

Today: Searching and sorting arrays

Notes: www.kamperh.com/cp143/

Computer Programming 143 – Lecture 15

Arrays III

Faculty of Engineering



6.10 Searching Arrays

Binary search

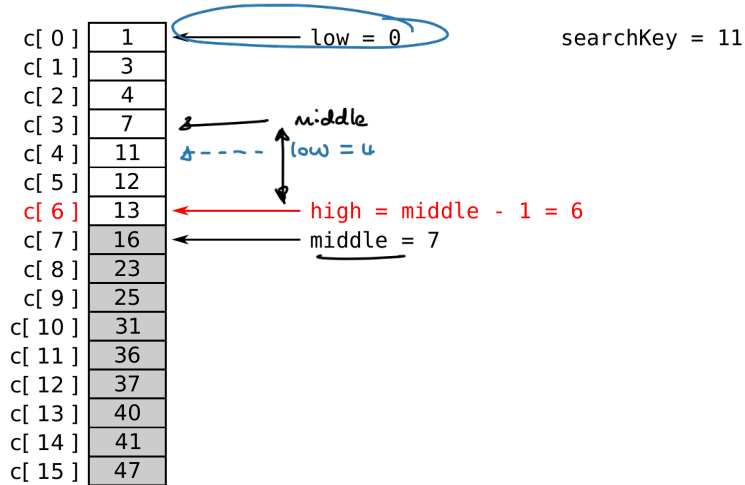
- Only for arrays **sorted** by key
- Compares **middle** element with **key**
 - If equal, match found
 - If **key** < **middle**, looks further in lower half of array
 - If **key** > **middle**, looks further in upper half of array
 - Repeat

6.10 Binary Search

Index	Values	
c[0]	1	← low = 0
c[1]	3	
c[2]	4	
c[3]	7	
c[4]	11	
c[5]	12	
c[6]	13	high = 6
c[7]	16	middle
c[8]	23	
c[9]	25	
c[10]	31	
c[11]	36	
c[12]	37	
c[13]	40	
c[14]	41	
c[15]	47	high = 15

searchKey = 11

6.10 Binary Search



6.10 Binary Search

searchKey = 11

c[0]	1	
c[1]	3	
c[2]	4	
c[3]	7	← middle = 3
c[4]	11	← low = middle + 1 = 4
c[5]	12	← middle
c[6]	13	← high = 6
c[7]	16	
c[8]	23	
c[9]	25	
c[10]	31	
c[11]	36	
c[12]	37	
c[13]	40	
c[14]	41	
c[15]	47	

6.10 Binary Search

c[0]	1
c[1]	3
c[2]	4
c[3]	7
c[4]	11
c[5]	12
c[6]	13
c[7]	16
c[8]	23
c[9]	25
c[10]	31
c[11]	36
c[12]	37
c[13]	40
c[14]	41
c[15]	47

```
searchKey = 11
```

low = 4

```
middle = 5
```

high = 6

```
searchKey < c[ middle ]
```

6.10 Binary Search

searchKey = 11

c[0]	1	
c[1]	3	
c[2]	4	
c[3]	7	
c[4]	11	$\leftarrow \text{high} = \text{middle} - 1 = 4$
c[5]	12	$\leftarrow \text{low} = 4$
c[6]	13	$\leftarrow \text{middle} = 5$
c[7]	16	
c[8]	23	
c[9]	25	
c[10]	31	
c[11]	36	
c[12]	37	
c[13]	40	
c[14]	41	
c[15]	47	

6.10 Binary Search

4 3 2 1

c[0]	1
c[1]	3
c[2]	4
c[3]	7
c[4]	11
c[5]	12
c[6]	13
c[7]	16
c[8]	23
c[9]	25
c[10]	31
c[11]	36
c[12]	37
c[13]	40
c[14]	41
c[15]	47

searchKey = 11

high = 4

low = 4

middle = (high + low) / 2 = 4

c[middle] == searchKey

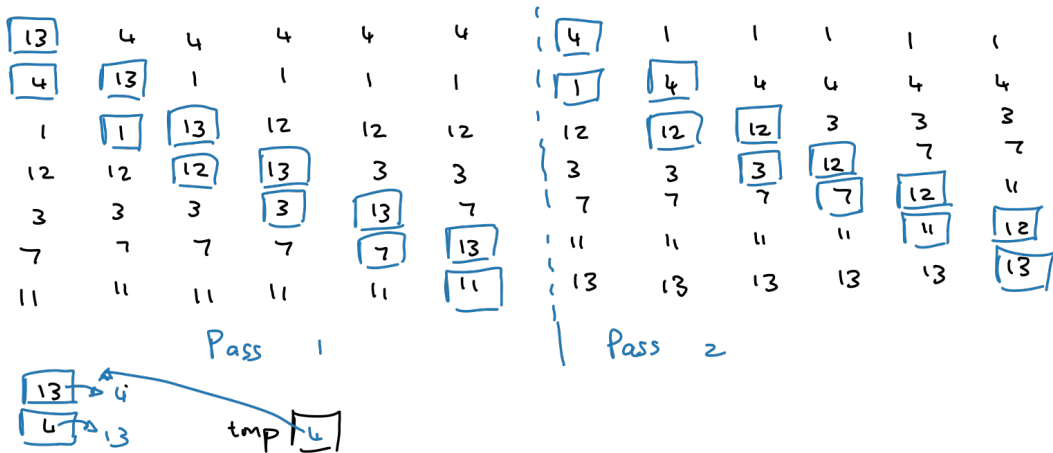
$$2^4 = 16$$

$2^{\text{steps}} = \text{elements}$

steps = $\log_2 \text{elements}$

How many steps does the search take in the worst case?

