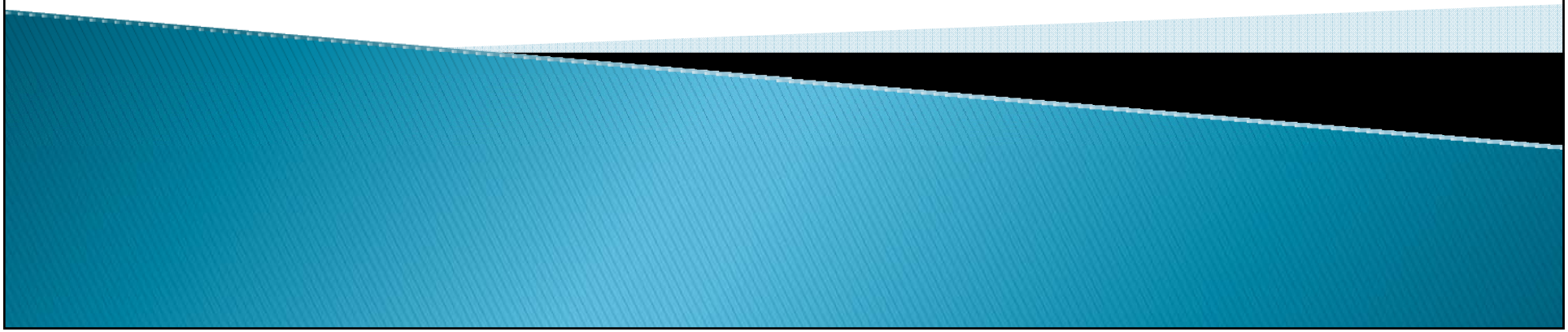
CSSE 220 Day 14

More algorithm efficiency analysis, Big-Oh
Work on Paint

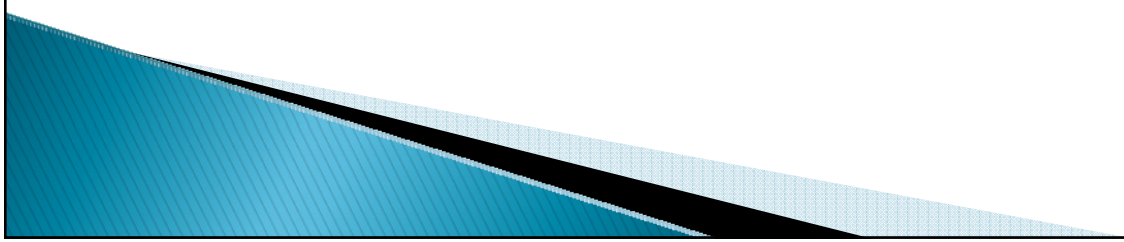


CSSE 220 Day 14

- BallWorlds grades are on ANGEL, and scoring sheets handed back.
- Another progress Report/IEP due today at the end of class.
- Today:
  - Algorithm analysis and Big-Oh
  - More Paint time
- Questions?

Questions on Angel part of exam?

- Or on programming part?
- This is all fair game for the Final Exam, only 6 short weeks away (Monday, 6pm in Crapo G310).



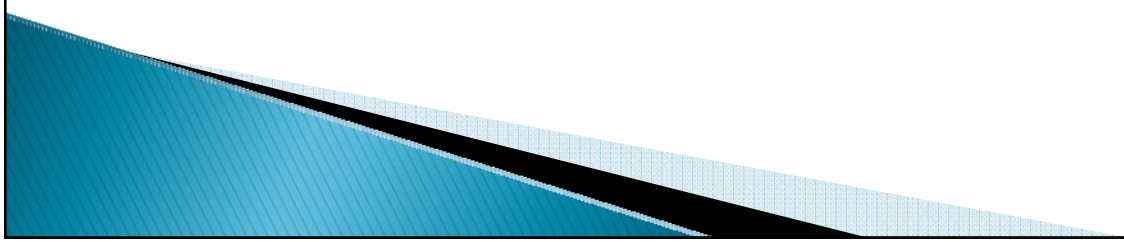
Key Concepts quiz tomorrow

- It is important that we not only be able to write object-oriented programs, but that we build a vocabulary that enables us to communicate with each other about them.
- That is why we asked you to spend four weeks learning the "lingo" of OOP in Java.
- Tomorrow is the check-up on that.
- This ANGEL-based quiz is closed book and notes.
- It consists of matching questions, and you will only have about 30 seconds per term to complete it. So know your terms well!

