Topic 24
Sorting and Searching arrays

"There's nothing in your head the sorting hat can't see. So try me on and I will tell you where you ought to be."

-The Sorting Hat,
Harry Potter and the Sorcerer's Stone

Searching

- Given an array of ints find the index of the first occurrence of a target int

index	*0*	*1*	*2*	*3*	*4*	*5*
value	89	0	27	-5	42	11

- Given the above array and a target of 27 the method returns 2
- What if not present?
- What if more than one occurrence?

Clicker 1

- Given an array with 1,000,000 distinct elements in random order, how many elements do you expect to look at (on average) when searching if:

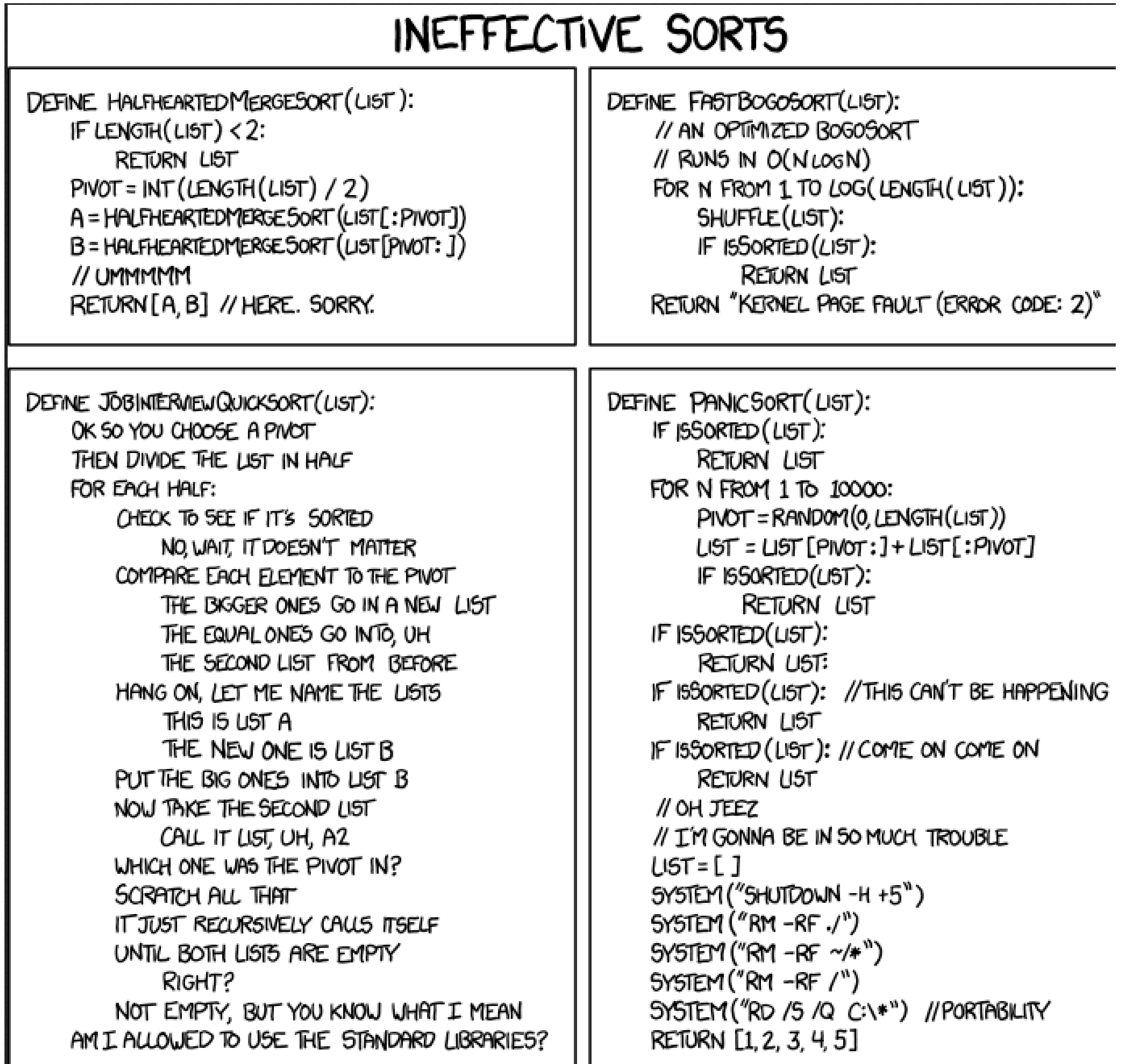
	item present	item not present
A.	1	1,000,000
B.	500,000	1,000,000
C.	1,000,000	1,000,000
D.	1,000	500,000
E.	20	1,000,000

linear or sequential search

Sorting

XKCD

http://xkcd.com/1185/



Sorting

- A fundamental application for computers
- Done to make finding data (searching) faster
- Many different algorithms for sorting
- One of the difficulties with sorting is working with a fixed size storage container (array)
  - if resize, that is expensive (slow)
  - Trying to apply a human technique of sorting can be difficult
  - try sorting a pile of papers and clearly write out the algorithm you follow

Selection Sort

- To sort a list into ascending order:
  - Find the smallest item in an array, the minimum
  - Put that value in the first element of the array
    - Where to put the value that was in the first location?
  - And now…?

