Genome Sciences 373
Genome Informatics

Quiz Section #1

March 31, 2015

About me, course logistics, etc.

Matthew's contact info

Email: mwsnyder@uw.edu

Phone: 206-685-3720

Office hours: Mondays 2:00-3:00pm

Foege building, room S110

…or by appointment

About me, course logistics, etc.

Homework policy:

No late homework accepted without PRIOR arrangements

Grading is equally about your **effort** and your **execution**

First homework assigned tomorrow

About me, course logistics, etc.

What is the quiz section all about?

not a “how-to” homework session

mostly we will learn Python and review in-class material

attendance is not required, but the material covered in section is required

Questions about
course logistics?

What is an algorithm?

Formally: an exact *procedure*, or set of instructions, to achieve a predictable final result from a given input

Colloquially: a thorough method for solving a problem according to step-by-step instructions

Some key features of algorithms

Typically written in “pseudocode” or similar

Inputs and outputs specified at the outset

Often designed to achieve some goal in the “best” way possible

- fastest
- least memory
- most accurate

Example of an algorithm: smallest number

Find the smallest of three numbers

Algorithm FindSmallestNumber

Input: three numbers A, B, and C

Output: the largest number

Example of an algorithm: smallest number

Find the smallest of three numbers

Algorithm FindSmallestNumber

Input: three numbers A, B, and C

Output: the largest number

```
current_smallest ← A
if B < current_smallest:
        current_smallest ← B
else:
        [do nothing]
if C < current_smallest:
        current_smallest ← C     [… else: do nothing]
return current_smallest
```

Example of an algorithm: smallest number

Find the smallest of three numbers

Algorithm FindSmallestNumber

Input: three numbers A, B, and C

Output: the largest number

```
current_smallest ← A
if B < current_smallest:
        current_smallest ← B
else:
        [do nothing]
if C < current_smallest:
        current_smallest ← C     [… else: do nothing]
return current_smallest
```

Another example: Euclid's algorithm

Find the greatest common divisor of two numbers

If A > B, and A & B have greatest common divisor G, then G is also the GCD of A and (A – B)

Example: A = 63, B = 18

- **What is the GCD?**
- **Can we generalize this process as a set of rules or steps to follow to ALWAYS find the GCD?**

Another example: Euclid's algorithm

Find the greatest common divisor of two numbers

Algorithm EuclidGCD

Input: two numbers: A and B

Output: the GCD of A and B

Another example: Euclid's algorithm

Find the greatest common divisor of two numbers

Algorithm EuclidGCD

Input: two numbers: A and B

Output: the GCD of A and B

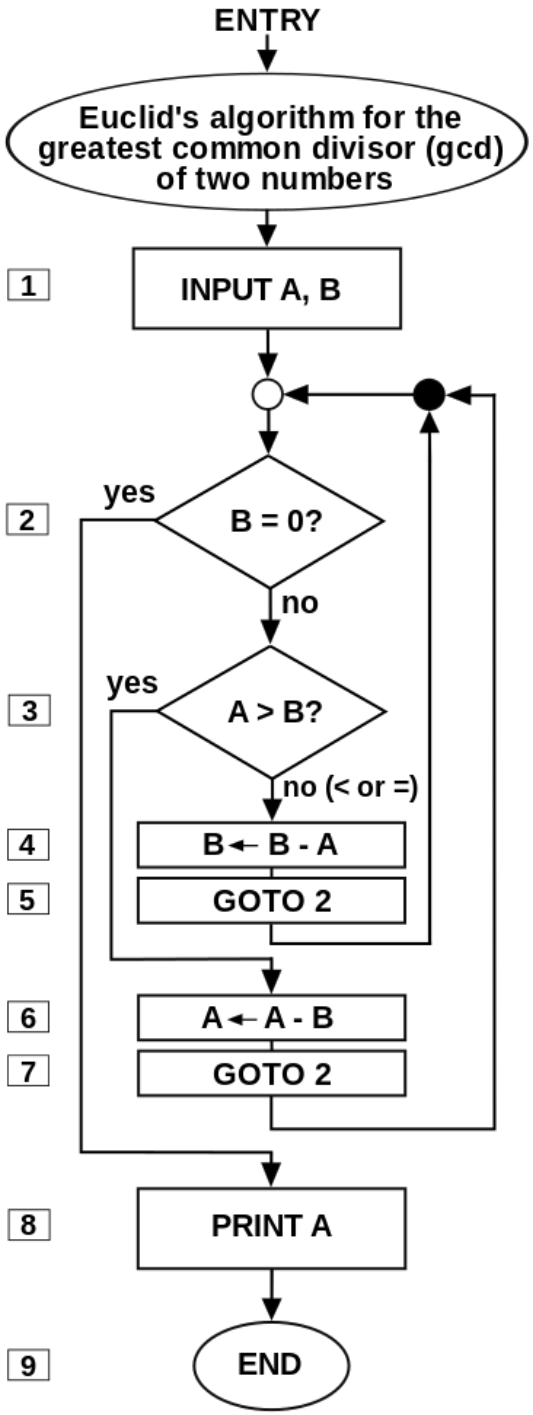
start:

