CS21: INTRODUCTION TO COMPUTER SCIENCE

Prof. Mathieson

Fall 2017

Swarthmore College

Outline Nov 6:

- Linear Search (searches.py)
- Wed: hand back Quiz 3

Notes

•I will finish TDD reviews by the start of lab on Tuesday
•Let me know if you want intermediate feedback
•Make sure to copy (**cp**) your design into a new file!

Linear Search

This week:

* Searching

Nex week:

* sorting

Motivation for searching:

data = [webpage1, webpage2, ...]

query = "internship in CS"

(complex relationship between query & elements of list)

index

0 1 2 3 4 5 6

data = [7, 10, 3, 2, 8, 3, 0]

query = 3

Ideal: "in" operator. (boolean)

While loop

```
def linear_search(query, lst):
    found = False
    i = 0
    while not found and i < len(lst):
        if query == lst[i]:
            found = True
        (idea: list accumulator)
        i += 1
    return found
```

i += 1

i = i + 1

Idea 2:

for loop

Idea 2: where does the element occur (index)

data.index(query)

for loop

i < len(lst):

)

```
def linear_search_index(query, lst):
    """Return the index (int) of the first occurence
                                        of query"""
    for i in range(len(lst)):
        if query == lst[i]:
            return i  # Stop the function!
    return -1 ← None
```

example: n=7

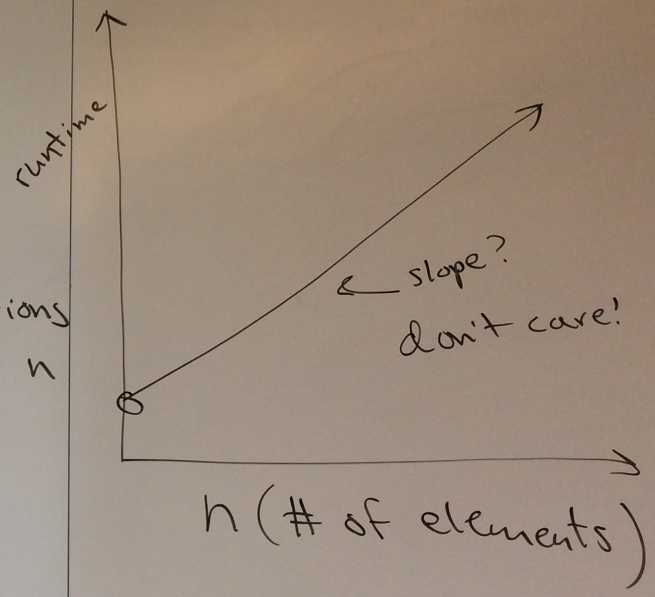
Runtime:

of elements in lst: (n)

(worst case)

↳ $2n + 3$ ← # of operations in terms of n

↳ ≈ $O(n)$ "order"



Pass vs. continue

Pass vs. Continue

- **pass** is typically used when you don't want to do anything, but need an indentation block for syntax reasons
- I almost always remove **pass** when the code is completely implemented, but it can be useful for readability
- **continue** will skip over remaining code in the loop and start the next iteration
- Try to avoid: **while True**, **break**, **continue**

```
lst = [1,2,3]

for elem in lst:
    if elem < 10:
        pass
    print(elem)
```

Output:
1
2
3

```
lst = [1,2,3]

for elem in lst:
    if elem < 10:
        continue
    print(elem)
```

Output: