Lecture 8: Growth Rates

CS442: Great Insights in Computer Science
Michael L. Littman, Spring 2006

Comparing Algorithms

- Today, we'll talk about how computer scientists compare algorithms to decide which is better.
- We'll apply this idea to the sock sorters from last time.
- But first, an aside to introduce some mathematical concepts.

Song Growth

- Let's talk about how my syllables we sing given a song of a certain type as the number of verses grows.
- In general, we're interested in the number of syllables as a function of *n*, the number of verses.

Generalized Dreidel Song

1. I had a little dreidel
I made it out of clay
And when it's dry and ready
Oh dreidel I shall play.

Chorus:
Oh dreidel dreidel dreidel
I made it out of clay
And when it's dry and ready
Oh dreidel I shall play.

2. I had a little dreidel
I made it out of plastic
If someone steals my dreidel
I'll do something very drastic.

Chorus

3. I had a little dreidel,
I made it out of glass
My mom said when I spin it,
to spin it on the grass.

Chorus

4. I had a little dreidel,
I made it out of chocolate,
but when I went to spin it,
it had melted in my pocket.

Chorus

5. I had a little dreidel,
I made it out of wood,
and when I went to spin it,
it spun just like it should.

Chorus

6. I had a little dreidel,
I made it out of ice,
but when I went to spin it,
it melted...that's not nice!!

Chorus

7. I had a little dreidel,
I made it out of mud,
and when I went to spin it,
it fell down with a thud.

Chorus

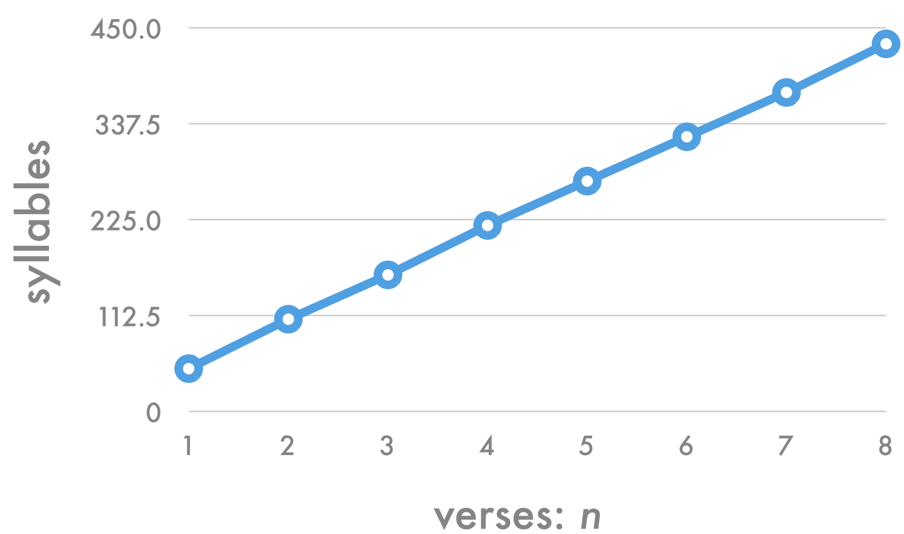
8. I had a little dreidel,
I made it out of tin,
I made it kind of crooked,
and so I always win.

Chorus

Counting Syllables

verses	syllables	verses	syllables
1	51	5	271
2	109	6	323
3	161	7	375
4	219	8	432

Plotting Syllables



- Total syllables roughly, $T(n) = 54\,n$.

Old Macdonald: Verse 4

Old Macdonald had a farm, E-I-E-I-O And on his farm he had a chick, E-I-E-I-O	25 syllables
With a "cluck, cluck" here and a "cluck, cluck" there Here a "cluck" there a "cluck" Everywhere a "cluck-cluck"	22 syllables
With a "neigh, neigh" here and a "neigh, neigh" there Here a "neigh" there a "neigh" Everywhere a "neigh-neigh"	22 syllables
With a (snort) here and a (snort) there Here a (snort) there a (snort) Everywhere a (snort-snort)	22 syllables
With a "moo-moo" here and a "moo-moo" there Here a "moo" there a "moo" Everywhere a "moo-moo"	22 syllables
Old Macdonald had a farm, E-I-E-I-O	12 syllables

