CS21: INTRODUCTION TO COMPUTER SCIENCE

Prof. Mathieson
Fall 2017
Swarthmore College
Welcome to CS21!  

TODO:

• **Registered**: sit at a computer and sign the attendance sheet by the end of class

• **Waitlist**: find a seat not at a computer and sign the waitlist sheet from Lauri

• **Everyone**: pick up a handout
Outline Sept 4:

• Introductions
• Algorithm example / what is CS?
• Areas of computer science
• Log in to lab machines
• Syllabus highlights

• Notes: this course has NO prerequisites. If you have extensive CS experience let me know.
Why should I take CS21?
Why should I take CS21?

Myth 1

Computer Science is about computers
Why should I take CS21?

Myth 1

Computer Science is about computers

“When human beings acquired language, we learned not just how to listen but how to speak. When we gained literacy, we learned not just how to read but how to write. And as we move into an increasingly digital reality, we must learn not just how to use programs but how to make them.”

-Douglas Rushkoff
Introductions
Course Staff

• Instructors
  • **Section 1, Lab A&C: Sara Mathieson**
  • Section 2: Jeff Knerr
  • Section 3, Lab D: Rich Wicentowski
  • Lab B: David Mauskop

• Academic Support Coordinator
  • **Lauri Courtenay**

• Ninjas for Section 1
  • **Clarissa Phillips**
  • **Anya Chaudhri**

• Grader for Section 1
  • **Ellen Liu**

Lauri will introduce the ninja program
Algorithm example
Example of an algorithm: nonograms

Image credit: www.nonograms.org
Handout example

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>2</th>
<th></th>
<th>5</th>
<th>3</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>5</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Image credit: www.nonograms.org
With a partner...

- Mini-bio (name, year, where from, summer, major...) remember this for Wed - you’ll introduce your partner then!

- Start the handout questions (more important to get to question 4 than finish the puzzle)

- For question 4, you can discuss ideas instead of writing down everything if you’re short on time
Example Algorithm for row checking

• Start at the beginning of the row
Example Algorithm for row checking

- Start at the beginning of the row
- Move to the right as long as the boxes are E’s

Modified from notes by Jeff Knerr
Example Algorithm for row checking

• Start at the beginning of the row
• Move to the right as long as the boxes are E’s
• When we find an F, start counting how many consecutive F’s there are

Modified from notes by Jeff Knerr
Example Algorithm for row checking

• Start at the beginning of the row
• Move to the right as long as the boxes are E’s
• When we find an F, start counting how many consecutive F’s there are
• Once we get to another E, store or save how many consecutive F’s we saw

Modified from notes by Jeff Knerr
Example Algorithm for row checking

• Start at the beginning of the row
• Move to the right as long as the boxes are E’s
• When we find an F, start counting how many consecutive F’s there are
• Once we get to another E, store or save how many consecutive F’s we saw
• Keep repeated the above 3 steps building up a list of counts of consecutive F’s

Modified from notes by Jeff Knerr
Example Algorithm for row checking

- Start at the beginning of the row
- Move to the right as long as the boxes are E’s
- When we find an F, start counting how many consecutive F’s there are
- Once we get to another E, store or save how many consecutive F’s we saw
- Keep repeated the above 3 steps building up a list of counts of consecutive F’s
- When we reach the end of the row and have a list of counts, compare it with the row numbers

Modified from notes by Jeff Knerr
Example Algorithm for row checking

- Start at the beginning of the row
- Move to the right as long as the boxes are E’s
- When we find an F, start counting how many consecutive F’s there are
- Once we get to another E, store or save how many consecutive F’s we saw
- Keep repeated the above 3 steps building up a list of counts of consecutive F’s
- When we reach the end of the row and have a list of counts, compare it with the row numbers
- If the lists of numbers are the same, the solution is valid
- If not, the solution is invalid

Modified from notes by Jeff Knerr
In these algorithms, can you find…

• Input/output
In these algorithms, can you find…

- **Input/output**
  - Input: target row numbers, candidate solution
  - Output: yes/no
In these algorithms, can you find…

• **Input/output**
  • **Input:** target row numbers, candidate solution
  • **Output:** yes/no

• **Branching**
In these algorithms, can you find...