An array with 32 elements



6.8 Bubble Sort

c[0]	13	pass = 1 i = 0
c[1]	4	
c[2]	1	
c[3]	12	
c[4]	3	
c[5]	7	
c[6]	11	

6.8 Bubble Sort

c[0]	4	pass = 1 i = 0
c[1]	13	
c[2]	1	
c[3]	12	
c[4]	3	
c[5]	7	
c[6]	11	

6.8 Bubble Sort

c[0]	4	pass = 1 i = 1
c[1]	13	
c[2]	1	
c[3]	12	
c[4]	3	
c[5]	7	
c[6]	11	

6.8 Bubble Sort

c[0]	4		pass = 1 i = 1
c[1]	1		
c[2]	13		
c[3]	12		
c[4]	3		
c[5]	7		
c[6]	11		

6.8 Bubble Sort

c[0]	4	pass = 1 i = 2
c[1]	1	
c[2]	13	
c[3]	12	
c[4]	3	
c[5]	7	
c[6]	11	

6.8 Bubble Sort

c[0]	4	pass = 1 i = 2
c[1]	1	
c[2]	12	
c[3]	13	
c[4]	3	
c[5]	7	
c[6]	11	

6.8 Bubble Sort

c[0]	4	pass = 1 i = 3
c[1]	1	
c[2]	12	
c[3]	13	
c[4]	3	
c[5]	7	
c[6]	11	

6.8 Bubble Sort

c[0]	4	pass = 1 i = 3
c[1]	1	
c[2]	12	
c[3]	3	
c[4]	13	
c[5]	7	
c[6]	11	

6.8 Bubble Sort

c[0]	4	pass = 1 i = 4
c[1]	1	
c[2]	12	
c[3]	3	
c[4]	13	
c[5]	7	
c[6]	11	

6.8 Bubble Sort

c[0]	4	pass = 1 i = 4
c[1]	1	
c[2]	12	
c[3]	3	
c[4]	7	
c[5]	13	
c[6]	11	

6.8 Bubble Sort

c[0]	4	pass = 1 i = 5
c[1]	1	
c[2]	12	
c[3]	3	
c[4]	7	
c[5]	13	
c[6]	11	

6.8 Bubble Sort

c[0]	4	pass = 1 i = 5
c[1]	1	
c[2]	12	
c[3]	3	
c[4]	7	
c[5]	11	
c[6]	13	

6.8 Bubble Sort

c[0]	4	pass = 2 i = 0
c[1]	1	
c[2]	12	
c[3]	3	
c[4]	7	
c[5]	11	
c[6]	13	

6.8 Bubble Sort

c[0]	1	pass = 2 i = 0
c[1]	4	
c[2]	12	
c[3]	3	
c[4]	7	
c[5]	11	
c[6]	13	

6.8 Bubble Sort

c[0]	1	pass = 2 i = 1
c[1]	4	
c[2]	12	
c[3]	3	
c[4]	7	
c[5]	11	
c[6]	13	

6.8 Bubble Sort

c[0]	1	pass = 2 i = 2
c[1]	4	
c[2]	12	
c[3]	3	
c[4]	7	
c[5]	11	
c[6]	13	

6.8 Bubble Sort

c[0]	1	pass = 2 i = 2
c[1]	4	
c[2]	3	
c[3]	12	
c[4]	7	
c[5]	11	
c[6]	13	

6.8 Bubble Sort

c[0]	1	pass = 2 i = 3
c[1]	4	
c[2]	3	
c[3]	12	
c[4]	7	
c[5]	11	
c[6]	13	

6.8 Bubble Sort

c[0]	1	pass = 2 i = 3
c[1]	4	
c[2]	3	
c[3]	7	
c[4]	12	
c[5]	11	
c[6]	13	

6.8 Bubble Sort

c[0]	1	pass = 2 i = 4
c[1]	4	
c[2]	3	
c[3]	7	
c[4]	12	
c[5]	11	
c[6]	13	

6.8 Bubble Sort

c[0]	1	pass = 2 i = 4
c[1]	4	
c[2]	3	
c[3]	7	
c[4]	11	
c[5]	12	
c[6]	13	

6.8 Bubble Sort

c[0]	1	pass = 2 i = 5
c[1]	4	
c[2]	3	
c[3]	7	
c[4]	11	
c[5]	12	
c[6]	13	

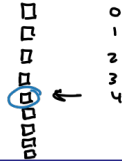
6.8 Bubble Sort

c[0]	1	pass = 3 i = 0
c[1]	4	
c[2]	3	
c[3]	7	
c[4]	11	
c[5]	12	
c[6]	13	

6.8 Bubble Sort

c[0]	1	pass = 3 i = 5
c[1]	3	
c[2]	4	
c[3]	7	
c[4]	11	
c[5]	12	
c[6]	13	

6.9 Case Study: Computing Mean, Median and Mode



Problem

- Ninety nine responses were collected in a survey. Each response is a number between 1 and 9. Place the responses in an array and summarize the results by giving the mean, median and mode of the survey data.

Homework

- Do the mean, median, mode example
- Study Sections 6.8-6.9 in Deitel & Deitel
- Do Self Review Exercise 6.1(d) in Deitel & Deitel
- Do Exercises 6.14 in Deitel & Deitel