37 + 4 x 22

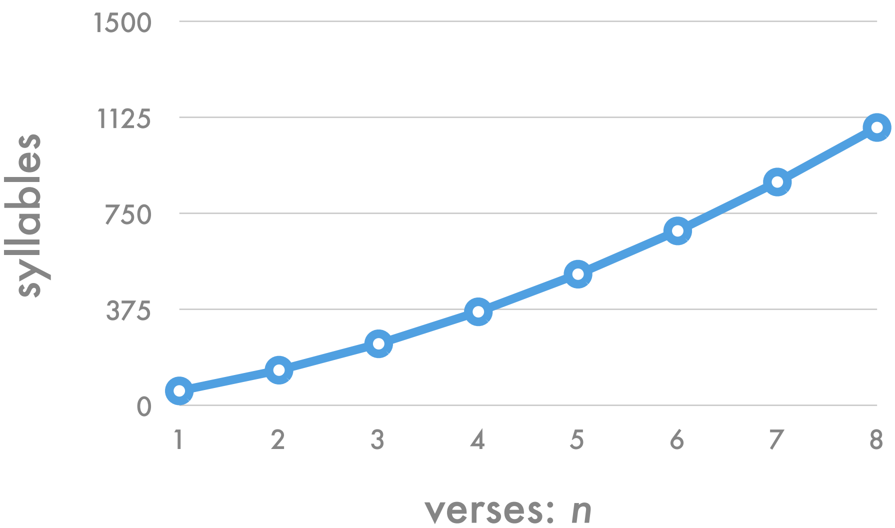
= 125 syllables

4-verse song: (37 + 1 x 22) + (37 + 2 x 22)
+ (37 + 3 x 22) + (37 + 4 x 22) = 368 syllables

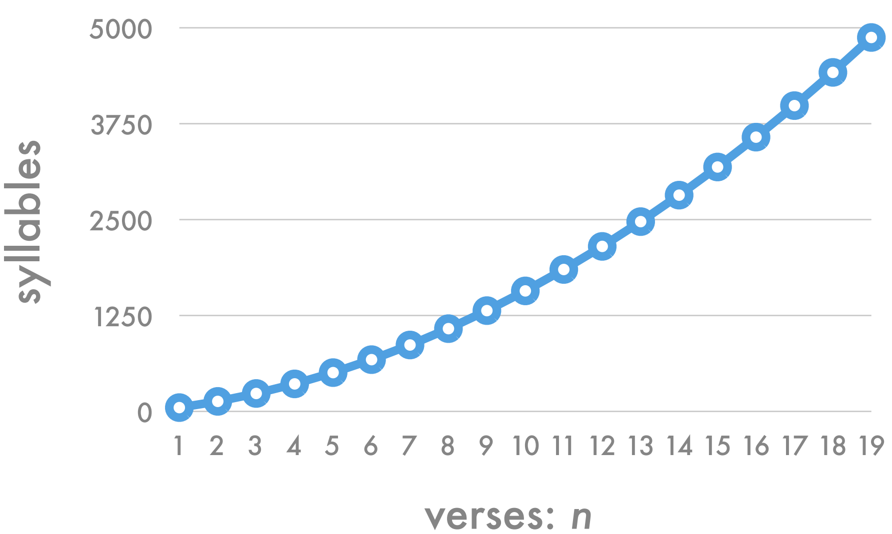
Counting Syllables

verses	syllables	verses	syllables
1	59	5	515
2	140	6	684
3	243	7	875
4	368	8	1088

Plotting Syllables



Plotting More Syllables



- Total syllables?

