Chapter 5: Algorithms and Heuristics

CS105: Great Insights in Computer Science
Fact Cheat Sheet!

- Fact 1: Average number of trials until an event happens if that event has probability $p$ is $1/p$
  
  - Example: If the probability of winning the lottery is $1/1000$, then it will take, on average 1000 plays to win

- Fact 2: The sum of all of the numbers from 1 to $n$ is simply $n(n+1)/2$, which is equal to $(n^2+n)/2$
  
  - The sum of the numbers from 1 to 100 is $100*101/2 = 5050$
Fact 5: A logarithm of base 10, $\log$, is approximately the number of times you can divide the number by 10. A logarithm of base two is approximately the number of times you can divide by 2.

- Example: $\log_{10}(1000) = 3$ and $\log_{2}(1024) = 10$

Fact 4: The number of syllables for any number, $n$, is approximately $\log(n)$ if the number is read one digit at a time.

- Example: The number of syllables in 1000 (read as one-zero-zero-zero) is $\log(1000) = 3$
Fact 5: Average number of peeks into a list to find a number, if the list is in random order, is $n/2$

Example: If I have a list of 20 numbers in random order and I would like to find if 5 is on the list. The average number of times I need to look at elements of the list is $20/2 = 10$. 
Song Growth

- Let’s talk about how many 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.
In a cavern, in a canyon,  
Excavating for a mine,  
Dwelt a miner forty niner,  
And his daughter Clementine.  
Oh my darling, oh my darling,  
Oh my darling, Clementine!  
Thou art lost and gone forever  
Dreadful sorry, Clementine.

Light she was and like a fairy,  
And her shoes were number nine,  
Herring boxes, without topses,  
Sandals were for Clementine.  
Oh my darling, oh my darling,  
Oh my darling, Clementine!  
Thou art lost and gone forever  
Dreadful sorry, Clementine.

Drove she ducklings to the water  
Ev'ry morning just at nine,  
Hit her foot against a splinter,  
Fell into the foaming brine.  
Oh my darling, oh my darling,  
Oh my darling, Clementine!  
Thou art lost and gone forever  
Dreadful sorry, Clementine.

Ruby lips above the water,  
Blowing bubbles, soft and fine,  
But, alas, I was no swimmer,  
So I lost my Clementine.  
Oh my darling, oh my darling,  
Oh my darling, Clementine!  
Thou art lost and gone forever  
Dreadful sorry, Clementine.

How I missed her! How I missed her,  
How I missed my Clementine,  
But I kissed her little sister,  
I forgot my Clementine.  
Oh my darling, oh my darling,  
Oh my darling, Clementine!  
Thou art lost and gone forever  
Dreadful sorry, Clementine.
### Counting Syllables

<table>
<thead>
<tr>
<th>verses</th>
<th>syllables</th>
<th>verses</th>
<th>syllables</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>55</td>
<td>5</td>
<td>271</td>
</tr>
<tr>
<td>2</td>
<td>109</td>
<td>6</td>
<td>323</td>
</tr>
<tr>
<td>3</td>
<td>161</td>
<td>7</td>
<td>375</td>
</tr>
<tr>
<td>4</td>
<td>219</td>
<td>8</td>
<td>432</td>
</tr>
</tbody>
</table>
Plotting Syllables

• Total syllables roughly, $T(n) = 54 \times n$. 
Old MacDonald had a farm, E-I-E-I-O
And on his farm he had a cow, E-I-E-I-O
With a "moo-moo" here and a "moo-moo" there
Here a "moo" there a "moo"
Everywhere a "moo-moo"
Old Macdonald had a farm, E-I-E-I-O

Old Macdonald had a farm, E-I-E-I-O
And on his farm he had a pig, E-I-E-I-O
With a (snort) here and a (snort) there
Here a (snort) there a (snort)
Everywhere a (snort-snort)
With a "moo-moo" here and a "moo-moo" there
Here a "moo" there a "moo"
Everywhere a "moo-moo"
Old Macdonald had a farm, E-I-E-I-O

