introduction to CS50 until 2pm

5-minute break *

introduction to pset0 until 2:30pm

* if shopping 2pm class, ask to leave early; watch rest at cs50.harvard.edu

experimental screencast at screencast.cs50.net
Week 0
73% of CS50 students have never taken CS before
what ultimately matters in this course is not so much where you end up relative to your classmates but where you, in **Week 11**, end up relative to yourself in **Week 0**
problem solving
inputs → outputs
binary
0, 1
decimal
0, 1, 2, 3, 4, 5, 6, 7, 8, 9
\[ 100 \times 1 + 10 \times 2 = 120 \]
100 \times 1 + 10 \times 2 + 1 \times 3
ASCII

A B C D E F G H I ...
65 66 67 68 69 70 71 72 73 ...
H

72  73  33
HI!
72  73  33
abstraction
inputs → → outputs
algorithms
0 Stand up and think of the number 1.

1 Pair off with someone standing.
   Add your numbers together.

2 One of you should then sit down.
   If you're still standing, go back to step 1.
0 pick up phone book
1 open to middle of phone book
2 look at names
3 if Smith is among names
   call Mike
5 else if Smith is earlier in book
   open to middle of left half of book
   go back to step 2
8 else if Smith is later in book
   open to middle of right half of book
   go back to step 2
11 else
12 quit
pick up phone book
open to middle of phone book
look at names
if Smith is among names
call Mike
else if Smith is earlier in book
open to middle of left half of book
go back to step 2
else if Smith is later in book
open to middle of right half of book
go back to step 2
else
quit
0   pick up phone book
1   open to middle of phone book
2   look at names
3   if Smith is among names
4       call Mike
5   else if Smith is earlier in book
6       open to middle of left half of book
7       go back to step 2
8   else if Smith is later in book
9       open to middle of right half of book
10      go back to step 2
11   else
12      quit
0  pick up phone book
1  open to middle of phone book
2  look at names
3  if Smith is among names
   4     call Mike
5  else if Smith is earlier in book
   6     open to middle of left half of book
   7     go back to step 2
6  else if Smith is later in book
   7     open to middle of right half of book
   8     go back to step 2
9  else
10    quit
0  pick up phone book
1  open to middle of phone book
2  look at names
3  if Smith is among names
   4    call Mike
5  else if Smith is earlier in book
6      open to middle of left half of book
7      go back to step 2
8  else if Smith is later in book
9      open to middle of right half of book
10     go back to step 2
11  else
12     quit
time to solve

size of problem

$n$
The graph shows the relationship between the size of the problem and the time to solve it. The red line represents a time complexity of $n$, while the yellow line represents a time complexity of $n/2$. This indicates that as the size of the problem increases, the time to solve it increases at a slower rate for the $n/2$ case compared to the $n$ case.
size of problem

n

n/2

n/2

log n

size of problem

time to solve
This is CS50
cs50.ly/new
lectures

attend first

...

attend last
<table>
<thead>
<tr>
<th>Lecture</th>
<th>Filmed</th>
<th>Released</th>
</tr>
</thead>
<tbody>
<tr>
<td>Week 0</td>
<td>Scratch</td>
<td>Wed 8/31, 1pm – 2:30pm</td>
</tr>
<tr>
<td>Week 1</td>
<td>C</td>
<td>Fri 8/26, 11am – 2pm</td>
</tr>
<tr>
<td>Week 2</td>
<td>Arrays</td>
<td>Tue 9/6, 12:30pm – 2:30pm</td>
</tr>
<tr>
<td>Week 3</td>
<td>Algorithms</td>
<td>Mon 9/12, 12:30pm – 3:30pm</td>
</tr>
<tr>
<td>Week 4</td>
<td>Memory</td>
<td>Tue 9/13, 12:30pm – 3:30pm</td>
</tr>
<tr>
<td>Week 5</td>
<td>Data Structures</td>
<td>Mon 9/26, 12pm – 3pm</td>
</tr>
<tr>
<td>Week 6</td>
<td>...</td>
<td>Mon 10/3, 11:30am – 2:30pm</td>
</tr>
<tr>
<td></td>
<td>Machine Learning</td>
<td>Thu 10/13, 4pm – 5:15pm</td>
</tr>
<tr>
<td>Week 7</td>
<td>Python</td>
<td>Wed 10/5, 12pm – 3pm</td>
</tr>
<tr>
<td>Week 8</td>
<td>HTTP</td>
<td>Tue 10/18, 12:30pm – 3:30pm</td>
</tr>
<tr>
<td>Week 9</td>
<td>SQL</td>
<td>Mon 10/24, 12pm – 3pm</td>
</tr>
<tr>
<td>Week 10</td>
<td>JavaScript</td>
<td>Mon 10/31, 11:30am – 2:30pm</td>
</tr>
<tr>
<td>Week 11</td>
<td>The End</td>
<td>Mon 11/21, 1pm – 2:30pm</td>
</tr>
</tbody>
</table>
walkthroughs