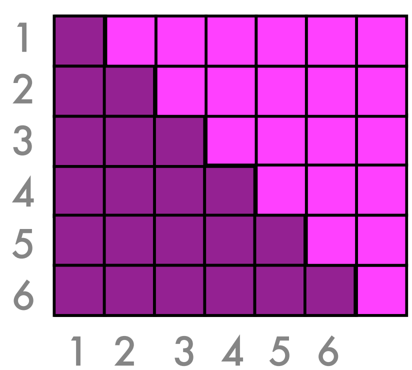
Summing Syllables

- Verse i has 37 + 22 x i syllables.
- Song with n verses:

$$(37 + 1 \times 22) + (37 + 2 \times 22) + \ldots + (37 + n \times 22)$$

$$37\,n + (1 + 2 + \ldots + n) \times 22$$

Sum of n Integers



$1 + 2 + \ldots + 6 =$
$(6 \times 7) / 2 = 21$
$1 + 2 + \ldots + n =$
$n \times (n+1) / 2$

- Old McDonald with n verses:

$$37n + 22 \times (1 + 2 + \ldots + n) = 11\,n^2 + 48\,n$$

N Bottles of Beer

- Verse 99:
- Verse *i*:

99 bottles of beer on the wall.
99 bottles of beer.
If one of those bottles should happen to fall.
98 bottles of beer on the wall.

29 syllables + 2 x syllables in *i* + syllables in *i*-1.

- Syllables in *i*?
  - Roughly the number of digits in *i*.
  - Very slow growing function... by googol, only reaches 101.

Logarithms

- lg 100 = 2
- lg 10000 = 4
- lg *x*: roughly the number of times you can divide *x* by 10 before you reach 1 or less.
- Other logs:
  - ln is the natural logarithm (base *e*)
  - log is base 2 logarithm: number of times you can halve before reaching 1 or less.

Whole Song

- So, syllables in verse i of N Bottles of Beer:
  - Approximately, $29 + 3 \lg i$.
- n verses: $29\,n + 3\,(\underbrace{\lg 1 + \lg 2 + \ldots + \lg n}_{n})$
  - 90% of sum is lg n, 9% is $\lg n - 1$, 0.9% is $\lg n - 2$, 0.09% is $\lg n - 2$, ...
  - Approximately, $2.8\,(n \lg n) + 29\,n$.

90s
80s
70s
60s
50s
40s
30s
20s
10s
0s

N Days of Christmas

- Verse 10:
- Verse i:

On the tenth day of Christmas,
my true love sent to me
Ten lords a-leaping,
Nine ladies dancing,
Eight maids a-milking,
Seven swans a-swimming,
Six geese a-laying,
Five golden rings,
Four calling birds,
Three French hens,
Two turtle doves,
And a partridge in a pear tree.

11 syllables + (4 + syllables in i) + (4 + syllables in i-1) + (4 + syllables in i-2) + ... + (4 + syllables in 1).

- Approx., $11 + 4\,i + \lg 1 + \lg 2 + \ldots + \lg i$
- Approx., $11 + 4\,i + i \lg i$.

Whole Song

- n verses:

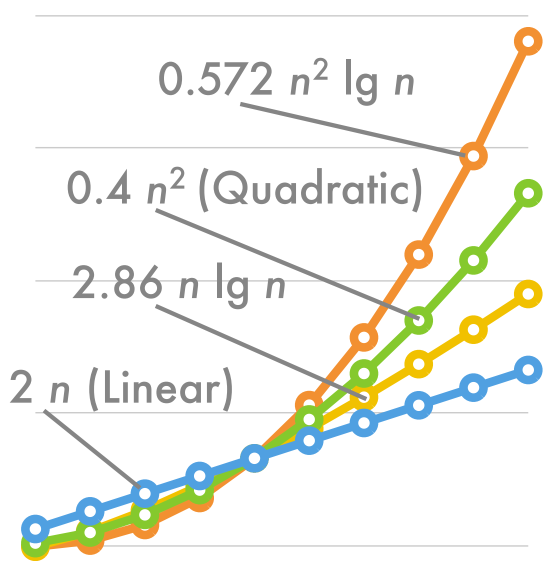
  $(11 + 4 + \lg 1) + (11 + 8 + 2 \lg 2) +$

  $(11 + 12 + 3 \lg 3) + \ldots + (11 + 4\,n + n \lg n)$

- Approx., $11\,n + 2\,n(n+1) + .46\,n(n+1)\ \lg n$.
- Approx., $.46\,n^2 \lg n + 2\,n^2 + .46\,n \lg n + 13\,n$.