Old Macdonald had a farm, E-I-E-I-O
And on his farm he had a horse, E-I-E-I-O
With a "neigh, neigh" here and a "neigh, neigh" there
Here a "neigh" there a "neigh"
Everywhere a "neigh-neigh"
With a (snort) here and a (snort) there
Here a (snort) there a (snort)
Everywhere a (snort-snort)
With a "moo-moo" here and a "moo-moo" there
Here a "moo" there a "moo"
Everywhere a "moo-moo"
Old Macdonald had a farm, E-I-E-I-O

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

<table>
<thead>
<tr>
<th>verses</th>
<th>syllables</th>
<th>verses</th>
<th>syllables</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>59</td>
<td>5</td>
<td>515</td>
</tr>
<tr>
<td>2</td>
<td>140</td>
<td>6</td>
<td>684</td>
</tr>
<tr>
<td>3</td>
<td>243</td>
<td>7</td>
<td>875</td>
</tr>
<tr>
<td>4</td>
<td>368</td>
<td>8</td>
<td>1088</td>
</tr>
</tbody>
</table>
Plotting Syllables

verses: \( n \)
N Bottles of Beer

Verse 1
One Bottles of Beer on the wall,
One bottles of beer,
If one of those bottles should happen to fall,
Two bottles of beer on the wall,

Verse 2
Two Bottles of Beer on the wall,
Two bottles of beer,
If one of those bottles should happen to fall,
Three bottles of beer on the wall,

Verse 3
Three Bottles of Beer on the wall,
Three bottles of beer,
If one of those bottles should happen to fall,
Four bottles of beer on the wall,

Verse 4
Four Bottles of Beer on the wall,
Four bottles of beer,
If one of those bottles should happen to fall,
Five bottles of beer on the wall,

Verse 2643
Two Six Four Three Bottles of Beer on the wall,
Two Six Four Three bottles of beer,
If one of those bottles should happen to fall,
Two Six Four Three bottles of beer on the wall,

Verse 10741
One Zero Seven Four One Bottles of Beer on the wall,
One Zero Seven Four One bottles of beer,
If one of those bottles should happen to fall
One Zero Seven Four Two bottles of beer on the wall
## Counting Syllables

<table>
<thead>
<tr>
<th>verses</th>
<th>syllables</th>
<th>verses</th>
<th>syllables</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>30</td>
<td>10</td>
<td>328</td>
</tr>
<tr>
<td>2</td>
<td>60</td>
<td>20</td>
<td>628</td>
</tr>
<tr>
<td>3</td>
<td>90</td>
<td>100</td>
<td>3560</td>
</tr>
<tr>
<td>4</td>
<td>120</td>
<td>1000</td>
<td>38400</td>
</tr>
</tbody>
</table>
• 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”.

\[
\begin{array}{c}
\text{0.4 } n^2 \text{ (Quadratic)} \\
\text{2.86 } n \lg n \\
\text{0.572 } n^2 \lg n \\
2 \ n \ (\text{Linear})
\end{array}
\]
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.

$$17n + 91.2 \log(n) + n^{1/2}$$
• Formally, big O is a notation that denotes a class of functions all of which are upper bounded asymptotically.

\[ O(n) \]

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

\[ 17n \]
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

• Quadratic: $O(n^2)$
  - An Old Lady
  - Old Macdonald
  - There Was a Tree
  - $O(n \log n)$
  - $N$ Bottles of Beer
  - $N$ Little Monkeys
  - $O(n^2 \log n)$
  - $N$ Days of Christmas
  - Who Knows $N$?
Common Growth Classes

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

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

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

- $O(n^2 \log n)$
  - $N$ Days of Christmas
  - Who Knows $N$?
# Common Growth Classes

