

# CS 1340 Introduction to Computing Concepts

Instructor: Xinyi Ding  
Oct 4 2019, Lecture 17



# Announcement

---

- Homework 3 due Today
- Homework 4 posted, due next Friday
- Lab 3 will be posted during weekend

# Agenda

---

- Agenda:
  - Data structures and algorithms

# Modules

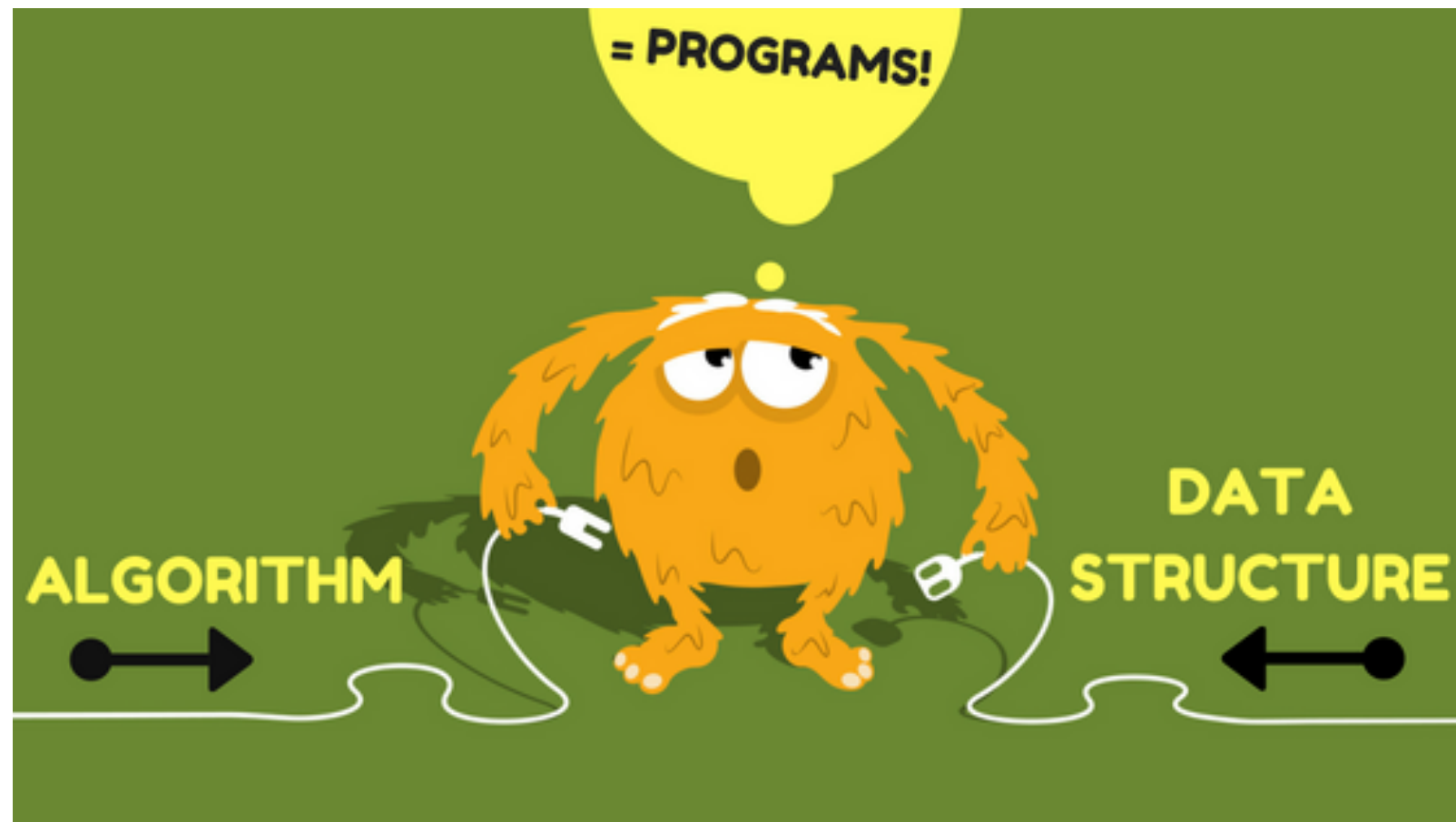
---

- Naming override
  - `math.py`

# Data Structure and Algorithm

---

- Programs = Data Structures + Algorithms



# Data Structure and Algorithm

---

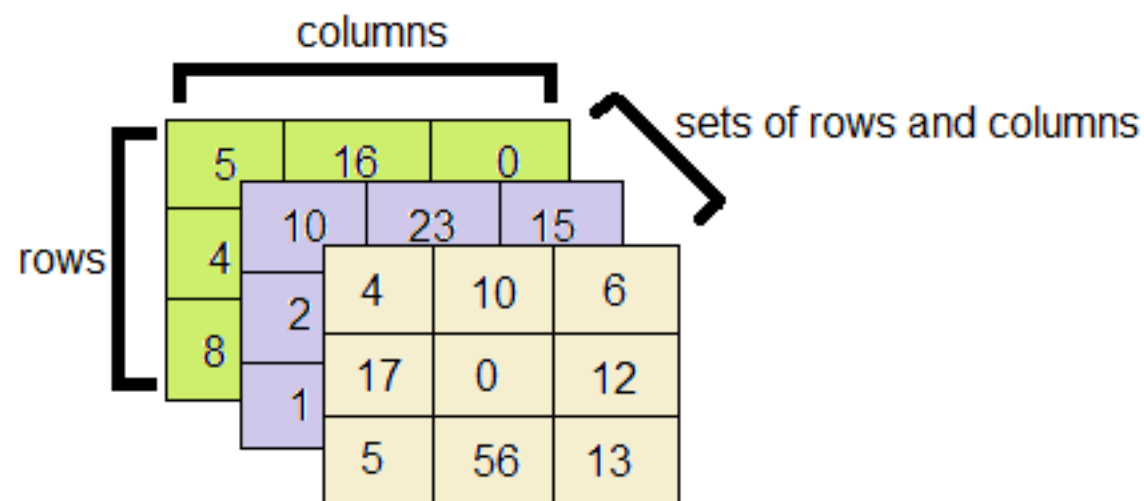
- Common data structures
  - Array (list in Python)
  - Linked-list
  - Queue
  - Stack
  - Trees
  - Graph
  - Hash-Tables
  - ...

# Data Structure and Algorithm

- Multiple dimensional list
  - 2-D list

	Column 0	Column 1	Column 2
Row 0	<b>x[0][0]</b>	<b>x[0][1]</b>	<b>x[0][2]</b>
Row 1	<b>x[1][0]</b>	<b>x[1][1]</b>	<b>x[1][2]</b>
Row 2	<b>x[2][0]</b>	<b>x[2][1]</b>	<b>x[2][2]</b>

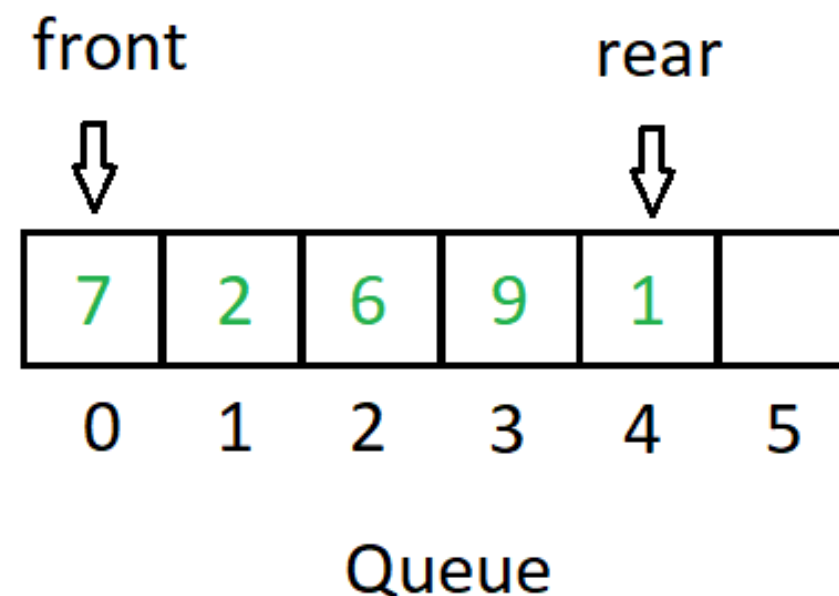
- 3-D list



# Data Structure and Algorithm

---

- Queue
  - first in first out (FIFO)
    - Handle orders
    - Ticket counter line where people who come first will get his ticket first

