"Big O" Notation

<table>
<thead>
<tr>
<th>Fancy Name</th>
<th>Grows Like</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exponential</td>
<td>$O(2^n)$ or $O(10^n)$</td>
</tr>
<tr>
<td>Quadratic</td>
<td>$O(n^2 \log n)$</td>
</tr>
<tr>
<td>Linear</td>
<td>$O(n^2)$</td>
</tr>
<tr>
<td>Logarithmic</td>
<td>$O(n \log n)$</td>
</tr>
<tr>
<td>Constant</td>
<td>$O(\log n)$</td>
</tr>
<tr>
<td>Constant</td>
<td>$O(1)$</td>
</tr>
</tbody>
</table>
Another Decision Problem

- Problem: “Is there a pair that sums to $x$”? Given a list of numbers, is there a pair of distinct numbers in the list that sums up to $x$?

- Is there a pair that sums to 86?

- What is the order of the runtime of your approach?
Another Decision Problem

Problem: “Is there a pair that sums to $x$”? Given a list of numbers, is there a pair of distinct numbers in the list that sums up to $x$?

• Is there a pair that sums to 86?

48, 41, 14, 46, 31, 2, 27, 12, 22, 71, 44, 63, 33, 64, 83, 28, 96, 87, 52

• What is the order of the runtime of your approach?
Yet Another Problem

• Problem: “Is there a subset that sums to $x$”? Given a list of numbers, is there a subset of distinct numbers in the list that sums up to $x$?

• Is there a subset that sums to 433?

• Does this problem seem harder? Is it solvable?
Yet Another Problem

• Problem: “Is there a subset that sums to $x$”? Given a list of numbers, is there a subset of distinct numbers in the list that sums up to $x$?

• Is there a subset that sums to 433?

48, 41, 14, 46, 31, 2, 27, 12, 22, 71, 44, 63, 33, 64, 83, 28, 96, 87, 52

• Does this problem seem harder? Is it solvable?
NP-complete Problems

• There’s another remarkable class of problems:
  - They *can* be solved (not incomputable).
  - The best algorithms we have take *exponential* time.
  - They are *ubiquitous*.
  - No one has proven they *can’t* be solved efficiently.
  - If we could solve *one* of them efficiently, we could solve *all* of them efficiently. (Reduction again!)
  - Essence: A “yes” answer has a short “proof”. 
NP and Puzzles

- Most newspaper puzzles can be thought of as NP problems. Why? Because the answer appears the next day. It might be hard to find the answer, but it’s easy to check once you hear it.

- Sometimes I say NP stands for “Nice Puzzle” for this reason.
Next, we’re going to look at several critical ideas in algorithm design and use the example of Google to motivate them.

Our question: How does Google find stuff?
The Internet (a piece)

- athos.rutgers.edu
- www.cs.rutgers.edu
- www.dcis.rutgers.edu
- porthos.rutgers.edu
- dir.yahoo.com
- paul.rutgers.edu
- google.com
- patmedia.net
The Internet (a piece)

athos.rutgers.edu
www.cs.rutgers.edu
www.dcis.rutgers.edu

porthos.rutgers.edu

(dir.yahoo.com

(patmedia.net

paul.rutgers.edu
google.com

(domain) name
computer
network
Hi, I’m porthos.rutgers.edu.
A Conversation

porthos.rutgers.edu

To: dir.yahoo.com
Hi, I’m porthos.rutgers.edu.

To: porthos.rutgers.edu
What gives?

dir.yahoo.com

Web Server
porthos.rutgers.edu

To: dir.yahoo.com
    Hi, I’m porthos.rutgers.edu.

To: porthos.rutgers.edu
    What gives?

To: dir.yahoo.com
    I’m told you have a web page called "Science/Computer_Science/
College_and_University_Departments/?b=20". Can you send me a copy?
porthos.rutgers.edu

To: dir.yahoo.com
    Hi, I’m porthos.rutgers.edu.

To: porthos.rutgers.edu
    What gives?

To: dir.yahoo.com
    I’m told you have a web page called
    "Science/Computer_Science/
    College_and_University_Departments/?b=20". Can
    you send me a copy?

To: porthos.rutgers.edu
    Sure:
A Conversation

To: dir.yahoo.com
Hi, I’m porthos.rutgers.edu.

To: porthos.rutgers.edu
What gives?

To: dir.yahoo.com
I’m told you have a web page called
"Science/Computer_Science/
College_and_University_Departments/?b=20". Can you send me a copy?

