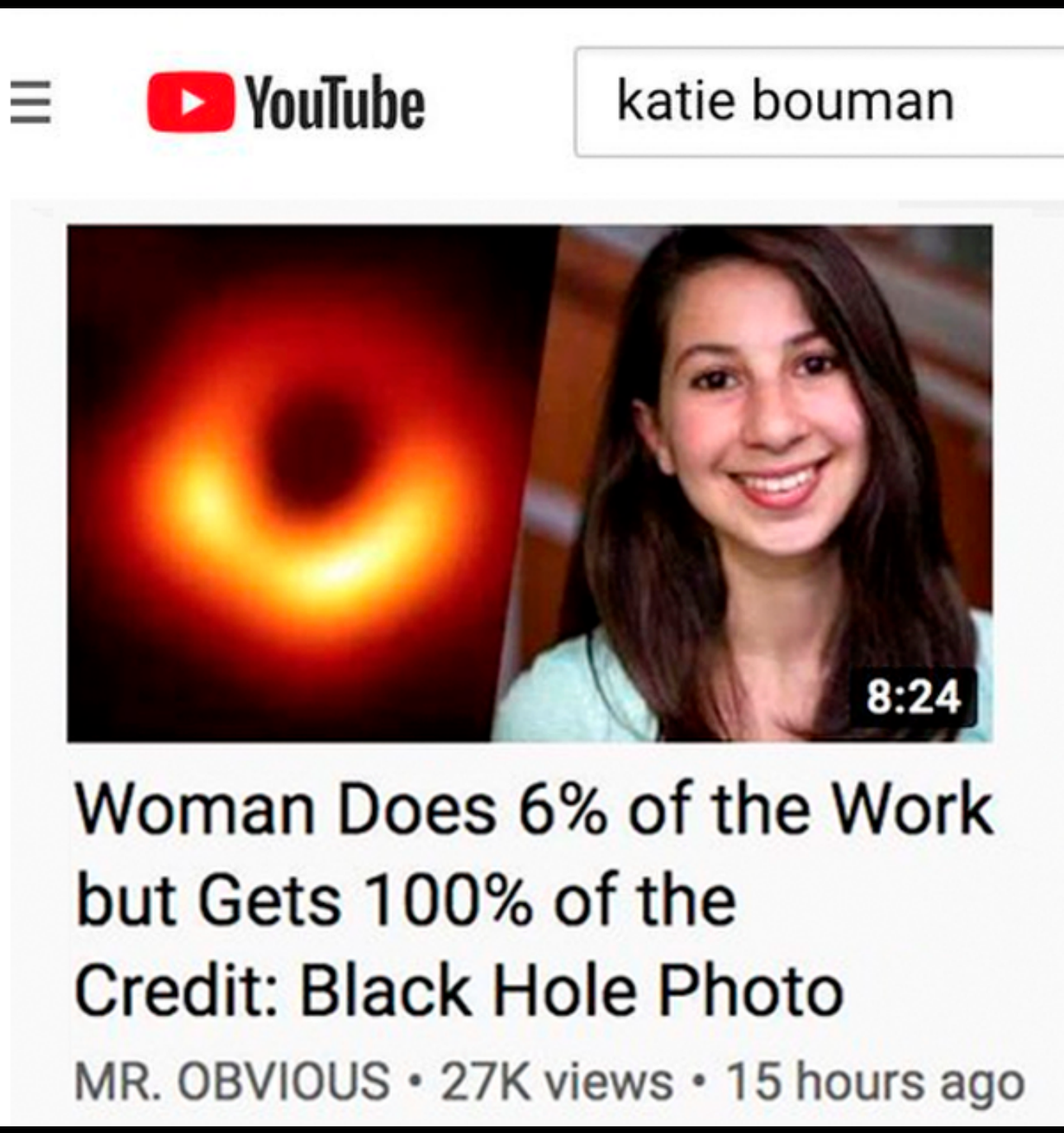
Software engineering — building working software for a variety of devices and purposes.

Computer science — the study of how data interacts in digital form.

Software engineering is part of computer science skillsets, but algorithms are not necessarily software engineering — but they help!

Thus, algorithms are not super verbose.
And in computer science line count is not
a good way to count anything.





Pseudocode

A way to write instructions that looks like code, but that probably wouldn’t be understood by a computer as code.

Example

- I have 12 cupcakes, I want to frost all of them.
- I want to alternate the colors of two frostings: red and blue.
- I can only frost one cupcake at a time and I have to wait 5 seconds between each frosting.

Pseudocode for cupcakes

```
count
for each cupcake:
   if count even: frost red
   if count odd: frost blue
   return cupcake
   delay 5 seconds
```

Example pseudocode:

```
dist[s] ←o
for all v ∈ V–{s}
      do dist[v] ←∞
S←∅
Q←V
while Q ≠∅
do   u ← mindistance(Q,dist)
     S←S ∪{u}
      for all v ∈ neighbors[u]
            do if  dist[v] > dist[u] + w(u, v)
                        then     d[v] ←d[u] + w(u, v)


return dist
```

Some Algorithm Families

Often divided by data structure or by method of the algorithm

There are so many algorithms go crazy!

https://en.wikipedia.org/wiki/List_of_algorithms

Efficiency

Algorithms are great at instructions
— but they also need to be efficient
in timing and often in space.