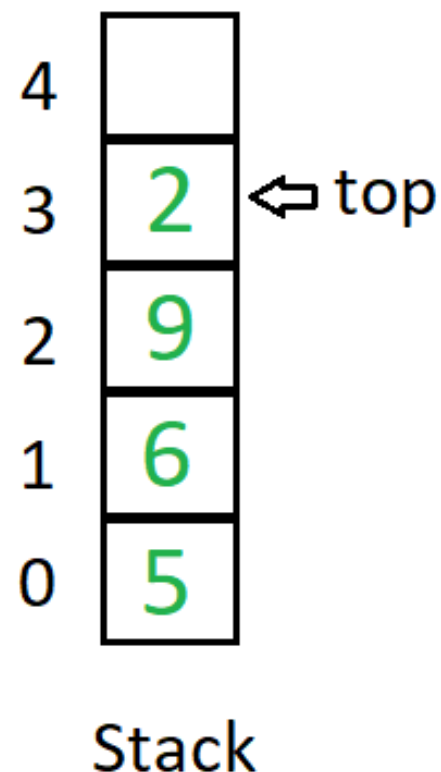




# Data Structure and Algorithm

---

- Stack
  - first in last out (FILO)
    - An "**undo**" mechanism in text editors; this operation is accomplished by keeping all text changes in a stack.
    - **Back/Forward** stacks on browsers.



# Data Structure and Algorithm

---

- Queues and Stacks in Python
  - Use list and pop()