Selection Sort in Practice

44 68 191 119 119 37 83 82 191 45 158 130 76 153 39 25

http://tinyurl.com/d7kxxxf

animation of selection sort algorithm

Implementation of Selection Sort

- Include println commands to trace the sort

Clicker 2

- Determine how long it takes to sort an array with 100,000 elements in random order using selection sort. When the number of elements is increased to 200,000 how long will it take to sort the array?

A. About the same
B. 1.5 times as long
C. 2 times as long
D. 4 times as long
E. 8 times as long

Insertion Sort

- Another of the Simple sort
- The first item is sorted
- Compare the second item to the first
  - if smaller swap
- Third item, compare to item next to it
  - need to swap
  - after swap compare again
- And so forth…

Insertion Sort in Practice

44 68 191 119 119 37 83 82 191 45 158 130 76 153 39 25

http://tinyurl.com/d8spm2l

animation of insertion sort algorithm

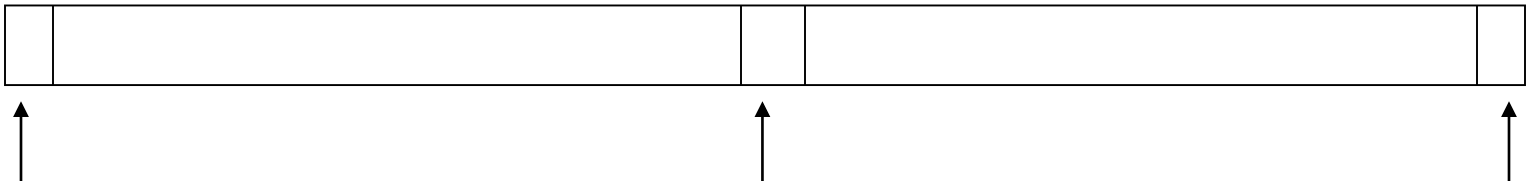
Binary Search

Searching in a Sorted List

- If items are sorted then we can *divide and conquer*
- dividing your work in half with each step
  - generally a good thing
- The Binary Search on List in Ascending order
  - Start at middle of list
  - is that the item?
  - If not is it less than or greater than the item?
  - less than, move to second half of list
  - greater than, move to first half of list
  - repeat until found or sub list size = 0

Binary Search

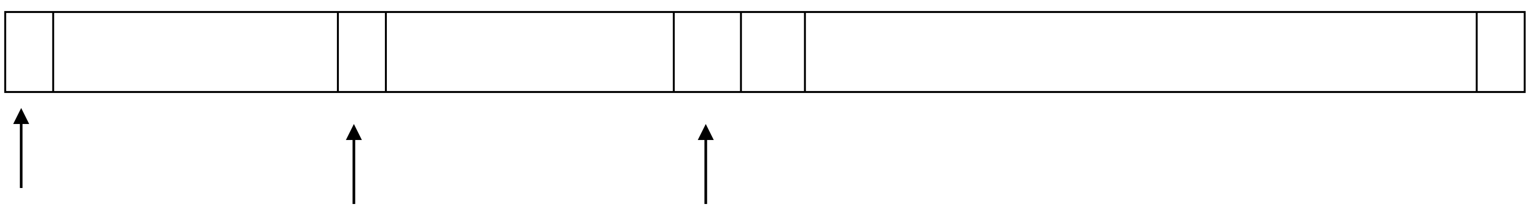
list



low item middle item high item

Is middle item what we are looking for? If not is it more or less than the target item? (Assume lower)

list



low item middle item high item

and so forth…

Implement Binary Search

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
2	3	5	7	11	13	17	19	23	29	31	37	41	43	47	53

Trace When Key == 3
Trace When Key == 30

Variables of Interest?

Clicker 3

- Given an array with 1,000,000 elements in sorted order, how many elements do you expect to look at when searching (with binary search) for a value if:

	item present once	item not present
A.	1	500,000
B.	20	20
C.	1	1,000,000
D.	1,000	500,000
E.	1,000	1,000