Measuring program effciency

- What kinds of things should we measure?
  1. CPU time
  2. memory used
  3. disk transfers
  4. network bandwidth

- Mostly in this course, we focus on the first two, and especially on CPU time.



Some simple efficiency tips

- If a statement in a loop calculates the same value each time through, move it outside the loop
- Store and retain data on a “need to know” basis.
- Don’t store what you won’t reuse!
  - Do store what you need to reuse!
- Don’t put everything into an array when you only need one or two consecutive items at a time.
- Don’t make a variable be a field when it can be a local variable of a method.

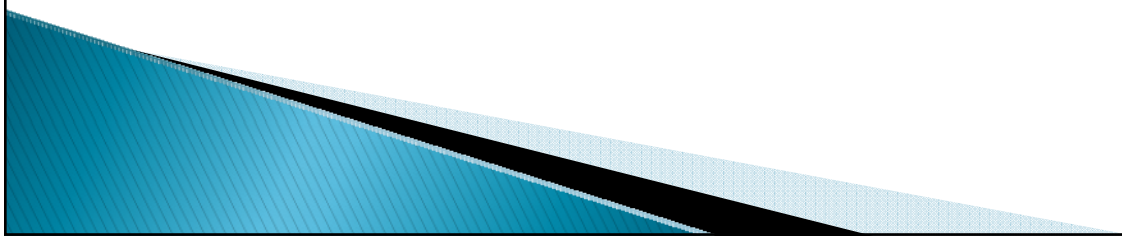
An example of running time

- How can we measure running time?
- System.currentTimeMillis
- Run Sieve example.
- When do we really care about efficiency?
- Can we get a rough idea of efficiency by analyzing algorithms on paper?

Familiar example:
Linear search of a sorted array of Comparable items

```
for (int i=0; i < a.length; i++)
  if ( a[i].compareTo(soughtItem) > 0 )
    return NOT_FOUND;
  else if ( a[i].compareTo(soughtItem) == 0 )
    return i;
return NOT_FOUND;
```

- What should we count?
- Best case, worst case, average case?



Another algorithm analysis example

Does the following method actually create and return a copy of the string `s`?

What can we say about the running time of the method?
(where N is the length of the string `s`)

What should we count?

```
public static String stringCopy(String s) {
  String result = "";
  for (int i=0; i<s.length(); i++)
     result += s.charAt(i);
  return result;
}
```

Don't be too quick to make assumptions when analyzing an algorithm!

How can we do the copy more efficiently?