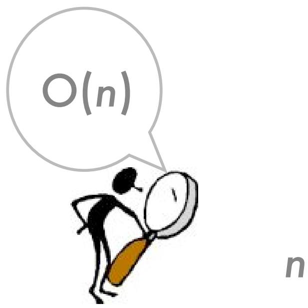
Different Growth Rates

- With these constants, $n^2 \lg n$ has fastest growth, then quadratic, then $n \lg n$, then linear.
- For big n, always the same order regardless of the constants!
- Leads to the notion of "Big O".



Big O

- Formally, big O is a notation that denotes a class of functions all of which are upper bounded asymptotically.



- In practice, however, it gives us a way of ignoring constants and low-order terms to cluster together functions that behave similarly.

Common Growth Classes

- Linear: $O(n)$
  - Dreidel
  - Clementine

constant size verse

- Quadratic: $O(n^2)$
  - An Old Lady
  - Old Macdonald
  - There Was a Tree

Each verse a constant size larger than the previous

- $O(n \log n)$
  - *N* Bottles of Beer
  - *N* Little Monkeys

Each verse contains the next higher number

- $O(n^2 \log n)$
  - *N* Days of Christmas
  - Who Knows *N*?

Each verse lists one more number than the previous

Non-Classical Songs

- As far as I know, classical songs are all linear ($O(n)$), quadratic ($O(n^2)$), $O(n \log n)$, and $O(n^2 \log n)$. [Extra credit for discovering another type!]
- Nevertheless, I can make up a few more songs to demonstrate a few other important growth rates.

Skip A Few...

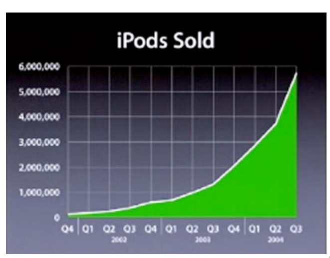
- My kids used to play this game: "I can count up to 100. One, two, skip-a-few, 99, 100!". Or "One, two, skip-a-few, 999, 1000!".
- Number of syllables to "skip count" to n?
  - $5 + 2 \lg n$: This song is $O(\lg n)$.
- With exponential notation: "One, two, skip-a-few, 10^{100}-1, 10^{100}. Now, the syllables depend on the number of digits: $O(\lg \lg n)$.

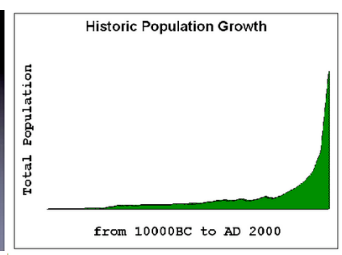
Very Slow Song

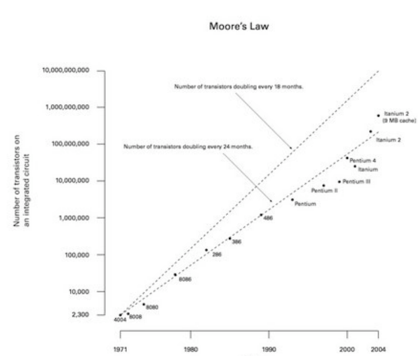
- On the flip side, consider a song in which verse *i* consists of singing all the numbers with exactly *i* digits.
- Now, a song with n verses is $O(10^n)$.
- This is an *exponential* growth. Something I'd like to say a bit more about.

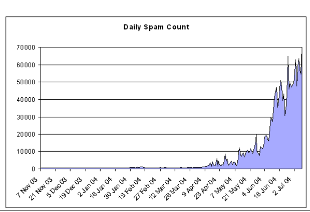
Exponential Growth

- ipods.
- Computer speed: Moore's Law.
- World Population.
- Bacterial growth (while the food lasts).
- Spam.









Pet Peeve Alert

- Because exponential growth rates are so common, the phrase has entered the public lexicon.
- Not always properly... Many people seem to use it to mean “a lot more”, which doesn’t really make sense.
- Let’s learn to recognize the proper use, ok?

Which Are Correct?

“exponential equation” summarizes the relationship

- Source: Newsw

- The country desperately needs to upgrade its roads and seaports, and to **exponentially** increase agricultural and manufactured exports.
  - $\text{exports}(t) = 10^t$
- **Exponentially** less expensive than a 20-hour flight to the Bushveld of South Africa or the remote rain forests of Costa Rica, domestic safaris can be nearly as exciting—and far more accessible for families with kids.

- e demands on an ganization to carry out [multiple] attacks like that probably increase **exponentially**. In other words, to carry out four simultaneous bombings is more difficult than simply just four times the difficulty of carrying out one bombing.
  - $\text{difficulty}(\text{targets}) = 10^{\text{targets}}$
- But a small number of others, knowing that their chance of success with PGD is **exponentially** better, are becoming pioneers in the newest form of family planning.

Continued

- Demand for IVF treatments, which climbed **exponentially** during the past 20 years, has plateaued.
  - demand(t) = 10^t
- Consequently, an unintended but **exponentially** growing number of middle-class Americans is being affected.
  - affectedpeople(t) = 10^t
- I have been on television for almost 12 years, and in that relatively short time I've seen the medium change **exponentially**.
- Now in the tsunami's aftermath, global health experts worry that the dangerous microbes already lurking in underdeveloped regions of Asia will spread **exponentially**, pushing the tsunami's enormous death toll even higher.
  - affectedArea(t) = 10^t
- Injury rates [for cheerleaders] are "**exponentially** higher for a flier than for a footballer," says NCCSI's Robert Cantu.

Analyzing Sock Sorting

- How many calls to getSocks does sorter1 take to sort 50 pairs of socks?
- sorter1: choose a random pair. Return to basket if no match.
- Number of getSocks before a pair is found?
- Probability of a match is 1/99.
- Number of tries before match found? 99, on average.
- Each of the 99 tries calls getSock twice, so 198 for the first pair, on average.

sorter1, Continued

- So, how many calls to getSock to find the first pair given n pairs in the basket? $2(2n-1) = 4n-2$.
- Now, there are $n-1$ pairs left. Finding the second pair will take $4(n-1)-2 = 4n-6$ calls.
- When there is one pair left, it takes 2 calls.
- Total
  $= 2 + 6 + 10 + ... + 4n-2$
  $= 4(1+2+...+n)-2n$
  $= 4\ n(n+1)/2 - 2n$
  $= 2n^2$.
- So, $O(n^2)$ algorithm.

How about sorter2?

- sorter2: Grab one sock, then grab socks one at a time until its mate is found.
- Number of getSocks before first pair is found?
- Probability of a match is 1/99 if 50 pairs.
- Number of tries before match found? 99, on average.
- Each of the 99 tries calls getSock once, plus the one in the beginning to get things started, so 100 for the first pair, on average.

sorter2, Continued

- Calls to getSock to find the first pair if n pairs in the basket? $1+2n-1 = 2n$.
- Now, there are n-1 pairs left. Finding the second pair will take $2n$-2 calls.
- When there is one pair left, it takes 2 calls.
- Total $= 2 + 4 + 6 + \ldots + 2n$ $= 2(1+2+\ldots+n)$ $= 2\, n(n+1)/2$ $= n^2 + 1$.
- So, $O(n^2)$ algorithm (smaller constant, though).

How about sorter3?

- sorter3: Grab one sock, then grab socks one at a time (without replacement) until its mate is found.
- Number of calls to getSocks before first pair is found?
- 99 socks in basket, the one we want is at the halfway mark, or 50, on average.
- On average 51 socks taken from basket before first pair is found.

sorter3, Continued

- Calls to getSock to find the first pair if n pairs in the basket? $n + 1$.
- Now, there are n-1 pairs left. Finding the second pair will take $(n-1)+1 = n$ calls.
- When there is one pair left, it takes 2 calls.
- Total
  $= 2 + 3 + 4 + \ldots + (n+1)$
  $= (1+2+\ldots+n)+(n+1) - 1$
  $= n(n+1)/2 + n$
  $= n^2/2 + 3n/2$.
- So, another $O(n^2)$ algorithm (even smaller constant, though).

How about sorter4?

- sorter4: Keep a pile on the bed. Grab a sock and check if its mate is on the bed. If not, add it to the pile.
- Since all socks are matched up and no socks are returned to the basket, there is precisely one getSock call per sock, $2n$ if n pairs.
- So, a $O(n)$ algorithm!

Next Time

- Tuesday:
  - Guest lecturer: Barbara Ryder on Compilers.
  - HW2 due.
- Thursday:
  - Guest lecturer: Gabriel Nieves on image processing.