Algorithmic Time

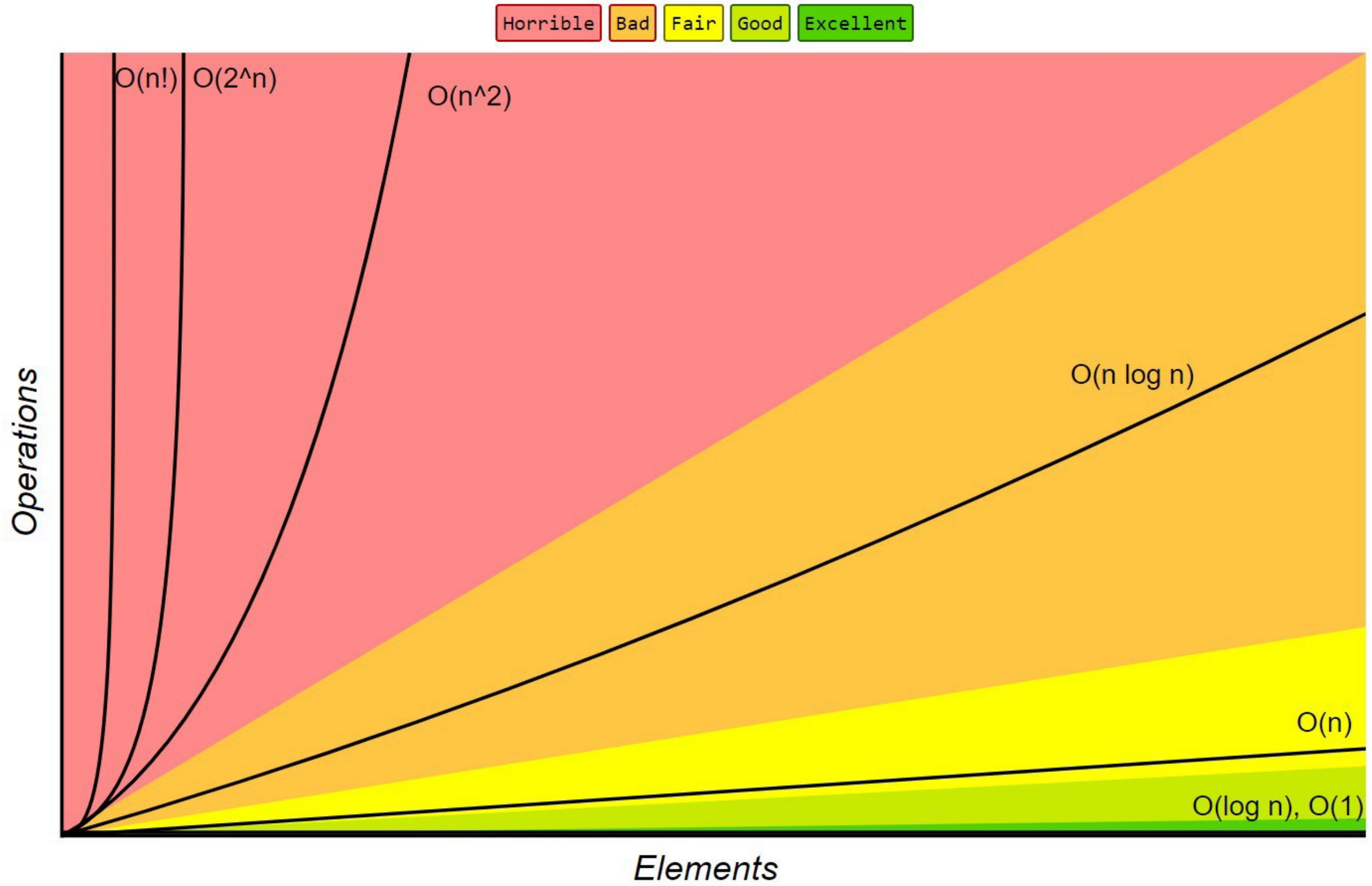
Big O notation
Time complexity
Runtime analysis

The computational complexity to run a process.

Big O Notation

- O(n) is a linear time algorithm that takes n bits of logic. n is usually the number of items and other constants have to do with the operations within it.
    - For example, our frosting algorithm takes n cupcakes with one process. Technically with our 5 second delay its O(5n) but it is still linear.

Big-O Complexity Chart



Good rule of thumb

- You can usually denote how long an algorithm will take with its pseudo code based on:
  - How many elements you’re performing operations on
  - How long those operations take
  - Any comparisons or duplicities of these operations

Example of O(n^2)

- You have n cupcakes, each is frosted red or blue and has a label on the bottom.
- You want to frost all of them, this is O(n)

- Now you have to go through each cupcake and see if there are duplicates. A naive way of doing this is O(n^2) — for each cupcake's label you check it against every other cupcakes. You do n checks per cupcake, which is n*n or n^2.

Good resources

- Time complexity lessons
- 6.006 OCW
- Next week I talk about sorting algorithms.