if B = 0 then output A (*else: keep going)*

if A > B then A ← A – B

else B ← B – A

go to *start*



Often we can draw an algorithm as a flowchart

What's the problem with this flowchart?

How could we improve it?

Source: http://en.wikipedia.org/wiki/Algorithm#/media/File:Euclid_flowchart.svg

Common pitfalls and issues to consider

What if I enter a zero?

What if I enter a negative number?

What if I enter a fraction?

Is my algorithm guaranteed to ever finish?

In class example: algorithm for factorial

Recall: for any positive integer *k*,

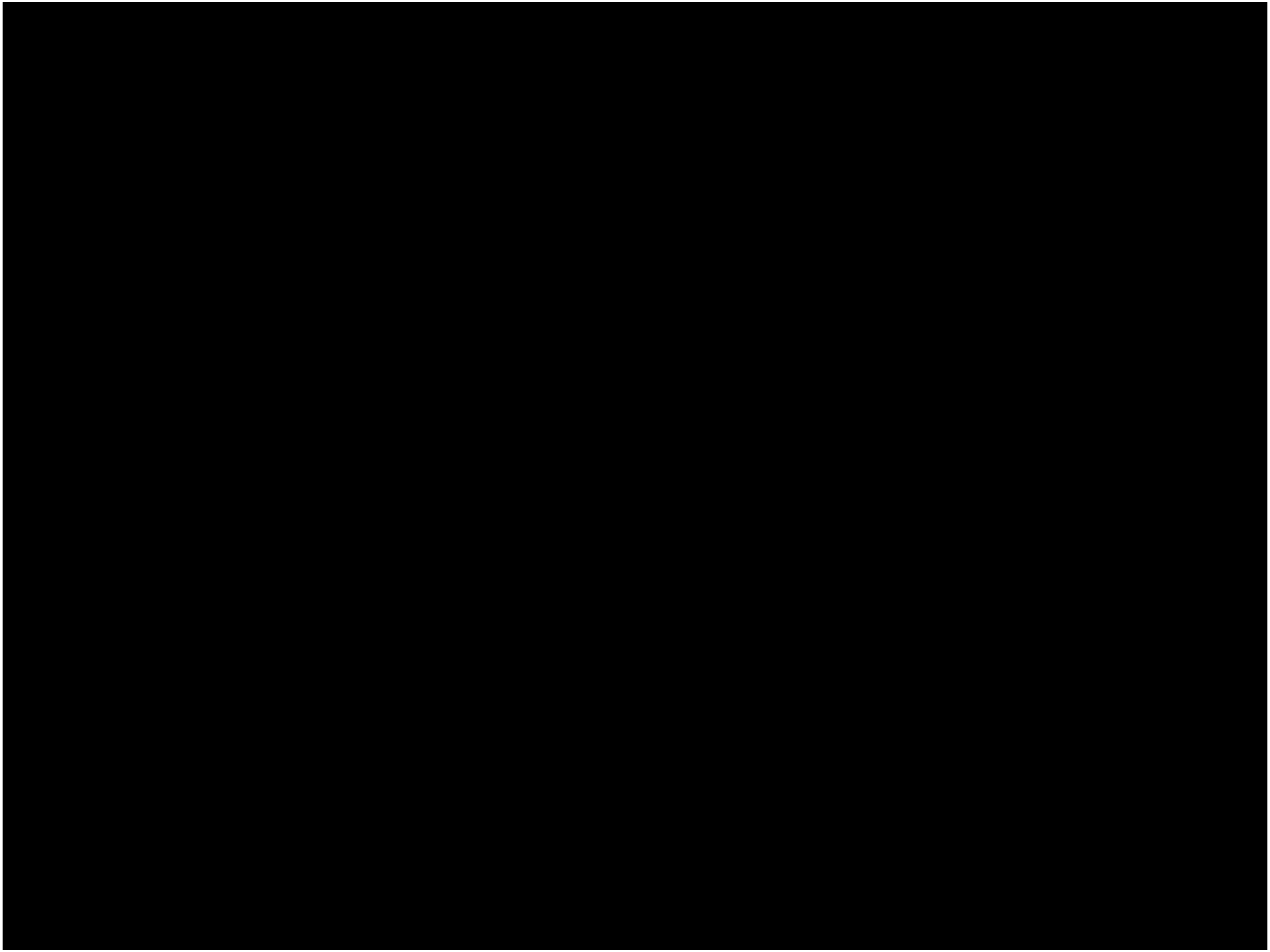
$$k! = k * (k - 1) * (k-2) * \ldots * 1$$

What is an algorithm for calculating the factorial?

Algorithms are not the same as computer code!

But, algorithms can be *implemented* in programming languages

You have already done hard work!



Why do we program?

Get Stuff Done.

- Automate repeated tasks
- Extract information from huge amounts of data
- Manipulate or convert data to get it in the right format

What tools do we need to write a program?

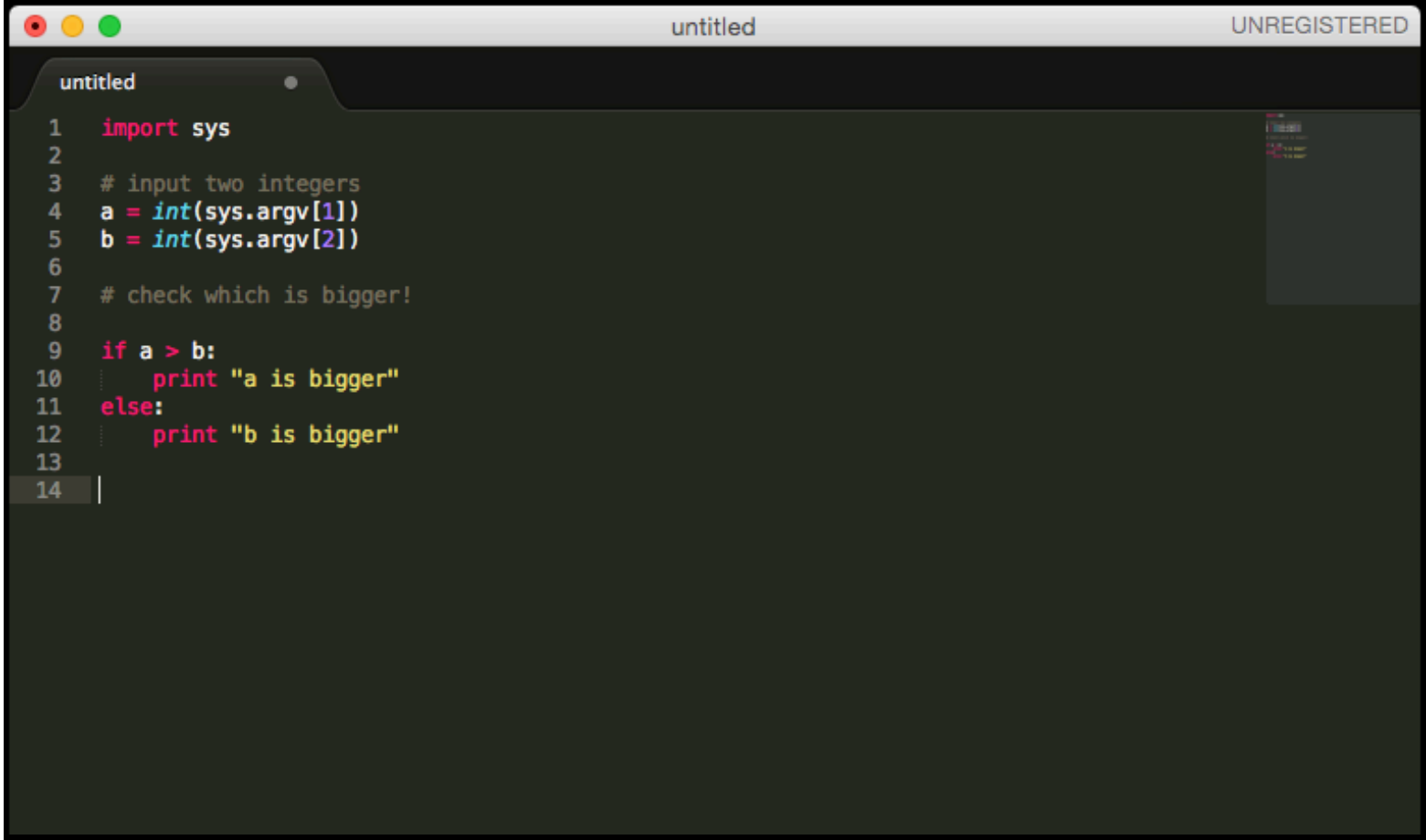
Technical stuff	*Important stuff*
Variables	Patience
Flow control	Practice
Syntax	Comments
Grammar	Internet