- **Input/output**
  - Input: target row numbers, candidate solution
  - Output: yes/no

- **Branching**
  - Operation changes based on \( E \) vs. \( F \)
In these algorithms, can you find…

- **Input/output**
  - Input: target row numbers, candidate solution
  - Output: yes/no
- **Branching**
  - Operation changes based on \( E \) vs. \( F \)
- **Looping**
In these algorithms, can you find…

- **Input/output**
  - Input: target row numbers, candidate solution
  - Output: yes/no

- **Branching**
  - Operation changes based on E vs. F

- **Looping**
  - Repeated the steps to count each block of F’s
In these algorithms, can you find…

- **Input/output**
  - Input: target row numbers, candidate solution
  - Output: yes/no

- **Branching**
  - Operation changes based on E vs. F

- **Looping**
  - Repeated the steps to count each block of F’s

- **Data structures**
In these algorithms, can you find…

- **Input/output**
  - Input: target row numbers, candidate solution
  - Output: yes/no

- **Branching**
  - Operation changes based on $E$ vs. $F$

- **Looping**
  - Repeated the steps to count each block of $F$’s

- **Data structures**
  - Way to store the input as well as the counts
Log in to lab machines
Steps for lab machines today

- Find your account sheet with username and password
- Make sure to sign and return (at the end of class)
- Change your password!

```
cilantro[~]$ passwd
Enter login(LDAP) password: 
New CompSci password:  
Retype new CompSci password:  
LDAP password information changed for smathieson
passwd: password updated successfully

cilantro[~]$
```

- Bookmark course webpage (click on the star in Firefox)

Syllabus highlights
Syllabus highlights and notes

- Notes will be posted *after* class on the webpage
Syllabus highlights and notes

• Notes will be posted after class on the webpage

• Textbook – free and online 😊
Syllabus highlights and notes

- Notes will be posted *after* class on the webpage
- Textbook – free and online 😊
- Make sure to attend lab! (mandatory, attendance taken)
Syllabus highlights and notes

• Notes will be posted *after* class on the webpage

• Textbook – free and online 😊

• Make sure to attend lab! (mandatory, attendance taken)

• Labs are (almost always) due Saturdays before midnight
Notes will be posted after class on the webpage.

Textbook – free and online 😊

Make sure to attend lab! (mandatory, attendance taken)

Labs are (almost always) due Saturdays before midnight

Extensions
  • Known conflicts: must be arranged now
  • Emergencies: must talk to your class dean
Syllabus highlights and notes

- Notes will be posted after class on the webpage
- Textbook – free and online 😊
- Make sure to attend lab! (mandatory, attendance taken)
- Labs are (almost always) due Saturdays before midnight

**Extensions**
- Known conflicts: must be arranged now
- Emergencies: must talk to your class dean

**Email** – allow 24 hours for a response
Syllabus highlights and notes

• Notes will be posted *after* class on the webpage

• Textbook – free and online 😊

• Make sure to attend lab! (mandatory, attendance taken)

• Labs are (almost always) due Saturdays before midnight

• **Extensions**
  • Known conflicts: must be arranged **now**
  • Emergencies: must talk to your class dean

• **Email** – allow 24 hours for a response

• **Piazza** should be used for all content/logistics questions
Notes will be posted after class on the webpage.

Textbook – free and online 😊

Make sure to attend lab! (mandatory, attendance taken)

Labs are (almost always) due Saturdays before midnight

**Extensions**

- Known conflicts: must be arranged now
- Emergencies: must talk to your class dean

**Email** – allow 24 hours for a response

**Piazza** should be used for all content/logistics questions

Office hours: **3-5pm on Fridays in 260** (often moved to lab)
<table>
<thead>
<tr>
<th>Class</th>
<th>Dean</th>
<th>Schedule Contact Information</th>
</tr>
</thead>
</table>
| First-Year | Dean Karen Henry         | Betsy Durning
610-690-5744
edurnin1@swarthmore.edu |
| Sophomore | Dean Jason Rivera        | Stephanie Holznagel (assists with schedule only)
610-690-3999
sholzna1@swarthmore.edu |
| Junior   | Dean Dion Lewis          | Bonnie Lytle
610-328-8456
dlytle1@swarthmore.edu |
| Senior   | Dean Nathan Miller       | Stephanie Holznagel
610-690-3999
sholzna1@swarthmore.edu |
Registering with the Student Disability Service

Please contact Leslie Hempling, Director of Student Disability Services, at lhempli1@swarthmore.edu or 610-690-5014 to arrange an intake appointment. We are happy to hold initial appointments for incoming students by phone. If at all possible, please submit documentation of your disability in advance so that we can review it prior to talking with you. We recommend that you contact us as early as possible since some accommodations (e.g., electronic books, interpreters, etc.) can take several weeks to arrange. We want to be sure that your needs are met in time for classes.

Visit the Accommodations Process and the Documentation Guidelines sections in the "For Students" section of this website for all details.
End of class

- Make sure to sign and return user account form!
- WiCS introduction and signup