Figure 5.1

Running times for small inputs

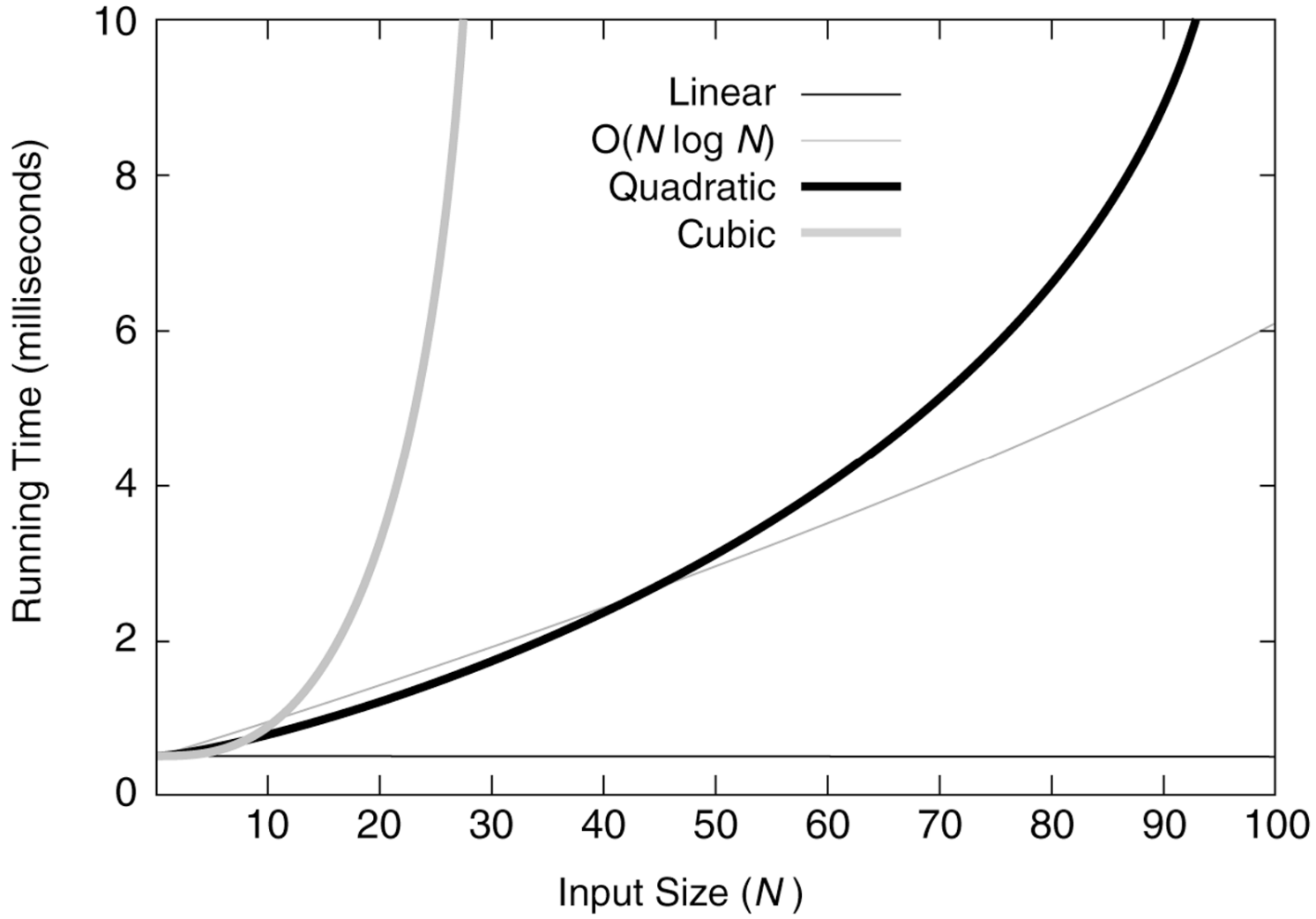


Figure 5.2

Running times for moderate inputs

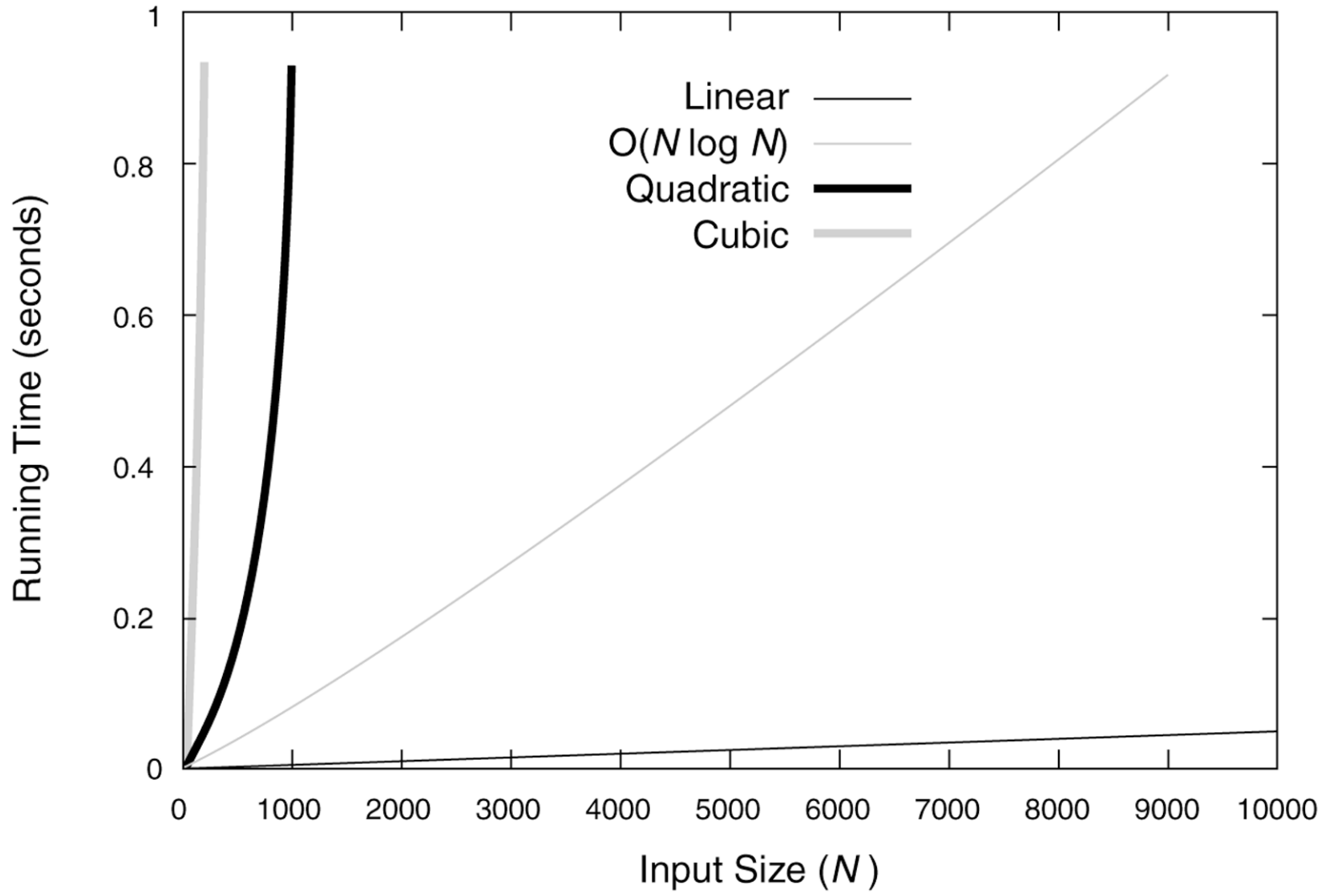


Figure 5.3

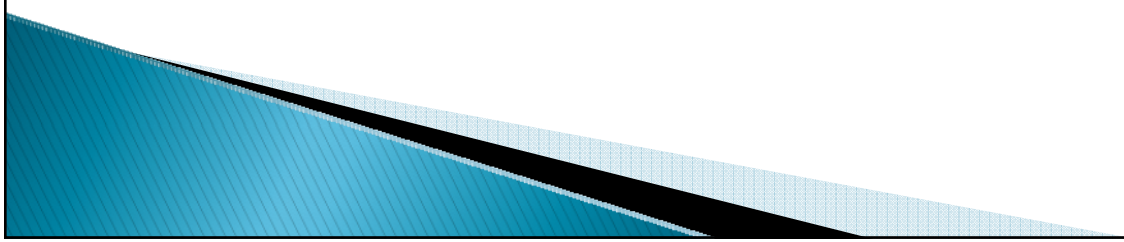
Functions in order of increasing growth rate

Function	Name
c	Constant
$\log N$	Logarithmic
$\log^2 N$	Log-squared
N	Linear
$N \log N$	N log N
N^2	Quadratic
N^3	Cubic
2^N	Exponential

a.k.a "log linear"

