CS21: INTRODUCTION TO COMPUTER SCIENCE

Prof. Mathieson

Fall 2018

Swarthmore College

Outline Nov 9:

- Binary search code
- Creating and using a search library
- Runtime of binary search
- Dictionary example (searching file)

Notes

•Office Hours **today 3-5pm!** (will probably be in lab)
•Lab 7 due tomorrow night (Saturday)

Mini-quiz (discuss with a partner)

1) How many parameters (inputs) do **linear search** and **binary search** need? What are they?

2) What is the most useful type of output (**return** value) from a search function?

3) Under what conditions can we use binary search?

4) If my **query** is 5 and my **list** is [-9, 2, 4, 5, 15, 25], what is the output of linear search? What is the output of binary search?

5) Same list, now my query is 7. What will be returned from each search? What will **low** and **high** be at the end of binary search?

lst[:n]

lst[3:5]

① 2: query & data

② location of query (index (int))

③ <u>sorted</u> list

④ output: 3 (both)

⑤ output: -1 (both)

low	mid	high
0	2	5
3	4	5
3	3	3
4		3

STOP

[-9, 2, 4, 5, 5, 25]

== 7? == 7? == 7?

0 1 2 3 4 5

7 would go <u>here</u>

$\frac{0+5}{2} = 2.5 \to 2$

Linear vs. Binary Search (dictionary example)

Reading the dictionary example

```
from searches import *

def read_dictionary(filename):
    """
    Purpose: read a dictionary file and produce a list of all the words
    Parameter: filename (a string representing the path to the dictionary file)
    Return: a list of strings
    """
    # open the file in read mode (not write, that will overwrite the file!)
    word_file = open(filename,'r')
    word_lst = [] # set up list accumulator

    # loop through each line (one word on each line)
    for line in word_file:
        word = line.strip()
        word_lst.append(word)

    word_file.close()
    return word_lst
```