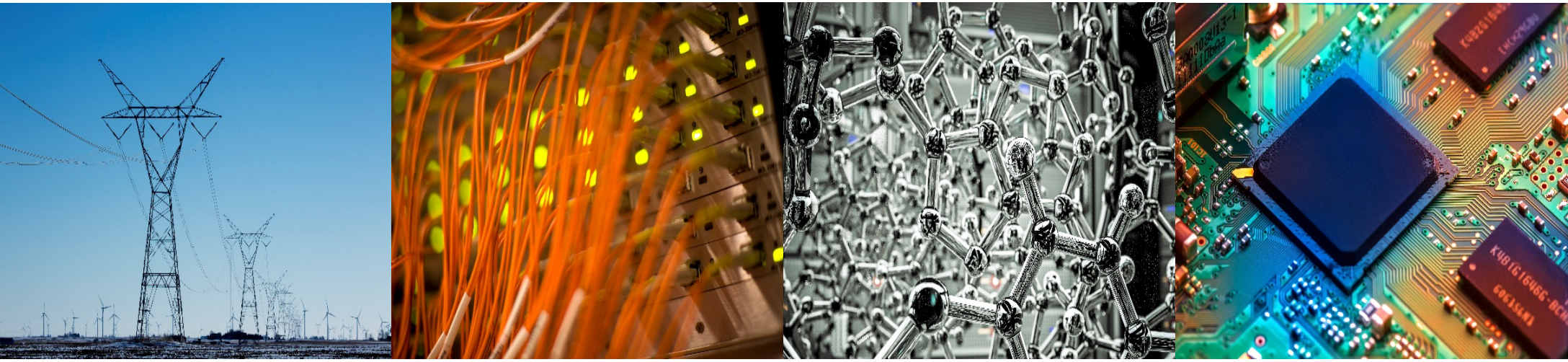
ECE 220 Computer Systems & Programming

Lecture 18 – Problem Solving with Pointers and Arrays

July 10, 2020



ILLINOIS
Electrical & Computer Engineering
GRAINGER COLLEGE OF ENGINEERING

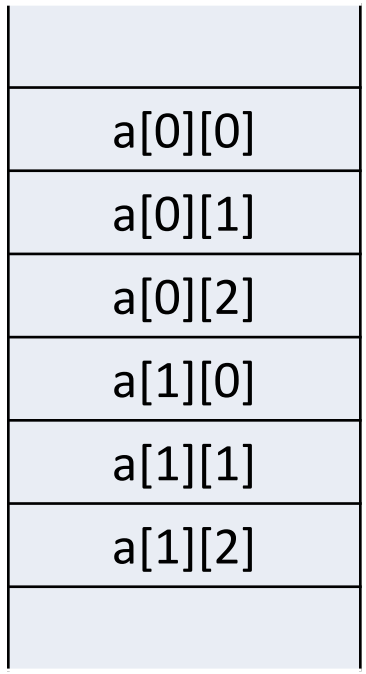
- MT1 score is released on Gradescope
- Regrade deadline: 10pm Central Time on July 10th

Multi-dimensional Arrays Recap

int a [2][3];

	Column 0	Column 1	Column 2
Row 0	a[0][0]	a[0][1]	a[0][2]
Row 1	a[1][0]	a[1][1]	a[1][2]

In memory



* multi-dimensional array is stored in **row-major** order

Exercise: implement a function that transpose an m x n matrix

```
#define ROW 3
#define COL 4
void transpose(int in_matrix[ROW][COL], int out_matrix[COL][ROW]){




}
```

Pointer Array & Pointer to an Array

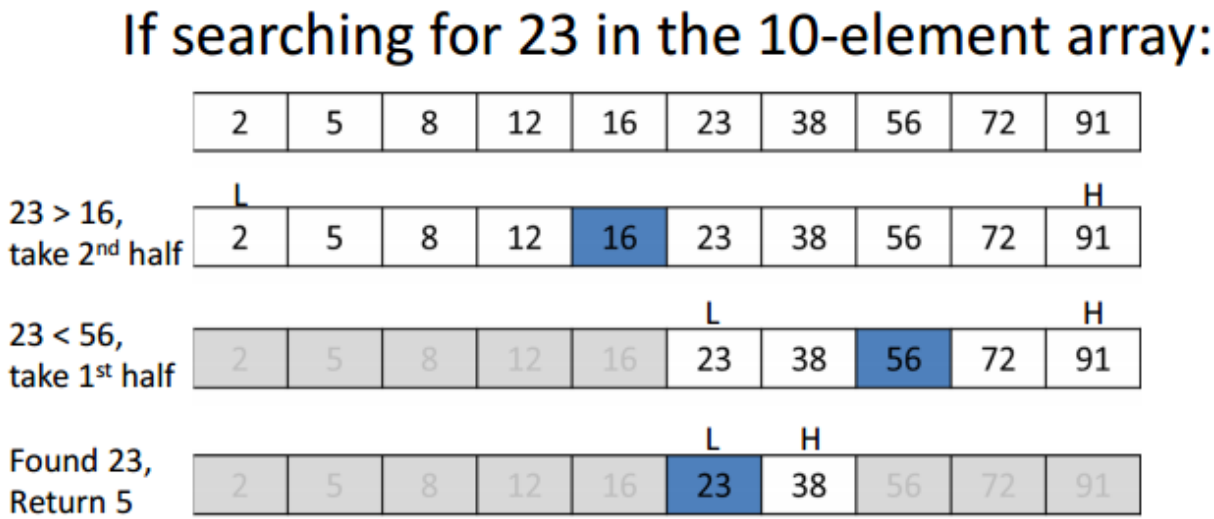
```
int a[4];
int b[5];
int *ptr_array[2];
ptr_array[0] = &a[0]; /* ptr_array[0] = a; */
ptr_array[1] = &b[0]; /* ptr_array[1] = b; */

or

int a[4];
int b[5];
int *ptr_array[2] = {a,b};
```

