ITC 250-CPET 499 Web Systems, Oct. 29

Tuesday, October 29, 2013 1:18 PM

Topics of Discussion

- **Chapter 10 JavaScript: Arrays**
  - Arrays and Functions
  - Array & Functions
    - Pass-by-Value
    - Pass-by-Reference
    - Example Fig. 10-13, Fig. 10-14
  - Sorting Arrays
    - Bubble sort (C programming example)
      - C Arrays & Applications, by Paul Lin, http://www.etcs.ipfw.edu/~lin/ECET264/2011-F/Lectures/264_Lecture-11_Arrays.pdf
    - Method: sort()
      - Example Fig. 10-16
  - Searching Arrays with Array Method: indexOf()
      - Example Fig. 10-18
  - Multidimensional Arrays
    - Two-D array layout: 3-rows, 4-columns
      a[0][0], a[0][1], a[0][2], a[0][3]
      a[1][0], a[1][1], a[1][2], a[1][3]
      a[2][0], a[2][1], a[2][2], a[2][3]
  - Examples of 2-D array Declaration and Assignment
    - Example 1: 2 x 3 Array

```
var b;
b = new Array(2); // allocate 2-rows
b[0] = new Array (5); // allocate 5 columns for row 0
b[1] = new Array(3);  // allocate 3 columns for row 1
```

    - Example 2:

```
var array1 = [[ 1,   2,  3], // row 0
              [ 4,  5,  6]]; // row 1
var array2 = [[1,  2],       // row 0
              [3],           // row 1
              [4,  5,  6];   // row 2
```

Chapter 11. JavaScript: Objects

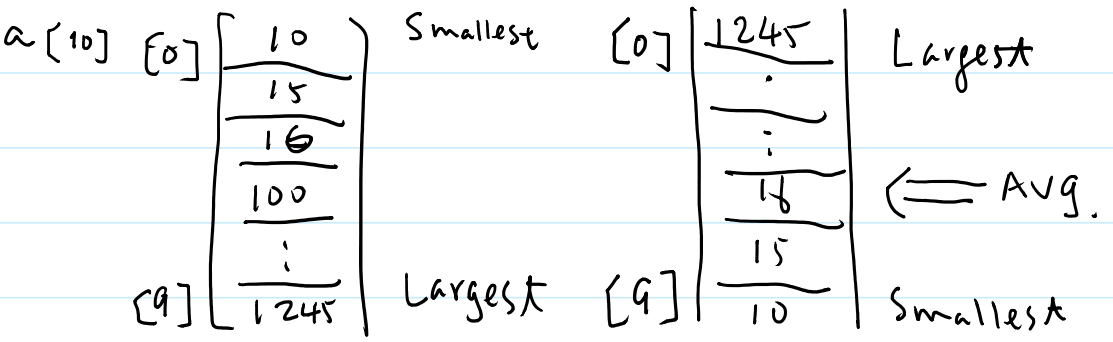
- **Math Object**
- **String Object**
- **Date Object**
- **Document Object**
- **Math object: Methods**
sqrt(x)	abs(x)	
ceil(x)	floor(x)	
round(x)		
cos(x)	sin(x)	tan(x)
exp(x)	log(x)	pow(x,y)
max(x, y)	min(x,y)
- **Properties of Math object**

Math.E	Base of Natural Log e	2.718

Math.LN2	Natural Log of 2	0.693
Math.LN10	Natural Log of 10	2.302
Math.LOG2E	Base 2 of Log of e	1.442
Math.LOG10	Base 10 of Log 10	0.434
Math.PI	Π the ratio of a circle's circumference to its diameter	3.141592653589793
Math.SQRT1_2	Square root of 0.5	0.707
Math.SQRT2	Square root of 2.0	1.414

- String Object
- Character Processing Methods

charAt()	Returns the character at a specific location
charCodeAt()	Returns the Unicode value of the char at a specific location
fromCharCode()	Returns a string created from a series of Unicode values
toLowerCase()	
toUpperCase()	
concat(string)	
indexOf(substring, index)	
lastIndexOf(substring, index)	
Replace(searchString, replaceString	
slice(start, end)	
split(string)	
substr(start, length)	
substring(start, length)	



H.R. Employee DB.

PID

$Avg = \frac{n_L + n_S}{2}$

Banking App. Acc #

Search data items in array

" d.B.

C, humidity, Noise Level

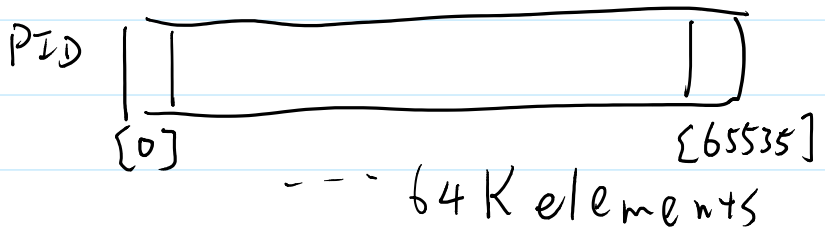
Noise Level

" d.B.

" multiple d.B

" around the world

(Sequential search
(Binary ")



--- 64K elements

Worse case # of comparison n array size

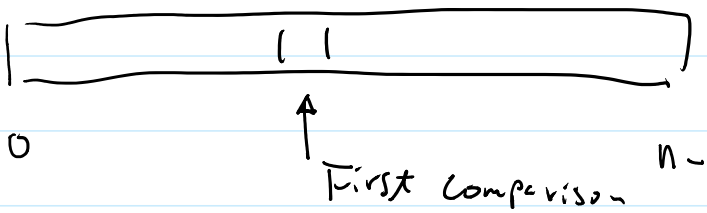
if (ID == PID[ ]) • member
• subscript

Binary search $2^{16} = 2 \times 2 \times \cdots \times 2 = 65536$

$= 64K$

$= 64 \times 1024$

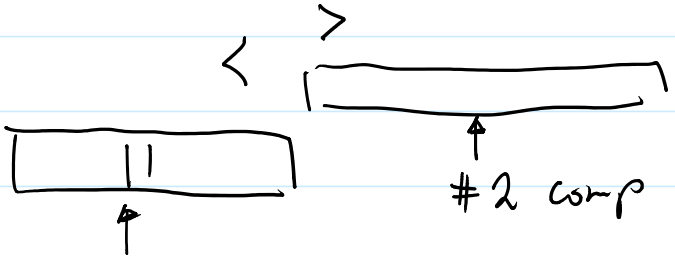
* max # of comparison is 16 time



$n = 2^{16}$

First comparison

index = $\frac{0+n-1}{2}$ = 36767, 68,



16 times

16 times

Comp  match / no match