



OMIS 30: Intro to Programming (with Python)

Week 8, Class 2

Introduction to Programming
Instructor: Denis Vrdoljak



Learn programming with Python



Office Hours

Instructor	Days available
Yuan Wang (our TA)	M 2:30-3:30p, W 9-10a
Mike Davis (other section's instructor)	Tu 9:30-10:20a, Th 12-1p
Denis Vrdoljak	Tu 3:40-4:20p, W 5-6p



Course Topics

- 1. Computer setup - intros
- 2. Shell - cd, mkdir, move, rename, copy, pwd, touch, echo, nano, vim
- 3. Grep, bash, scripts
- 4. Python Basics - print, input, math, PEMDAS
- 5. Pseudo-code, algorithm design, comments
- 6. Loops & Nested Loops
- 7. Moving files around, input, export
- 8. Dates, times, epoch, time-series
- 9. Arrays, lists, dicts, sets, etc.
- 10. And, or, If, elif, try, except
- 11. Functions
- 12. Strings, upper, lower, regex
- 13. Computation time, flops, sorting
- 14. JSON, iteritems
- 15. Pandas
- 16. Jupyter, virtual environments
- 17. Web-apps/web-pages
- 18. Web-scraping
- 19. Plotting, graphing
- 20. Git

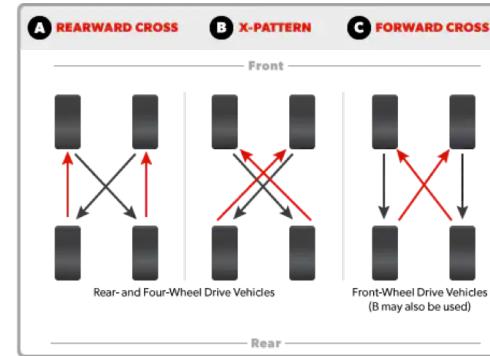


Algorithm Design

- Break down the problem into steps
- Organize the steps to solve the problem
- Think about how a person solves the problem; and break down the human steps into smaller steps
- (for example, finding the smallest number in a list turns into: pick the first number in the list and store it as the smallest value; iterate through the list one number at a time; see if the next value is smaller than the smallest value, if it is smaller store it as the new smallest value)
- Iterate and Optimize

Rotating the tires on a car

- If taken to a mechanic, they'll put the car on a lift, pull all the tires off at once then put them where they go
- If done in your driveway, you'll remove one at a time using your spare as a placeholder





Class Activity

Design pseudocode to rotate your tires, as you would in your driveway

- You can only lift up one corner at a time
- You can use your spare as a placeholder



Class Activity pt.2

You now have a jack, you can lift up an entire side at a time (side, not front or back). Modify your algorithm to work with this.



Python sorting a list

As a human, if you have a list that you want to sort, you will:

- Take the smallest number
- Remove it from the list
- Add it to a new list
- Repeat until the first list is empty



Class Activity II

Turn our list sorter algorithm into Python Code!



Class Activity II

Modify your code so you don't use a second list

- Pseudocode first
- Then turn your pseudocode into python code



Class Activity II

- Congratulations! You've just programmed a Selection Sort.



O(n)

- The complexity which is expressed in Big-O notation is a measure of how the compute time of an algorithm increases as the size of the data increases. For example, iterating through a list in a for loop is $O(n) = n$ complexity.
- If you double the size of the data (n) the compute time doubles
- A nested for loop, where you iterate through the list in each loop (calculating permutations of a loop) is $O(n) = n^2$
- If you double the size of the data you quadruple the runtime



Class Activity III

What is the complexity of the Selection Sort algorithm



Class Activity IV

Can you come up with a more efficient algorithm?