Search Algorithms

Linear Search: search from the beginning of the array until item is found

Binary Search: (for ***sorted*** array)

1) find the middle of the array and check if it's the search item;
2) search the first half if the search item is smaller than the center item, else search the second half;
3) repeat step 1 & 2 until search item is found.



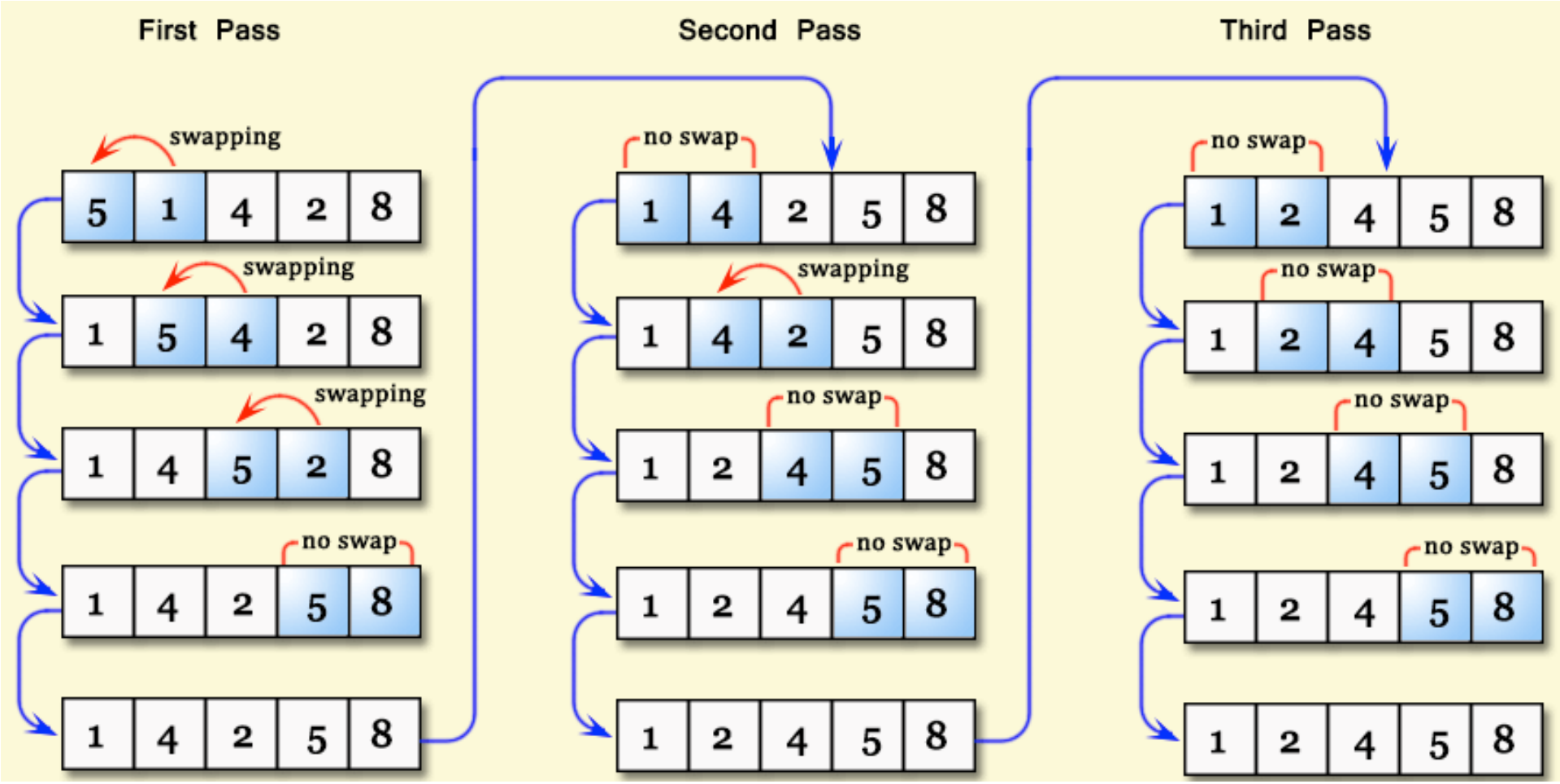
Exercise: implement a function that performs binary search