To: porthos.rutgers.edu
Sure:
http://dir.yahoo.com/Science/Computer_Science/College_and_University_Departments/?b=140

- **MUST University**
  Online accredited associate, bachelors, masters, and doctorate level as well as diploma programs in computer science.

- **New Mexico State University**
  www.cs.nmsu.edu

- **Technische Universität Braunschweig**
• Most web browsing consists of requests for specific pages.
• But, what happens if we don’t know what we want?
• A “search engine” is just a web server that can respond to a particular request for web pages.
Another Conversation

porthos.rutgers.edu

Web Server

google.com
Hi, I'm porthos.rutgers.edu.
Another Conversation

porthos.rutgers.edu

To: google.com
Hi, I’m porthos.rutgers.edu.

To: porthos.rutgers.edu
Whazzup?

google.com

Web Server
porthos.rutgers.edu

To: google.com
Hi, I’m porthos.rutgers.edu.

To: porthos.rutgers.edu
Whazzup?

To: google.com
Can I have a copy of the page “search?hl=en&q=rutgers+computer+science&btnG=Google+Search”?
Another Conversation

To: google.com
   Hi, I’m porthos.rutgers.edu.

To: porthos.rutgers.edu
   Whazzup?

To: google.com
   Can I have a copy of the page "search?
   hl=en&q=rutgers+computer+science&btnG=Google
   +Search"?

To: porthos.rutgers.edu
   Sure:
To: google.com
   Hi, I’m porthos.rutgers.edu.

To: porthos.rutgers.edu
   Whazzup?

To: google.com
   Can I have a copy of the page "search?hl=en&q=rutgers+computer+science&btnG=Google+Search"?

To: porthos.rutgers.edu
   Sure:
Indirection

• Porthos asked Google where it could find pages about “rutgers computer science”.

• Google responded with a page that included addresses of other pages.

• Porthos can now request those pages directly from the web servers that “host” (store and dispense) them.

• Note that Google does not search Rutgers in response to this query.
How Does Google Know?

• So, somehow, Google has to put together a web page in response to *any* query, which includes a list of names of pages that contain those terms.

• But, how does it know which pages contain which terms?

• Theories?
  1.
  2.
  3.
An Experiment

• 9/28/09: Created a page http://www.cs.rutgers.edu/~cmansley/fall09/cs105/googletest1.html with the word “talimoneri”. This page is linked from the course syllabus.

• 9/28/09: Created a page http://www.cs.rutgers.edu/~cmansley/fall09/cs105/googletest2.html with the word “teexpaey”. No other page on the web links to it.

• 10/14/08: Still no hits for either word.

• 10/19/08: talimoneri is found!
What Do You Think Now?

- Google learned the word “talimoneri” around a month after I put up a web page with the word and linked it to the course web page.

- Google still didn’t know the word “teexpaey” more than a month later in spite of being on a similar (but unlinked) page at the same time.

- We need to understand how pages link to each other.
A Piece of the Web

1. www.cs.rutgers.edu/~mlittman/courses/cs442-06/: 2, 3, 1, 2, 4, 5
2. www.cs.rutgers.edu/~mlittman/: 1, 6, 7, 10
3. paul.rutgers.edu/~babes/: 1
4. www.cs.rutgers.edu/~mlittman/courses/cs442-06/python/
5. www.cs.rutgers.edu/~mlittman/courses/cs442-06/googletest1.html: 7, 1
6. www.cs.rutgers.edu/rl3/: 8, 10
7. www.cs.rutgers.edu/~mlittman/topics/googlewhacks
9. www.cs.rutgers.edu/~mlittman/courses/cs442-06/googletest2.html: 1
Graphs

- In CS and discrete math, this kind of structure is known as a graph.
  - Nodes: Web pages, in this case.
  - Links: Pointer from one web page to another, in this case.
Some Graph Terms

- **source**: a node with no incoming links.
- **sink**: a node with no outgoing links.
- **path**: a list of nodes such that each adjacent pair of nodes has a link from the first to the second.
- **cycle**: a path in which the first and last node are the same.
- **connected component**: a set of nodes such that there is a path from any node to any other node in the set.
- **tour**: a cycle including all nodes in the graph.
Graphs Are Everywhere

- What are the nodes, links, paths, source, sinks, connected components of each?

- Two more definitions: A graph is **undirected** if each connected pair of nodes is connected in both directions.

- An undirected graph is a **tree** if it has no cycles.