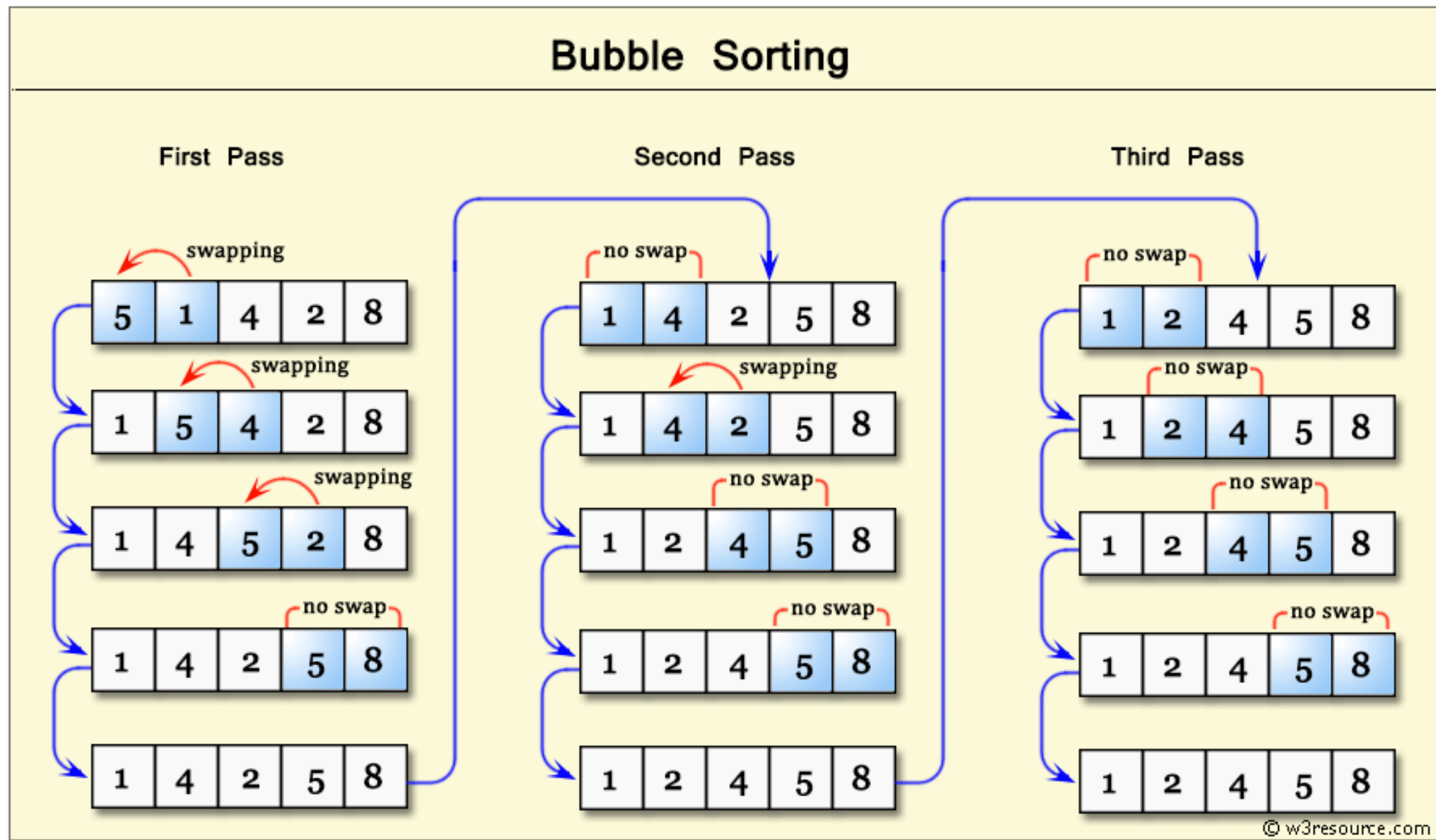
# Data Structure and Algorithm

---

- Common algorithms
  - Sort
    - Bubble sort
    - Selection sort
    - Quick sort
    - ...
  - Search
    - Binary search
    - Breadth First Search (BFS)
    - Depth First Search (DFS)
  - ...

# Data Structure and Algorithm

- Example 1, Bubble sort



# Data Structure and Algorithm

- Example 1, Bubble sort

```
1  # Bubble sort
2  def bubble_sort(nlist):
3      for passnum in range(len(nlist)-1,0,-1):