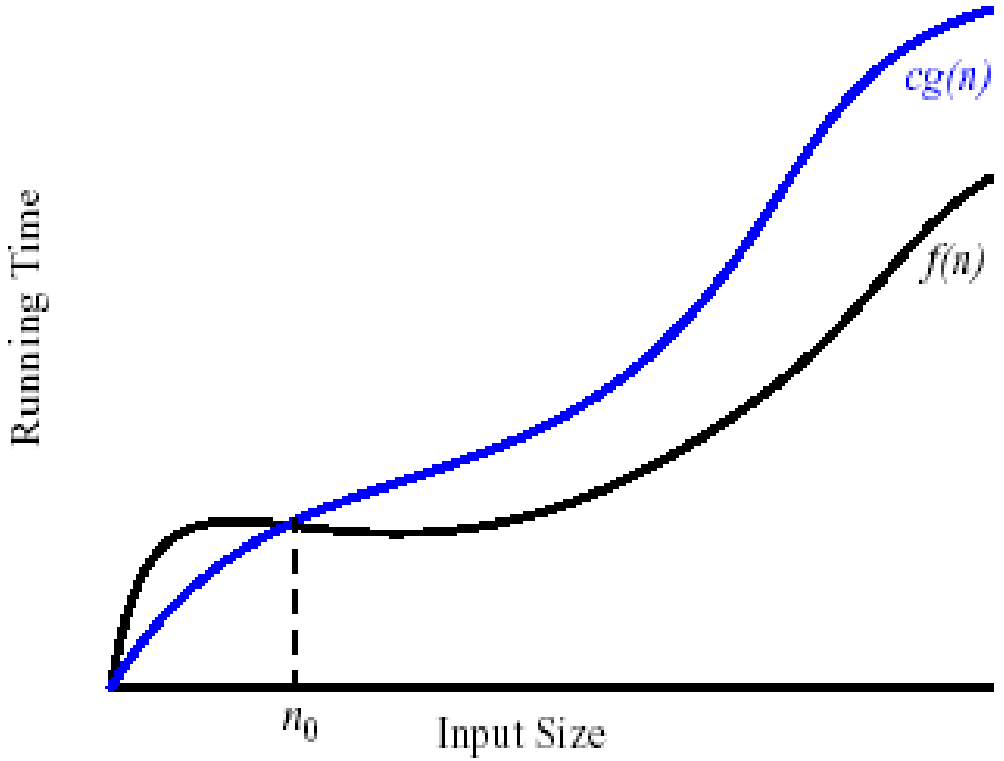
Asymptotic analysis

- We only really care what happens when N (the size of a problem) gets large.
- Is the function linear? quadratic? etc.



- The "Big-Oh" Notation
  - given functions f(n) and g(n), we say that f(n) is O(g(n)) if and only if f(n) ≤ c g(n) for $n \geq n_0$
  - c and n_0 are constants, f(n) and g(n) are functions over non-negative integers

In this course, we won't be so formal .
We'll just say that
f(N) is O(g(N) means that f(n) is eventually smaller than a constant times g(n).



- Simple Rule: Drop lower order terms and constant factors.
  - $7n - 3$ is $\boldsymbol{O}(n)$
  - $8n^2 \log n + 5n^2 + n$ is $\boldsymbol{O}(n^2 \log n)$

- Special classes of algorithms:
  - logarithmic: $\boldsymbol{O}(\log n)$
  - linear $\boldsymbol{O}(n)$
  - quadratic $\boldsymbol{O}(n^2)$
  - polynomial $\boldsymbol{O}(n^k)$, $k \geq 1$
  - exponential $\boldsymbol{O}(a^n)$, $n > 1$

- "Relatives" of the Big-Oh
  - $\Omega(f(n))$: Big Omega
  - $\Theta(f(n))$: Big Theta

Limits and asymptotics

- consider the limit $$\lim_{n \to \infty} \frac{f(n)}{g(n)}$$
- What does it say about asymptotics if this limit is zero, nonzero, infinite?
- We could say that knowing the limit is a sufficient but not necessary condition for recognizing big-oh relationships.
- It will be sufficient for all examples in this course.

Apply this limit property to the following pairs of functions

Assume a and b are constants.

- N and N^2
- $N^2 + 3N + 2$ and N^2
- $N + \sin(N)$ and N
- $\log N$ and N
- $N \log N$ and N^2
- N^a and N^n
- a^N and b^N $(a < b)$
- $\log_a N$ and $\log_b N$ $(a < b)$
- $N!$ and N^N

Big-Oh Style

- **Give tightest bound you can**
  - Saying that $3N+2$ is $O(N^3)$ is true, but not as useful as saying it's $O(N)$ [What about $\Theta(N^3)$?]
- **Simplify:**
  - You *could* say:
  - $3n+2$ is $O(5n-3\log(n) + 17)$
  - and it would be technically correct…
  - It would also be poor taste … and put me in a bad mood.
- **But… if I ask "true or false: $3n+2$ is $O(n^3)$", what's the answer?**
  - True!
  - There may be "trick" questions like this on assignments and exams.
  - But they aren't really tricks, just following the big-Oh definition!

Rest of Class

- Please provide me with some mid-term feedback on how the course is going
  - Angel > Lessons > Course Discussion Forums > Midterm Plus/Delta Feedback
- Thanks!
- After that you have all class to work on Paint with your partner.
- Please commit your updated IEP at the end of class.