- Is each example directed or undirected? Tree or not? Can it have cycles?
IGN Comics' March Madness Tournament
Circuit Diagram

DC motor direction controller

Q1 and Q2 - BC148 / 2N2222
Q3 and Q5 - SL100
Q4 and Q6 - SK100
Diodes are 1N4001

CAUTION: The max motor current rating not to exceed max. SL/SK100 rating (1A with heat sink)
Sexual Contact Network
Molecular Diagram
Algorithms on Graphs

We can represent a graph in the computer by a list of nodes, and a function that, given a node \( i \), returns the list of nodes to which \( i \) is linked.
Reachable

• A node $j$ is reachable from a node $i$ if there is a path that begins at $i$ and ends at $j$.

• Let’s list all the nodes reachable from $i$.

• Any node that is reachable from a node that $i$ is linked to is also reachable.
Don’t Revisit!

- What goes wrong? Once we realize we can reach some node, we should mark it as “reached” and never pursue it again.
To-Do List Helps

• Keep track of what you need to do later once your current activity is done.
• Stack. (See ‘paintcan’ video.)
• So, how does Google do it?

I. Web crawl: download known pages, collect links to other pages, repeat.

II. Indexing: Build a giant index that associates each word with a list of pages on which it appears.

III. Distributed search: Use lots and lots and lots of computers to do fast lookups.
Another name for the lecture is “Google II”.

Sorting is a great topic in CS:
- relatively simple
- extremely important
- illustrates lots of different algorithms and analysis techniques

There’s more than one way to skin a cat.
Three Pages of Words
What Can We Do?

• All the information is there, and we can sift through it.

• But, it’s slow and error-prone to skim through every page every time we want to find something.

• If there are $N$ words (total) on the web pages, how long would it take to sift through them each time? (Use “big O” notation.)

• How can we organize the data to simplify?
| 4 a | 9 course | 3 frames | 9 http | 3 no | 4 python |
| 9 a | 4 courses | 3 frameset | 9 i | 3 noframes | 3 right |
| 4 add | 9 courses | 9 from | 9 in | 3 noresize | 3 rows |
| 4 address | 4 cs | 9 google | 4 index | 9 index | 3rutgers |
| 9 and | 9 cs | 9 googleblackout | 9 include | 9 index | 4rutgers |
| 9 any | 9 did | 4 googletest | 9 insights | 9 not | 9 rutgers |
| 4 apache | 4 differences | 9 greater | 9 old | 4 science | 9 science |
| 4 at | 4 directory | 4 h | 9 on | 9 title | 9rutgers |
| 4 b | 9 discovery | 9 h | 9 other | 9 title | 4 title |
| 9 b | 4 doctype | 9 has | 9 own | 9 to | 9 title |
| 3 babes | 9 documents | 3 head | 9 own | 4 ul | 9 unknown |
| 3 banner | 9 does | 4 head | 9 own | 9 unknown | 9 unlinked |
| 3 body | 3 doesn | 3 header | 9 p | 3 uses | 9 unlinked |
| 4 body | 4 dtd | 9 header | 3 page | 9 verified | 9 unlinked |
| 4 browser | 4 edu | 9 history | 9 page | 9 visit | 3 uses |
| 3 but | 9 edu | 9 homepage | 9 pages | 4 w | 9 word |
| 4 c | 4 edu | 9 home | 4 parent | 4 walkthru | 9 www |
| 3 cols | 4 en | 9 how | 4 part | 9 was | 9 www |
| 9 computer | 4 exception | 4 href | 4 pdf | 9 which | 3 your |
| 9 concatenating | 4 final | 9 href | 4 port | 9 which | 9 your |
| 9 concocted | 4 find | 3 htm | 9 primary | 9 word | 9 your |
| 9 consists | 9 for | 3 html | 9 public | 9 www | 3 your |
| 3 contents | 3 frame | 4 html | 9 purpose | 3 them | 3 your |
| 9 frame | 9 html | 4 nim | 9 purpose | 3 them | 3 your |
• Phonebook, look for a last name vs. look for a first name.

• “Is there a pair that sums to 86?” Don’t have to consider all pairs.

• Is there a repeated number in the list?

• Not to mention min, max, median.
Selection Sort

- Idea is quite simple. We go through the list one item at a time.
- We keep track of the smallest item we’ve found.
- When we’re through the list, we pull the smallest item out and add it to a list of sorted items.
- We repeat until all the items have been removed.
About a 2 1/2 min. to sort 100 items.
Selection Sort Analysis

• How many comparisons does Selection Sort do in the worst case? Assume the list is length $N$. Hint: What song is it like? You can use “big O” notation.