most Wednesdays at 1pm

embedded in every problem set
problem sets

released on Fridays

due (10 days later) on Mondays at noon
<table>
<thead>
<tr>
<th>Problem Set</th>
<th>Language</th>
<th>Released</th>
<th>Due</th>
</tr>
</thead>
<tbody>
<tr>
<td>Problem Set 0</td>
<td>Scratch</td>
<td>Wed 8/31</td>
<td>Mon 9/5, noon</td>
</tr>
<tr>
<td>Problem Set 1</td>
<td>C</td>
<td>Fri 9/2</td>
<td>Mon 9/12, noon</td>
</tr>
<tr>
<td>Problem Set 2</td>
<td>C</td>
<td>Fri 9/9</td>
<td>Mon 9/19, noon</td>
</tr>
<tr>
<td>Problem Set 3</td>
<td>C</td>
<td>Fri 9/16</td>
<td>Mon 9/26, noon</td>
</tr>
<tr>
<td>Problem Set 4</td>
<td>C</td>
<td>Fri 9/23</td>
<td>Mon 10/3, noon</td>
</tr>
<tr>
<td>Problem Set 5</td>
<td>C</td>
<td>Fri 9/30</td>
<td>Mon 10/10, noon</td>
</tr>
<tr>
<td>Problem Set 6</td>
<td>Python</td>
<td>Fri 10/21</td>
<td>Mon 10/31, noon</td>
</tr>
<tr>
<td>Problem Set 7</td>
<td>Python, SQL</td>
<td>Fri 10/28</td>
<td>Mon 11/7, noon</td>
</tr>
<tr>
<td>Problem Set 8</td>
<td>JavaScript</td>
<td>Fri 11/4</td>
<td>Mon 11/14, noon</td>
</tr>
</tbody>
</table>
take as first year?

take with other courses?
sections

less comfortable

more comfortable

somewhere in between
<table>
<thead>
<tr>
<th>Section</th>
<th>Dates</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>Wed 9/7*</td>
</tr>
<tr>
<td>Arrays</td>
<td>Mon 9/12, Tue 9/13, Wed 9/14</td>
</tr>
<tr>
<td>Algorithms</td>
<td>Mon 9/19, Tue 9/20, Wed 9/21</td>
</tr>
<tr>
<td>File I/O</td>
<td>Mon 9/26, Tue 9/27, Wed 9/28</td>
</tr>
<tr>
<td>Data Structures</td>
<td>Mon 10/3, Tue 10/4, Wed 10/5</td>
</tr>
<tr>
<td>Review for Test</td>
<td>Mon 10/10, Tue 10/11, Wed 10/12</td>
</tr>
<tr>
<td>TF’s Choice</td>
<td>Mon 10/17, Tue 10/18, Wed 10/19</td>
</tr>
<tr>
<td>Python</td>
<td>Mon 10/24, Tue 10/25, Wed 10/26</td>
</tr>
<tr>
<td>SQL</td>
<td>Mon 10/31, Tue 11/1, Wed 11/2</td>
</tr>
<tr>
<td>JavaScript</td>
<td>Mon 11/7, Tue 11/8, Wed 11/9</td>
</tr>
<tr>
<td>Review for Quiz</td>
<td>Mon 11/14*</td>
</tr>
</tbody>
</table>

* Course-wide and filmed.
sections

Mondays

Tuesdays

Wednesdays
office hours

Wednesdays
Thursdays

Sundays
office hours

Mondays
Tuesdays
Wednesdays
Thursdays
Fridays
Saturdays
Sundays
tutoring
CS50 Puzzle Day

Hab
Sat 9/3, 12pm - 3:30pm
cs50.harvard.edu/register
CS50 Coding Contest
introduction to CS50 until 2pm

* 5-minute break *

introduction to pset0 until 2:30pm

* if shopping 2pm class, aok to leave early; watch rest at cs50.harvard.edu
introduction to CS50 until 2pm

5-minute break *

introduction to pset0 until 2:30pm

* if shopping 2pm class, aok to leave early; watch rest at cs50.harvard.edu
```c
#include <stdio.h>

int main(void)
{
    printf("hello, world\n");
}
```
when clicked

say hello, world
say hello, world
forever

say hello, world
repeat 50

say "hello, world"
$x < y$
if $x < y$ then
  say $x$ is less than $y$
else
  if $x > y$ then
    say $x$ is greater than $y$
  else
    say $x$ is equal to $y$
define cough n
repeat n
  say cough for 1 secs
  wait 1 secs