CS 152: Review

CS 152: Python for STEM

Colorado State University
Computer Science Department
Slides Originally Created by Marcia Moraes

Colorado State University

Weekly Announcements!

TODO Reminders:

- Reading 17 (zybooks) – more recursion algorithms ☺
- Lab 11
- Wednesday Review Lecture
- Thursday Review in Lab
- Friday Oct 11th : Final Coding Exam and Final Canvas Exam

Binary Search

- Search over a sorted list
- Start from the middle point
- Check to see if that is the element you are looking for, if yes, return the index where you found the element
- If not, test if the number is less than the middle element, if yes, search from the beginning until the middle point
- If the number is greater than the middle element, search from the middle point until the end of the list

Binary Search

```
def binary_search(numbers, key):
    low = 0
    high = len(numbers) - 1

    while high >= low:
        mid = (high + low) // 2
        if numbers[mid] < key:
            low = mid + 1
        elif numbers[mid] > key:
            high = mid - 1
        else:
            return mid
    return -1 # not found
```

Binary Search

```
numbers = [2, 4, 7, 10, 11, 32, 45, 87]
print('NUMBERS:', end=' ')
for num in numbers:
    print(num, end=' ')
print()

key = int(input('Enter a value: '))
key_index = binary_search(numbers, key)

if key_index == -1:
    print(str(key) + ' was not found.')
else:
    print('Found ' + str(key) + ' at index ' + str(key_index) + '.')
```

Binary Search

```
def binary_search(numbers, key):
    low = 0
    high = len(numbers) - 1

    while high >= low:
        mid = (high + low) // 2
        if numbers[mid] < key:
            low = mid + 1
        elif numbers[mid] > key:
            high = mid - 1
        else:
            return mid
    return -1 # not found
```

Let’s think about the recursive version of the binary search

What is our case base?

What is our recursive call?

How many parameters our function needs?

Coding Along

- Write a function that receives a list of integers as a parameter and return a new list containing the elements of the original list without repetitions.
- Write a function that receives two lists of integers as parameters and returns a list that is the union of both list. The union list cannot contain any repeated element.
- Write a function that receives two lists of integer as parameters and returns a list that is the intersection of both lists.