<table>
<thead>
<tr>
<th>Linear: $O(n)$</th>
<th>Quadratic: $O(n^2)$</th>
<th>$O(n \log n)$</th>
<th>$O(n^2 \log n)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Dreidel</td>
<td>- An Old Lady</td>
<td>- N Bottles of Beer</td>
<td>- N Days of Christmas</td>
</tr>
<tr>
<td>- Clementine</td>
<td>- Old Macdonald</td>
<td>- N Little Monkeys</td>
<td>- Who Knows $N$?</td>
</tr>
<tr>
<td>constant size verse</td>
<td>each verse contains the next higher number</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Common Growth Classes

- Linear: $O(n)$
  - Dreidel
  - Clementine
- Quadratic: $O(n^2)$
  - An Old Lady
  - Old Macdonald
  - There Was a Tree
- $O(n \log n)$
  - $N$ Bottles of Beer
  - $N$ Little Monkeys
- $O(n^2 \log n)$
  - $N$ Days of Christmas
  - Who Knows $N$?

Each verse contains the next higher number.

Constant size verse.

Each verse a constant size larger than the previous.
Common Growth Classes

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

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

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

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

linear ($O(n)$)

versus

syllables

quadratic ($O(n^2)$)

O($n \log n$)
As far as I know, classical songs are all linear (\(O(n)\)), quadratic (\(O(n^2)\)), \(O(n \log n)\), and \(O(n^2 \log n)\).

Nevertheless, I can make up a few more songs to demonstrate a few other important growth rates.
• Kids like 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 $n$?
  - $5 + 2 \log n$: This song is $O(\log n)$. 
Very Annoying 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?
The country desperately needs to upgrade its roads and seaports, and to exponentially increase agricultural and manufactured exports.

Expontially 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.

The demands on an organization 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.

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.
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.

The demands on an organization 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.
• Demand for IVF treatments, which climbed **exponentially** during the past 20 years, has plateaued.
  
  • \( \text{demand}(t) = 10^t \)

• Consequently, an unintended but **exponentially** growing number of middle-class Americans is being affected.
  
  • \( \text{affectedpeople}(t) = 10^t \)

• 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.
  
  • \( \text{affectedArea}(t) = 10^t \)

• Injury rates [for cheerleaders] are "**exponentially** higher for a flier than for a footballer," says NCCSI's Robert Cantu.
Algorithm Analysis

• Now, that we have a sense of how various quantities grow as a function of other quantities.

• Let’s apply this idea to analyzing our sock sorters.

• For each algorithm, how does the number of reaches into the laundry basket grow as a function of the number of pairs of socks $n$?

• Let’s remember the cheat sheet!
Analyzing Sock Sorting

- How many socks does sockA take out of the basket to sort 50 pairs of socks?

- sockA: choose a random pair. Return to basket if no match.

- # of socks removed 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 removes two socks, so 198 sock pulls for the first pair, on average.
sockA, Continued

- So, how many socks removed 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 \( 4(n-1)-2 = 4n-6 \) sock removals.

- When there is one pair left, it takes 2 sock removals.

- Total
  \[
  = 2 + 6 + 10 + ... + 4n-2
  = 4(1+2+...+n)-2n
  = 4 \frac{n(n+1)}{2} - 2n
  = 2n^2.
  \]

- So, \( O(n^2) \) algorithm.
Intuitive Analysis

- Since the time to find each pair is proportional to the number of pairs left, the total amount of time until all pairs are found is roughly $n^2$.

- sockC is the same, except the time is halved. Still order $n^2$. 

quadratic ($O(n^2)$)
How about sockB?

- **sockB**: Keep a pile on the table. Grab a sock and check if its mate is already out. If not, add it to the pile.

- Since all socks are matched up and no socks are returned to the basket, each sock is removed from the basket precisely once, $2n$ if $n$ pairs.

- So, a $O(n)$ algorithm! Linear, order $n$, etc.

- No wonder it’s fast.
Algorithm Design Goal

- Not just trying to solve a problem, but solve it well with respect to some goal.

- Best way to the airport?
  - Time?
  - Money?
  - Gas?
  - What else?