4          for i in range(passnum):
5              if nlist[i]>nlist[i+1]:
6                  temp = nlist[i]
7                  nlist[i] = nlist[i+1]
8                  nlist[i+1] = temp
9
10     a_list = [34, 1, 2, 10, 12, 9]
11     bubble_sort(a_list)
12     print(a_list)
13
```

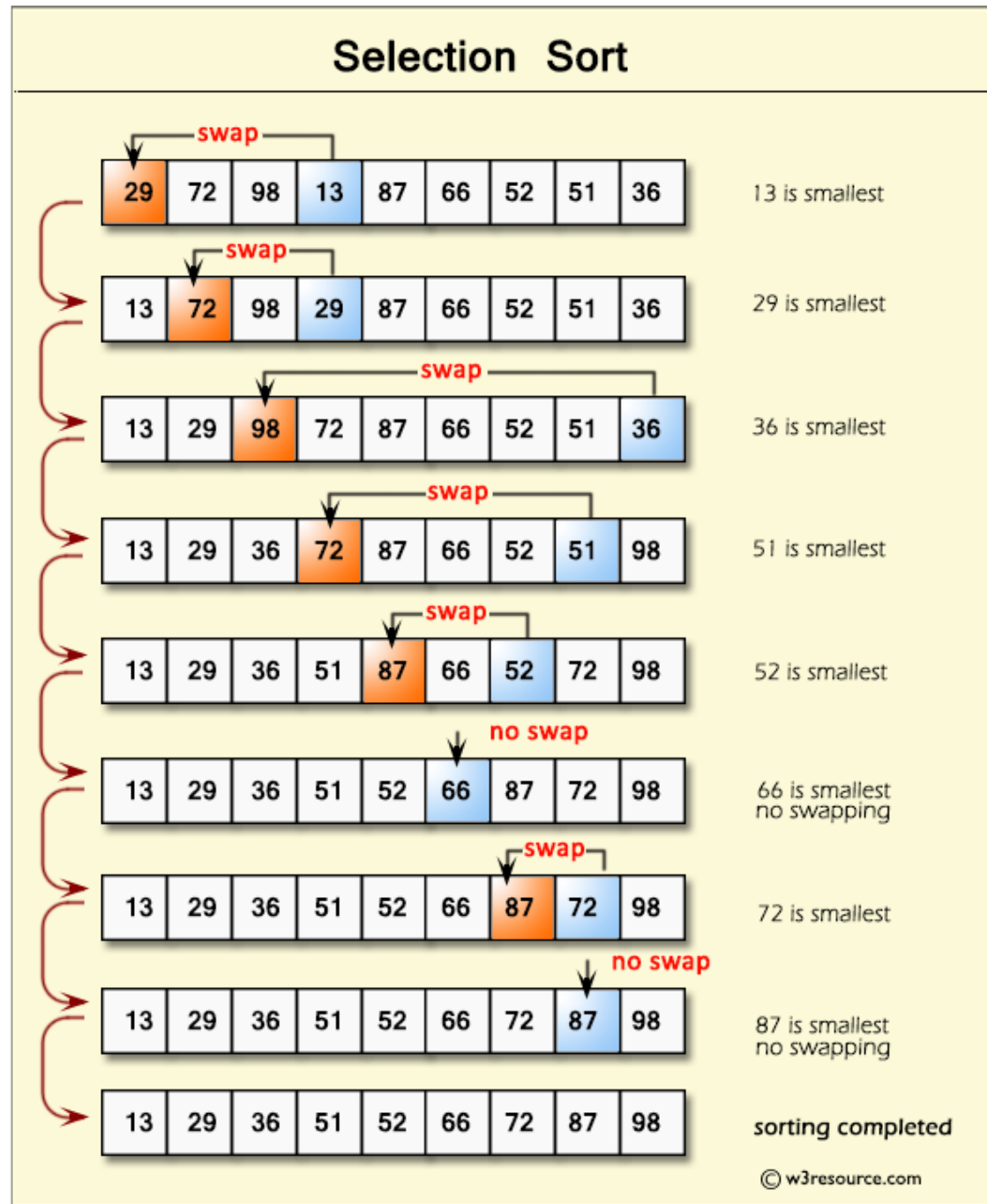
sort\_alg ×

```
/Users/xinyi/anaconda/envs/mlearn/bin/python /Users/xinyi/Courses/cs1340/week6/sort_alg.py
[1, 2, 9, 10, 12, 34]
```

