

This is CS50

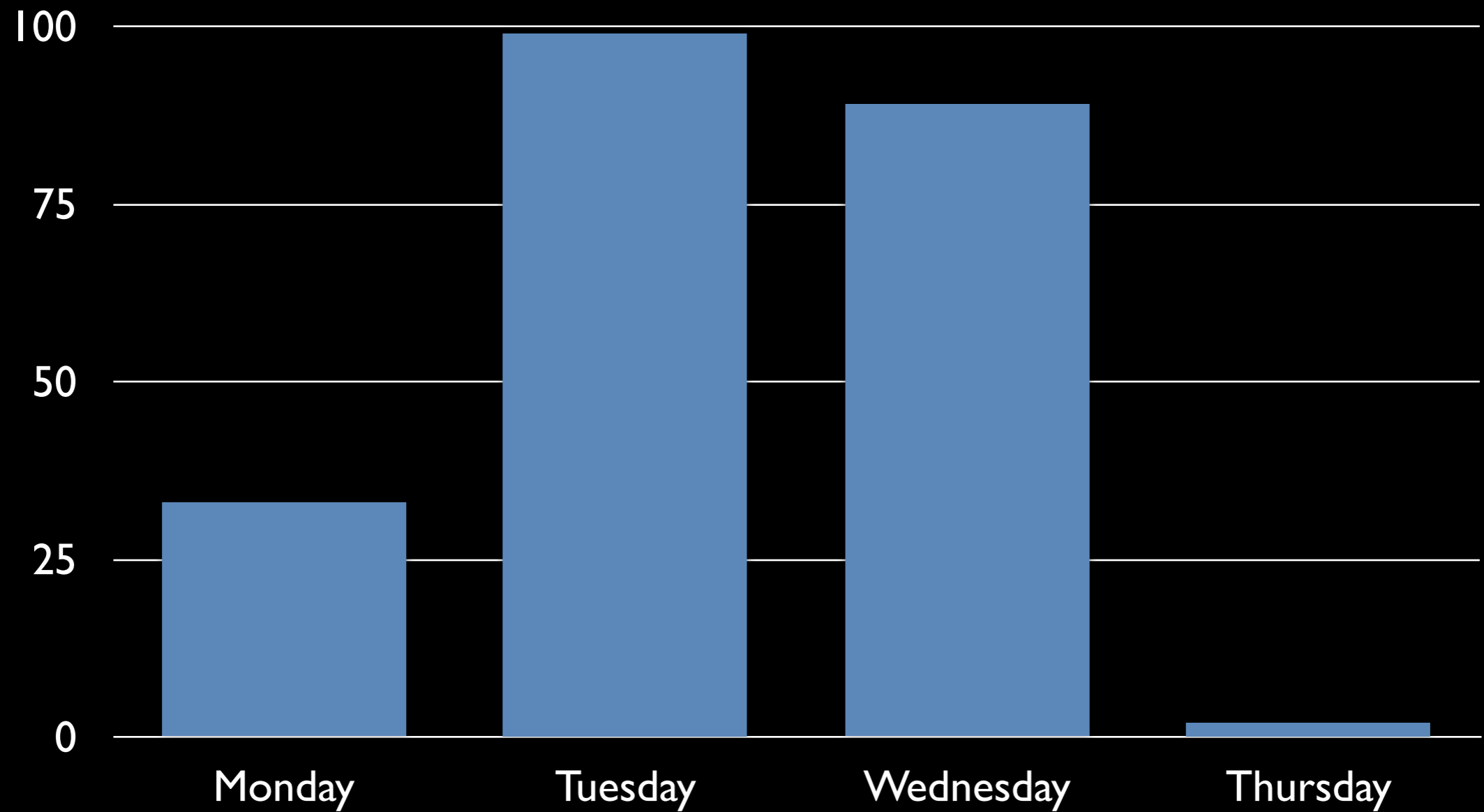
Pass/Fail

CS50 Lunches

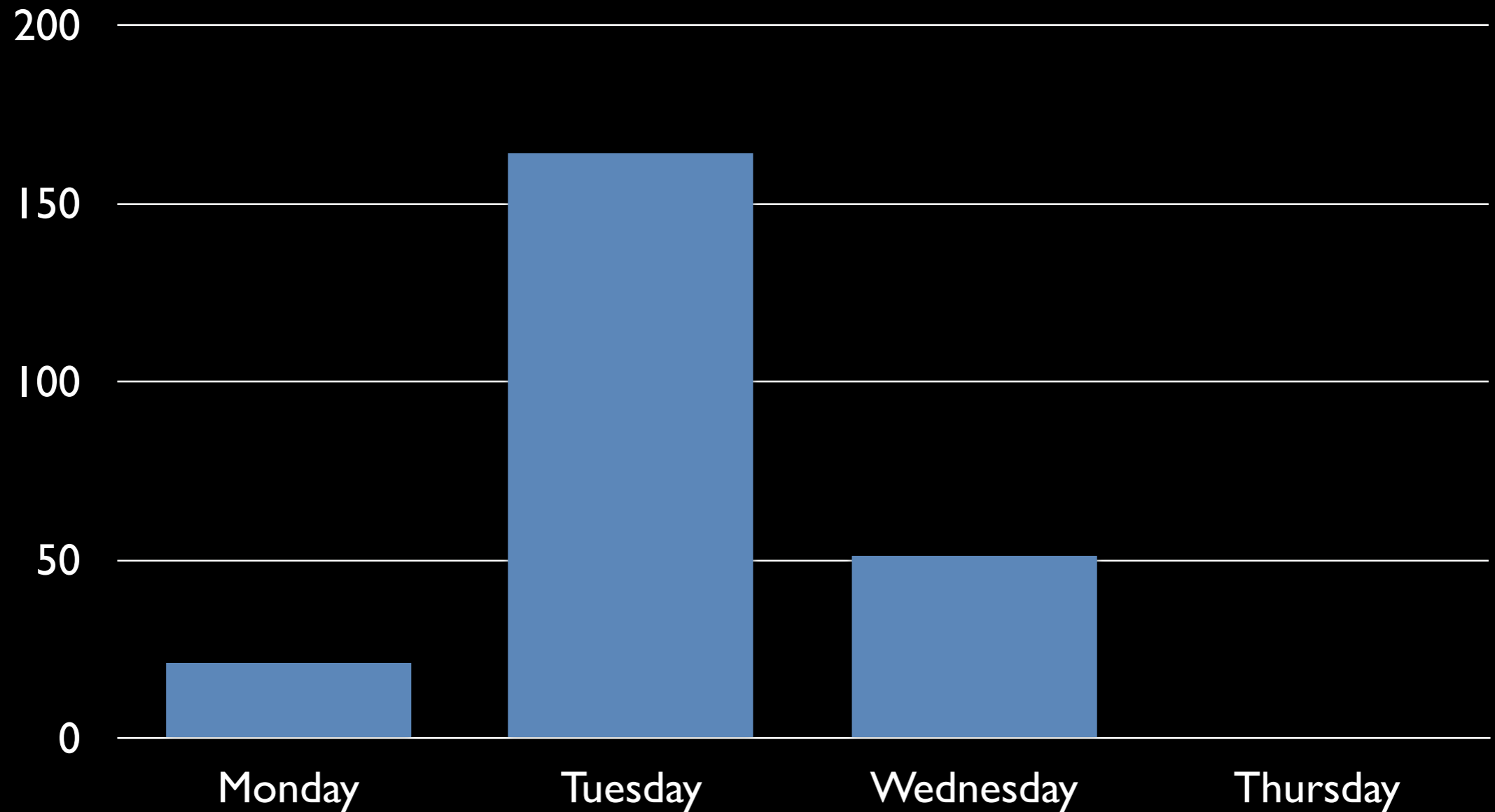
Fridays, 1:15pm

CS50 Queue

Questions



Average Wait Time (sec)



sectioning

starts later today, ends mon at noon

supersections

this sun, mon, tue

problem set I

standard edition

problem set 1

Hacker Edition

walkthrough I

this Sun, 7pm

how to compile a program

make hello

how to run a program

```
./hello
```

functions

main

Standard Library

stdio.h

printf

...

CS50 Library

cs50.h

GetChar

GetDouble

GetFloat

GetInt

GetLongLong

GetString

primitive types

char double float int long long ...

CS50 types

`bool` `string` ...

printf

`%c` `%d` `%f` `%lld` `%s` ...

escape sequences

`\n` `\r` `\t` `\'` `\"` `\\` `\0` ...

math

+ - * / %

precedence

Operator	Description	Associativity
() [] . ->	Parentheses (grouping) Brackets (array subscript) Member selection via object name Member selection via pointer	left-to-right
++ --	Postfix increment/decrement (see Note 1)	
++ -- + - ! ~ (type) * & sizeof	Prefix increment/decrement Unary plus/minus Logical negation/bitwise complement Cast (change type) Dereference Address Determine size in bytes	right-to-left
* / %	Multiplication/division/modulus	left-to-right
+ -	Addition/subtraction	left-to-right
<< >>	Bitwise shift left, Bitwise shift right	left-to-right
< <= > >=	Relational less than/less than or equal to Relational greater than/greater than or equal to	left-to-right
== !=	Relational is equal to/is not equal to	left-to-right
