CS21: INTRODUCTION TO COMPUTER SCIENCE

Prof. Mathieson
Fall 2018
Swarthmore College
Outline Nov 19:

- Recap binary search and sorting runtime
- Begin: writing our own classes
- Pie class example
- Student class example

Notes

- Lab 9 due **Monday** after Thanksgiving
- There is lab this week! (Tues/Wed)
- **Ninja session**: Sunday after Thanksgiving 7-10pm
Lab 8 worksheet
(binary search)
Selection Sort Runtime

Selection Sort Runtime

for i in range(n):
    m = i  # min index

for j in range(i+1, n):
    if lst[j] < lst[m]:
        m = j

swap(i, m, lst)

Average # steps: \( \frac{n}{2} \)

i is where min should go where min is now

\( O(n^2) \)

\( \frac{\sum_{i=1}^{n} i^2}{\sum_{i=1}^{n} i} \)

Sorted!

Exact runtime

\( \frac{n(n-1)}{2} \)
Sorting: didn’t get to in class (good practice!)

- `is_sorted(lst)` function to determine if a list is sorted or not
- `sort_runtime.py` to see runtimes in action
Classes
So far with classes…

• We have already seen existing classes (Point, Circle, Line, etc)
So far with classes…

- We have already seen existing classes (Point, Circle, Line, etc)
- Classes allow us to encapsulate common data structures and actions so we don’t have to define them over and over again
So far with classes…

• We have already seen existing classes (Point, Circle, Line, etc)
• Classes allow us to encapsulate common data structures and actions so we don’t have to define them over and over again
• We can create a new instance of a class using the constructor

```
dot = Circle(Point(x,y), r)
```
So far with classes…

- We have already seen existing classes (*Point, Circle, Line*, etc)
- Classes allow us to encapsulate common data structures and actions so we don’t have to define them over and over again
- We can create a new instance of a class using the *constructor*
  ```python
dot = Circle(Point(x,y),r)
```
- We can access the instance’s data using *methods*
  ```python
r = dot.getRadius()
```
So far with classes…

- We have already seen existing classes (Point, Circle, Line, etc)
- Classes allow us to encapsulate common data structures and actions so we don’t have to define them over and over again
- We can create a new instance of a class using the constructor
  
  ```
  dot = Circle(Point(x,y),r)
  ```
- We can access the instance’s data using methods
  
  ```
  r = dot.getRadius()
  ```
- We can use/modify class instances using methods
  
  ```
  dot.move(dx,dy)
  ```
Motivation for classes: LOLs

- List-of-lists let us keep track of things that should be “together”, but they get cumbersome to modify:

```python
>>> pie_lst = [['apple', 8], ['cherry', 8], ['chocolate', 8]]
>>> pie_lst[2][1] -= 1
>>> pie_lst
[['apple', 8], ['cherry', 8], ['chocolate', 7]]
```