• Does it matter whether the list is sorted or not?
Other Sorting Approaches

- How else can you imagine sorting?
- Fewer comparisons than $O(N^2)$?
  - bubblesort
  - counting sort
  - insertion sort
  - Shell sort
Guess Who?

- Each player picks a character.
- Players take turns asking each other yes/no questions.
- First player to uniquely identify the other player’s character wins!
Mindreader: Set Cards
Mindreader: Set Cards

A
B
C
D
E
F
G
H
I
J
K
L
M
N
O
P
<table>
<thead>
<tr>
<th>Cross-Hatched?</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="https://via.placeholder.com/150" alt="Images" /></td>
</tr>
</tbody>
</table>

| ![Images](https://via.placeholder.com/150) |
| ![Images](https://via.placeholder.com/150) |
| ![Images](https://via.placeholder.com/150) |
| ![Images](https://via.placeholder.com/150) |

| ![Images](https://via.placeholder.com/150) |
| ![Images](https://via.placeholder.com/150) |
| ![Images](https://via.placeholder.com/150) |
| ![Images](https://via.placeholder.com/150) |

| ![Images](https://via.placeholder.com/150) |
| ![Images](https://via.placeholder.com/150) |
| ![Images](https://via.placeholder.com/150) |
| ![Images](https://via.placeholder.com/150) |

| ![Images](https://via.placeholder.com/150) |
| ![Images](https://via.placeholder.com/150) |
| ![Images](https://via.placeholder.com/150) |
| ![Images](https://via.placeholder.com/150) |

| ![Images](https://via.placeholder.com/150) |
| ![Images](https://via.placeholder.com/150) |
| ![Images](https://via.placeholder.com/150) |
| ![Images](https://via.placeholder.com/150) |
Cross-Hatched?
Cross-Hatched?
Squiggle?
Squiggle?
Squiggle?
Insight

- Each question splits the remaining set of possibilities into two subsets (yes and no).
Insight

- Each question splits the remaining set of possibilities into two subsets (yes and no).

- We want to pick a question so that the larger of the two subsets is as small as possible.
Insight

• Each question splits the remaining set of possibilities into two subsets (yes and no).

• We want to pick a question so that the larger of the two subsets is as small as possible.

• Half!
Insight

- Each question splits the remaining set of possibilities into two subsets (yes and no).
- We want to pick a question so that the larger of the two subsets is as small as possible.
- How many questions?
- Half!
Insight

• Each question splits the remaining set of possibilities into two subsets (yes and no).

• We want to pick a question so that the larger of the two subsets is as small as possible.

• Half!

• How many questions?
  • \( n=1 \), questions = 0
Insight

• Each question splits the remaining set of possibilities into two subsets (yes and no).

• We want to pick a question so that the larger of the two subsets is as small as possible.

• Half!

• How many questions?
  • $n=1$, questions = 0
  • $n=2$, questions = 1
Insight

- Each question splits the remaining set of possibilities into two subsets (yes and no).
- We want to pick a question so that the larger of the two subsets is as small as possible.
- Half!

- How many questions?
  - $n=1$, questions = 0
  - $n=2$, questions = 1
  - $n=4$, questions = 2
Insight

• Each question splits the remaining set of possibilities into two subsets (yes and no).

• We want to pick a question so that the larger of the two subsets is as small as possible.

• Half!

• How many questions?
  • $n=1$, questions = 0
  • $n=2$, questions = 1
  • $n=4$, questions = 2
  • $n=8$, questions = 3
Insight

• Each question splits the remaining set of possibilities into two subsets (yes and no).

• We want to pick a question so that the larger of the two subsets is as small as possible.

• Half!

• How many questions?
  • $n=1$, questions = 0
  • $n=2$, questions = 1
  • $n=4$, questions = 2
  • $n=8$, questions = 3
  • $n=16$, questions = 4
Insight

• Each question splits the remaining set of possibilities into two subsets (yes and no).

• We want to pick a question so that the larger of the two subsets is as small as possible.

• Half!

• How many questions?

  • $n=1$, questions $= 0$
  • $n=2$, questions $= 1$
  • $n=4$, questions $= 2$
  • $n=8$, questions $= 3$
  • $n=16$, questions $= 4$
  • $n$, questions $= \log n$. 