http://www.cs.cornell.edu/courses/cs1110/2018sp

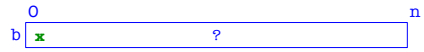
Lecture 27: Sorting & Searching

CS 1110

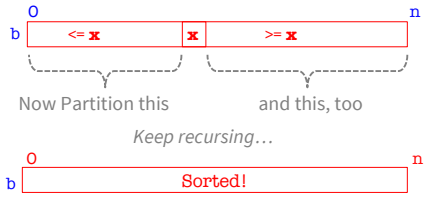
Introduction to Computing Using Python

[E. Andersen, A. Bracy, D. Gries, L. Lee, S. Marschner, C. Van Loan, W. White]

Sorting with Partitions



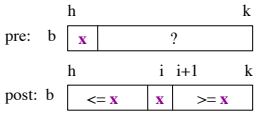
- **Idea:** Pick a *pivot* element **x**
- Partition sequence into <= x and >= x



5

QuickSort

```
def quick_sort(b, h, k):
    """Sort the array fragment b[h..k]"""
    if k<=h:
        return
    i = partition(b, h, k)
    # INV: b[h..i-1] <= b[i] <= b[i+1..k]
    # Sort b[h..i-1] and b[i+1..k]
    quick_sort (b, h, i-1)
    quick_sort (b, i+1, k)
```



6

https://www.youtube.com/watch?v=m1PS8IR6Td0

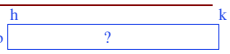
Linear Search Definition

- **Vague**: Find first occurrence of v in b[h..k-1].
- **Better**: Store an integer in **i** to make this post-condition true:
  post: 1. v is not in b[h..**i**-1]
  2. **i** = k OR v = b[**i**]

9

Implementing Linear Search

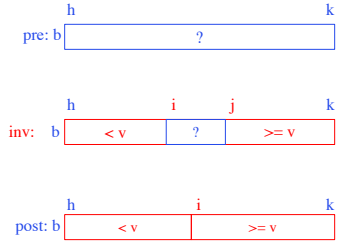
```
def linear_search(b,v,h,k):
    """Returns: first occurrence of v in b[h..k-1]"""
    # Store in i index of the first v in b[h..k-1]
    i = h

    # invariant: v is not in b[0..i-1]
    while i < k and b[i] != v:
        i = i + 1

    # post: v is not in b[h..i-1]
    #       i >= k or b[i] == v
    return i if i < k else -1
```



11

Binary Search: What's the Invariant?

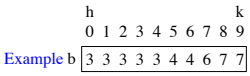
- Look for v in **sorted** sequence segment b[h..k].
  - Precondition: b[h..k-1] is sorted (in ascending order).
  - Postcondition: b[h..i-1] < v and v <= b[i..k]



Called binary search because each iteration of the loop cuts the array segment still to be processed in half

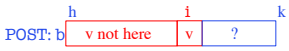
13

Q: Binary Search Examples



- if v is 3, set i to ?
- if v is 4, set i to ?
- if v is 5, set i to ?
- if v is 8, set i to ?

A: 0
B: 3
C: 5
D: 7
E: None of the Above



OR

Implementing Binary Search