&	Bitwise AND	left-to-right
^	Bitwise exclusive OR	left-to-right
	Bitwise inclusive OR	left-to-right
&&	Logical AND	left-to-right
	Logical OR	left-to-right
?:	Ternary conditional	right-to-left
= += -= *= /= %= &= ^= = <<= >>=	Assignment Addition/subtraction assignment Multiplication/division assignment Modulus/bitwise AND assignment Bitwise exclusive/inclusive OR assignment Bitwise shift left/right assignment	right-to-left
,	Comma (separate expressions)	left-to-right

,	Comma (separate expressions)	left-to-right
<<= >>=	Bitwise shift left/right assignment	
^= =	Bitwise exclusive/inclusive OR assignment	
%= &=	Modulus/bitwise AND assignment	
+= -=	Addition/subtraction assignment	

conditions

```
if (condition)
{
    // do this
}
```

conditions

```
if (condition)
{
    // do this
}
else (condition)
{
    // do that
}
```


conditions

```
if (condition)
{
    // do this
}
else if (condition)
{
    // do that
}
else
{
    // do this other thing
}
```

Boolean expressions

```
if (condition || condition)
{
    // do this
}
```

Boolean expressions

```
if (condition && condition)
{
    // do this
}
```

switches

```
switch (expression)
{
    case i:
        // do this
        break;

    case j:
        // do that
        break;

    default:
        // do this other thing
}
```

loops

```
for (initializations; condition; updates)
{
    // do this again and again
}
```

loops

```
while (condition)
{
    // do this again and again
}
```

loops

```
do  
{  
    // do this again and again  
}  
while (condition);
```

to be continued...