What tools do we need to write a program?

Technical stuff	*Important stuff*	*Practical stuff*
Variables	Patience	**Editor**
Flow control	Practice	**Interpreter**
Syntax	Comments	
Grammar	Internet	

Today: focus on editor & interpreter

Python Editors: too many choices!



SublimeText: http://www.sublimetext.com/

PyCharm: https://www.jetbrains.com/pycharm/download/

The Python Interpreter

```
0330 18:32 adityas:~%
0330 18:32 adityas:~%
0330 18:32 adityas:~% python -V
Python 2.7.6
0330 18:32 adityas:~% python --version
Python 2.7.6
0330 18:32 adityas:~%
```

```
0330 18:32 adityas:~%
0330 18:33 adityas:~%
0330 18:33 adityas:~% python
Python 2.7.6 (default, Sep  9 2014, 15:04:36)
[GCC 4.2.1 Compatible Apple LLVM 6.0 (clang-600.0.39)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> import sys
>>> my_name = "Matthew"
>>> my_zip = 98107
>>> print my_name, my_zip
Matthew 98107
>>>
>>> print my_phone
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'my_phone' is not defined
>>>
```

Typing your script line-by-line: not a good plan

The Python Interpreter

```
0330 18:37 adityas:~% python max_of_two.py 4 8
b is bigger
0330 18:37 adityas:~% python max_of_two.py 15 2
a is bigger
0330 18:37 adityas:~%
0330 18:37 adityas:~%
0330 18:37 adityas:~% python max_of_two.py 15
Traceback (most recent call last):
  File "max_of_two.py", line 5, in <module>
    b = int(sys.argv[2])
IndexError: list index out of range
0330 18:37 adityas:~%
```

Write your script in an editor, and then “call” it or “run” it from the command line

In-class example: Hello, world!

And now, a few comments about comments

What is a comment in code?

A comment is a line, or part of a line, that is skipped by the interpreter.

In other words, it's not interpreted. It's just there.

In python, comments start with the pound sign ("#")

```
0330 18:59 adityas:~% python
Python 2.7.6 (default, Sep  9 2014, 15:04:36)
[GCC 4.2.1 Compatible Apple LLVM 6.0 (clang-600.0.39)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> a = 1
>>> b = 2
>>> # I can type whatever I want if I put the pound sign at the beginning of this line
...
>>> I cannot type whatever I want if I forget to do that!
  File "<stdin>", line 1
    I cannot type whatever I want if I forget to do that!
            ^
SyntaxError: invalid syntax
>>>
```

Why do we comment our code?

Help yourself remember what you were thinking

Help other people understand what you were thinking

Help your grader figure out what you were trying to do, and what went wrong!

Commenting for beginners

Your homework MUST HAVE COMMENTS

It’s OK to “over-comment”

Usually you put comments just above / before the part of the program you’re referring to