Process finished with exit code 0

# Data Structure and Algorithm

- Example 2, Selection sort



# Data Structure and Algorithm

- Example 2, Selection sort

```
14 # Selection sort
15 def selection_sort(nlist):
16     for fillslot in range(0, len(nlist)):
17         min_index = fillslot
18         for location in range(fillslot, len(nlist)):
19             if nlist[location] < nlist[min_index]:
20                 min_index = location
21
22         temp = nlist[fillslot]
23         nlist[fillslot] = nlist[min_index]
24         nlist[min_index] = temp
25
26
27 a_list = [34, 1, 2, 10, 12, 9]
28 selection_sort(a_list)
29 print(a_list)
30
```

sort\_alg ×

/Users/xinyi/anaconda/envs/mlearn/bin/python /Users/xinyi/Courses/cs1340/week6/sort\_alg.py

[1, 2, 9, 10, 12, 34]

Process finished with exit code 0

# Data Structure and Algorithm

---

- Built in sort() in Python

```
31 a_list = [34, 1, 2, 10, 12, 9]
32 a_list.sort()
33 print(a_list)
```

sort\_alg ×

```
/Users/xinyi/anaconda/envs/mlearn/bin/python /Users/xinyi/Courses/cs1340/week6/sort_alg.py
[1, 2, 9, 10, 12, 34]
```

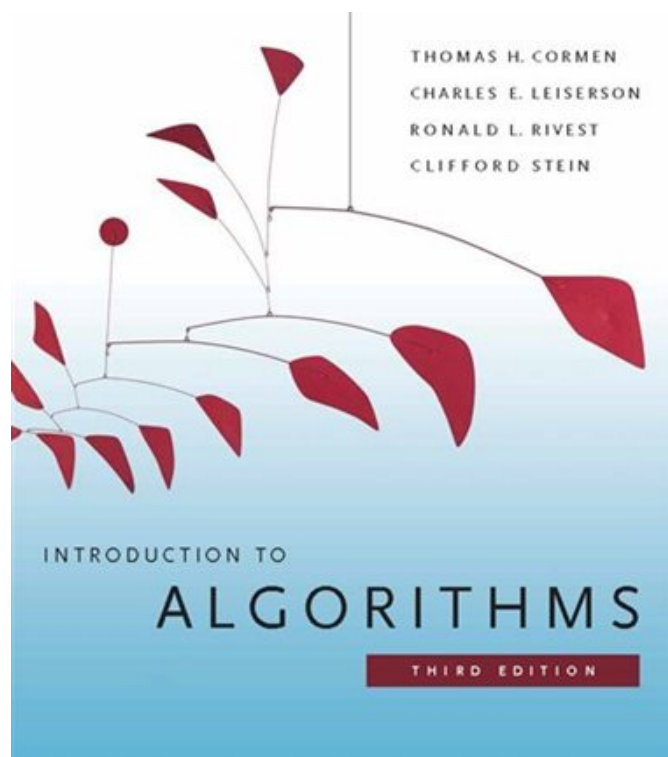
Process finished with exit code 0



# Data Structure and Algorithm

---

- Resources about data structures and algorithms



# Next Week

---

- Jupyter notebook
- Data science
- numpy