Some Common Sorting Algorithms

- Merge Sort
- Bubble Sort
- Insertion Sort



Some Common Sorting Algorithms

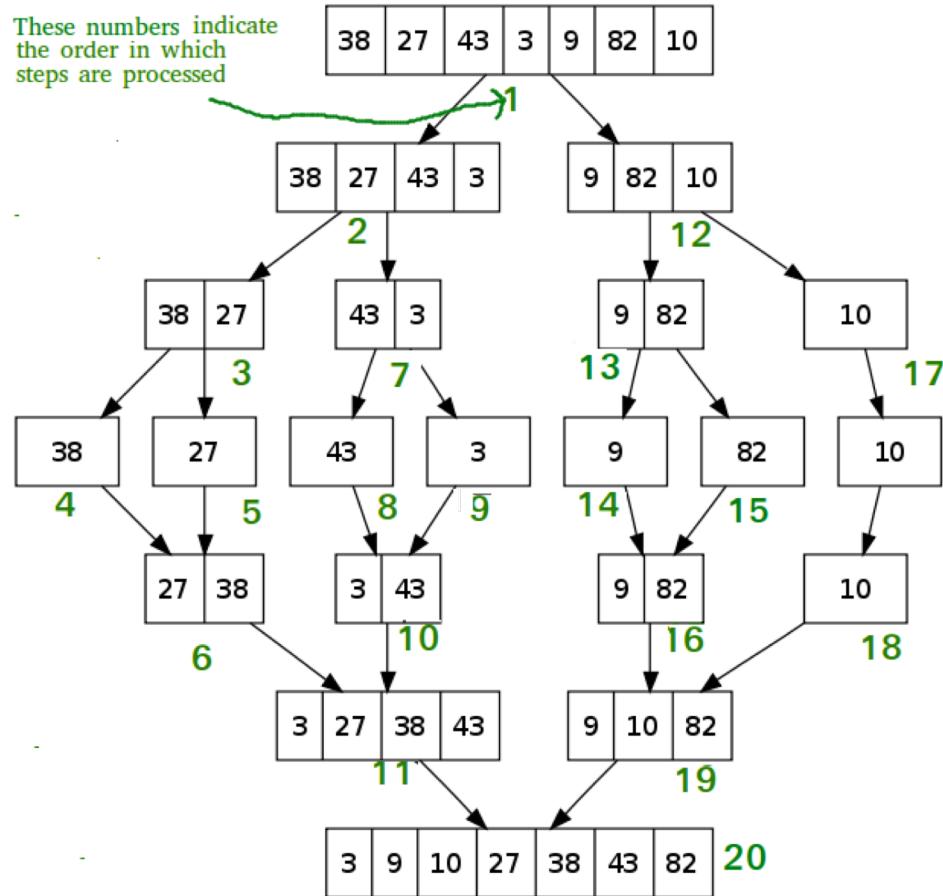
Merge sort

- Mergesort is a comparison-based algorithm that focuses on how to merge together two pre-sorted arrays such that the resulting array is also sorted.



Merge Sort

These numbers indicate
the order in which
steps are processed





Some Common Sorting Algorithms

Insertion sort

- Insertion sort is a comparison-based algorithm that builds a final sorted array one element at a time. It iterates through an input array and removes one element per iteration, finds the place the element belongs in the array, and then places it there.



Insertion Sort

<table border="1"><tr><td>54</td><td>26</td><td>93</td><td>17</td><td>77</td><td>31</td><td>44</td><td>55</td><td>20</td></tr></table>	54	26	93	17	77	31	44	55	20	Assume 54 is a sorted list of 1 item
54	26	93	17	77	31	44	55	20		
<table border="1"><tr><td>26</td><td>54</td><td>93</td><td>17</td><td>77</td><td>31</td><td>44</td><td>55</td><td>20</td></tr></table>	26	54	93	17	77	31	44	55	20	inserted 26
26	54	93	17	77	31	44	55	20		
<table border="1"><tr><td>26</td><td>54</td><td>93</td><td>17</td><td>77</td><td>31</td><td>44</td><td>55</td><td>20</td></tr></table>	26	54	93	17	77	31	44	55	20	inserted 93
26	54	93	17	77	31	44	55	20		
<table border="1"><tr><td>17</td><td>26</td><td>54</td><td>93</td><td>77</td><td>31</td><td>44</td><td>55</td><td>20</td></tr></table>	17	26	54	93	77	31	44	55	20	inserted 17
17	26	54	93	77	31	44	55	20		
<table border="1"><tr><td>17</td><td>26</td><td>54</td><td>77</td><td>93</td><td>31</td><td>44</td><td>55</td><td>20</td></tr></table>	17	26	54	77	93	31	44	55	20	inserted 77
17	26	54	77	93	31	44	55	20		
<table border="1"><tr><td>17</td><td>26</td><td>31</td><td>54</td><td>77</td><td>93</td><td>44</td><td>55</td><td>20</td></tr></table>	17	26	31	54	77	93	44	55	20	inserted 31
17	26	31	54	77	93	44	55	20		
<table border="1"><tr><td>17</td><td>26</td><td>31</td><td>44</td><td>54</td><td>77</td><td>93</td><td>55</td><td>20</td></tr></table>	17	26	31	44	54	77	93	55	20	inserted 44
17	26	31	44	54	77	93	55	20		
<table border="1"><tr><td>17</td><td>26</td><td>31</td><td>44</td><td>54</td><td>55</td><td>77</td><td>93</td><td>20</td></tr></table>	17	26	31	44	54	55	77	93	20	inserted 55
17	26	31	44	54	55	77	93	20		
<table border="1"><tr><td>17</td><td>20</td><td>26</td><td>31</td><td>44</td><td>54</td><td>55</td><td>77</td><td>93</td></tr></table>	17	20	26	31	44	54	55	77	93	inserted 20
17	20	26	31	44	54	55	77	93		



Some Common Sorting Algorithms

Bubble sort

- Bubble sort is a comparison-based algorithm that compares each pair of elements in an array and swaps them if they are out of order until the entire array is sorted. For each element in the list, the algorithm compares every pair of elements.