This function takes two arguments: a pointer to the sorted array and the search item. If the search item is found, the function returns its index in the array. Otherwise, it returns -1.

```
#define SIZE 8
int binary_search(int array[], int item){

}
```

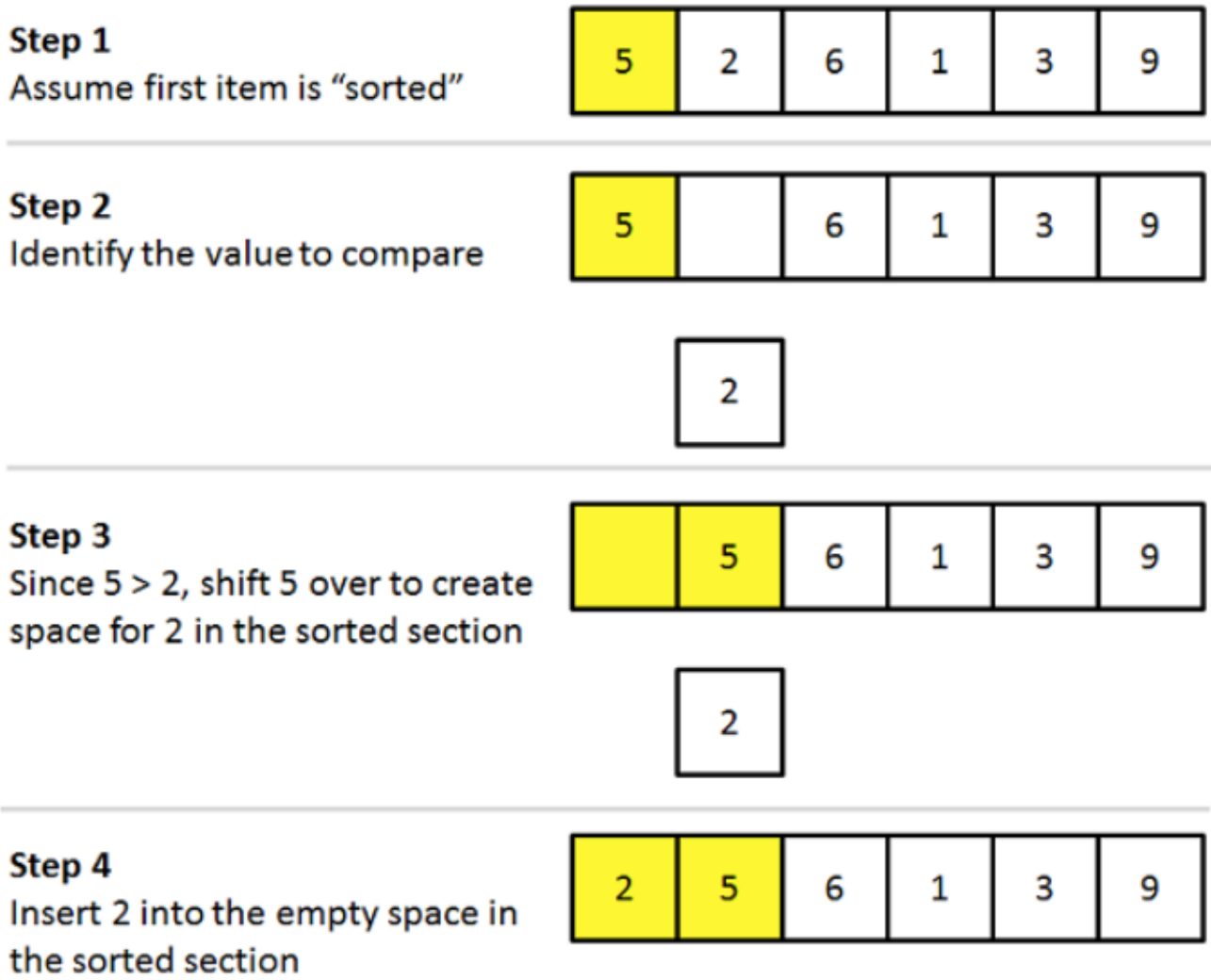
Sorting Algorithms (http://visualgo.net/sorting)

Bubble Sort: 1) compare items next to each other and swap them if needed;
2) repeat this process until the entire array is sorted.



Insertion Sort:

1) remove item from array, insert it at the proper location in the sorted part by shifting other items;
2) repeat this process until the end of array is reach.



Quick Sort: also called divide-and-conquer

1) pick a pivot and partition array into 2 subarrays;

2) then sort subarrays using the same method.