COT 6405: Analysis of Algorithms
Giri NARASIMHAN
Text

3rd Edition

- ISBN-10: 9788120340077
Course Preliminaries

Course Webpage:
- Lecture Slides; Reading Material; Announcements; HWs
- VISIT OFTEN!

Class meets MW 2:00 – 3:15 PM, CASE/ECS 235
Office ECS 254B; Office Hours: By Appointment Only
Phone: x-3748; Email: giri@cis.fiu.edu
Final: Monday, 12/11/2019, 12:00 – 2:00 PM, CASE 235
Momentos

- Slides and Audio online
- Need to register
  - Go to https://fiu.momentos.life
  - If you don’t already have an account
    - Click on “Sign up”
    - Follow instructions & use referral code: 5T6LSV
  - If you have an account, “Add Course” with code 5T6LSV
  - Verify account using link sent to email
Why I am here?

I am here because …

- It’s required
  - What do I expect to learn in this class?
  - Who should know about **Algorithms**?
  - Is there a future in this field?
  - Would I ever need it if I want to be a software engineer or work with databases?

Hate being here because …

- It’s required
Questions you should ask …

- What do I expect to learn in this class?
- Who should know about Algorithms?
- Is there a future in this field?
- Would I ever need it if I want to be a software engineer or work with databases?
Person of the Year ...
Time’s Person of the Year

2018

2017
The first hundred votes ...  

<table>
<thead>
<tr>
<th></th>
<th>48</th>
<th>12</th>
<th>9</th>
<th>12</th>
<th>23</th>
<th>12</th>
<th>22</th>
<th>12</th>
<th>12</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>48</td>
<td>93</td>
<td>93</td>
<td>93</td>
<td>12</td>
<td>12</td>
<td>93</td>
<td>12</td>
<td>93</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>12</td>
<td>93</td>
<td>48</td>
<td>48</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>33</td>
<td>79</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>12</td>
<td>12</td>
<td>93</td>
<td>12</td>
<td>12</td>
<td>9</td>
<td>12</td>
<td>23</td>
<td>12</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>12</td>
<td>12</td>
<td>12</td>
<td>33</td>
<td>93</td>
<td>93</td>
<td>93</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>12</td>
<td>9</td>
<td>12</td>
<td>23</td>
<td>93</td>
<td>48</td>
<td>48</td>
<td>12</td>
<td>12</td>
<td>44</td>
<td>55</td>
</tr>
<tr>
<td>93</td>
<td>93</td>
<td>93</td>
<td>12</td>
<td>12</td>
<td>9</td>
<td>12</td>
<td>23</td>
<td>12</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>12</td>
<td>12</td>
<td>48</td>
<td>12</td>
<td>48</td>
<td>48</td>
<td>12</td>
<td>48</td>
<td>88</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>12</td>
<td>12</td>
<td>93</td>
<td>12</td>
<td>12</td>
<td>9</td>
<td>12</td>
<td>23</td>
<td>12</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>12</td>
<td>12</td>
<td>12</td>
<td>33</td>
<td>93</td>
<td>93</td>
<td>93</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>12</td>
</tr>
</tbody>
</table>

Every number in the table corresponds to a vote for a person with that ID.

**Majority**: More than 50% of the votes

Who won a majority?
Standard Approaches

- Keep a list of candidates and their counts
  - Every vote needs to be compared against every candidate in the worst case
- Sort the list and count
  - Sorting is the bottleneck
  - Can we avoid sorting?
Wacky Ideas, anyone?

- What if I pick two random votes and they turn out to be different?
  - Discard and reduce the problem size
- What if I pick two random votes and they are the same?
  - Well, this needs work and you will need to think about it!
<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>48</td>
<td>12</td>
<td>9</td>
<td>12</td>
<td>23</td>
<td>12</td>
<td>22</td>
</tr>
<tr>
<td>48</td>
<td>93</td>
<td>93</td>
<td>93</td>
<td>12</td>
<td>12</td>
<td>93</td>
</tr>
<tr>
<td>12</td>
<td>93</td>
<td>48</td>
<td>48</td>
<td>12</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>12</td>
<td>12</td>
<td>93</td>
<td>12</td>
<td>12</td>
<td>9</td>
<td>12</td>
</tr>
<tr>
<td>12</td>
<td>12</td>
<td>93</td>
<td>12</td>
<td>9</td>
<td>12</td>
<td>23</td>
</tr>
<tr>
<td>12</td>
<td>9</td>
<td>12</td>
<td>23</td>
<td>93</td>
<td>48</td>
<td>48</td>
</tr>
<tr>
<td>93</td>
<td>93</td>
<td>93</td>
<td>12</td>
<td>12</td>
<td>9</td>
<td>12</td>
</tr>
<tr>
<td>12</td>
<td>12</td>
<td>48</td>
<td>12</td>
<td>48</td>
<td>48</td>
<td>12</td>
</tr>
<tr>
<td>12</td>
<td>12</td>
<td>93</td>
<td>12</td>
<td>12</td>
<td>9</td>
<td>12</td>
</tr>
<tr>
<td>12</td>
<td>12</td>
<td>12</td>
<td>33</td>
<td>93</td>
<td>93</td>
<td>93</td>
</tr>
<tr>
<td>12</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>12</td>
</tr>
</tbody>
</table>
Difference between Intro and Analysis of Algorithms?

- More on how to analyze ... more on complexity
- More on correctness ...
- More algorithms ...
- More data structures ... bigger bag of tricks
- More on different algorithmic models ...
  - Randomized, online, amortized, adaptive, approximation, heuristic, quantum,
Evaluation

- Exams (2) 45%
- Quizzes 10%
- HW Assignments 35%
- Class Participation 10%
What you should already know ...

- Array Lists
- Linked Lists
- Sorted Lists
- Stacks and Queues
- Basic Sorting Algorithms
- Trees
- Binary Search Trees
- Heaps and Priority Queues
- Graphs
  - Adjacency Lists
  - Adjacency Matrices
The Algorithmic Process

- Formulate the question
- Write down a basic idea, an approach
- Write down pseudocode
- Prove correctness
- Analyze pseudocode and think about upper and lower bounds
- Iterate
History of Algorithms

- Euclid, 300 BC
- Bhaskara, 6th c
- Al Khwarizmi, 9th c
- Fibonacci, 13th c
- Gauss, 18-19th c
- Babbage, 19th c
- Turing, 20th c
- von Neumann, 20th c
- Knuth, Karp, Tarjan, Rabin, …, 20-21st c
Reading for next class

- Big-Oh notation
  - Chapter 2, 3
- All sorting algorithms and their analysis
  - Chapters 2, 6, 7, 8