First pass

Bubble Sort

<table border="1"><tr><td>54</td><td>26</td><td>93</td><td>17</td><td>77</td><td>31</td><td>44</td><td>55</td><td>20</td></tr></table>	54	26	93	17	77	31	44	55	20	Exchange
54	26	93	17	77	31	44	55	20		
<table border="1"><tr><td>26</td><td>54</td><td>93</td><td>17</td><td>77</td><td>31</td><td>44</td><td>55</td><td>20</td></tr></table>	26	54	93	17	77	31	44	55	20	No Exchange
26	54	93	17	77	31	44	55	20		
<table border="1"><tr><td>26</td><td>54</td><td>93</td><td>17</td><td>77</td><td>31</td><td>44</td><td>55</td><td>20</td></tr></table>	26	54	93	17	77	31	44	55	20	Exchange
26	54	93	17	77	31	44	55	20		
<table border="1"><tr><td>26</td><td>54</td><td>17</td><td>93</td><td>77</td><td>31</td><td>44</td><td>55</td><td>20</td></tr></table>	26	54	17	93	77	31	44	55	20	Exchange
26	54	17	93	77	31	44	55	20		
<table border="1"><tr><td>26</td><td>54</td><td>17</td><td>77</td><td>93</td><td>31</td><td>44</td><td>55</td><td>20</td></tr></table>	26	54	17	77	93	31	44	55	20	Exchange
26	54	17	77	93	31	44	55	20		
<table border="1"><tr><td>26</td><td>54</td><td>17</td><td>77</td><td>31</td><td>93</td><td>44</td><td>55</td><td>20</td></tr></table>	26	54	17	77	31	93	44	55	20	Exchange
26	54	17	77	31	93	44	55	20		
<table border="1"><tr><td>26</td><td>54</td><td>17</td><td>77</td><td>31</td><td>44</td><td>93</td><td>55</td><td>20</td></tr></table>	26	54	17	77	31	44	93	55	20	Exchange
26	54	17	77	31	44	93	55	20		
<table border="1"><tr><td>26</td><td>54</td><td>17</td><td>77</td><td>31</td><td>44</td><td>55</td><td>93</td><td>20</td></tr></table>	26	54	17	77	31	44	55	93	20	Exchange
26	54	17	77	31	44	55	93	20		
<table border="1"><tr><td>26</td><td>54</td><td>17</td><td>77</td><td>31</td><td>44</td><td>55</td><td>93</td><td>20</td></tr></table>	26	54	17	77	31	44	55	93	20	93 in place after first pass
26	54	17	77	31	44	55	93	20		



Time in Python

- Modules
- Format
- Measuring time of Algorithms
- Appending a Timestamp to a File



Time Modules in Python

- import time
 - Module that has the basic time related function
- import datetime
 - Module that has the methods to make a string into a datetime



Time - Common Formatting Strings

%b - abbreviated month name

%B - full month name

%d - day of the month (01 to 31)

%e - day of the month (1 to 31)

%H - hour, using a 24-hour clock (00 to 23)

%I - hour, using a 12-hour clock (01 to 12)

%m - month (01 to 12)

%M - minute

%S - second

%T - current time, equal to %H:%M:%S

%y - year without a century (range 00 to 99)

%Y - year including the century

Full list: <https://docs.python.org/3.7/library/time.html>



Measuring time of an Algorithm

- Easy to measure the time of an algorithm
- Wherever you want to ‘start the clock’:

```
import time
```

```
start = time.time()
```

- Wherever you want to ‘stop the clock’:

```
end = time.time()
```

```
print('Total time was:', end - start)
```



Appending a timestamp to a filename

- From the formatting above can make a timestamp by:

```
import time  
time      str = time.strftime("%Y%m%d-%H%M%S")  
filename = 'test_file_' + timestr + '.txt'  
  
with open (filename,'w') as fp:  
  
    ...
```

- Will yield a file named:

test_file_20181108-143015

%Y = 2018, %m = 11, %d = 08, '-' , %H = 14, %M = 30 , %S = 15



Git material moved to Week 9 slides



Appendix

- Reserved keywords in Python:
 - <https://pentangle.net/python/handbook/node52.html>
 - DO NOT USE THESE as VARIABLE NAMES!
- Built-in Functions:
 - <https://docs.python.org/3/library/functions.html>
 - Review/Reference Material:
 - <https://developers.google.com/edu/python/lists>
- Dos/Windows vs Unix/Linux:
 - [https://access.redhat.com/documentation/en-US/Red Hat Enterprise Linux/4/html/Step by Step Guide/ap-doslinux.html](https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/4/html/Step_by_Step_Guide/ap-doslinux.html)
 - You should be familiar with the basic commands for navigating around the file structure and modifying/creating files/folders. You should be aware of the harder to remember ones (like grep and vi) so you know what to Google when you need them!



Appendix

- Learning Python - O'reilly Books (5th Edition)
 - <https://github.com/MrAlex6204/Books/blob/master/Learning%20Python%2C%205th%20Edition.pdf>