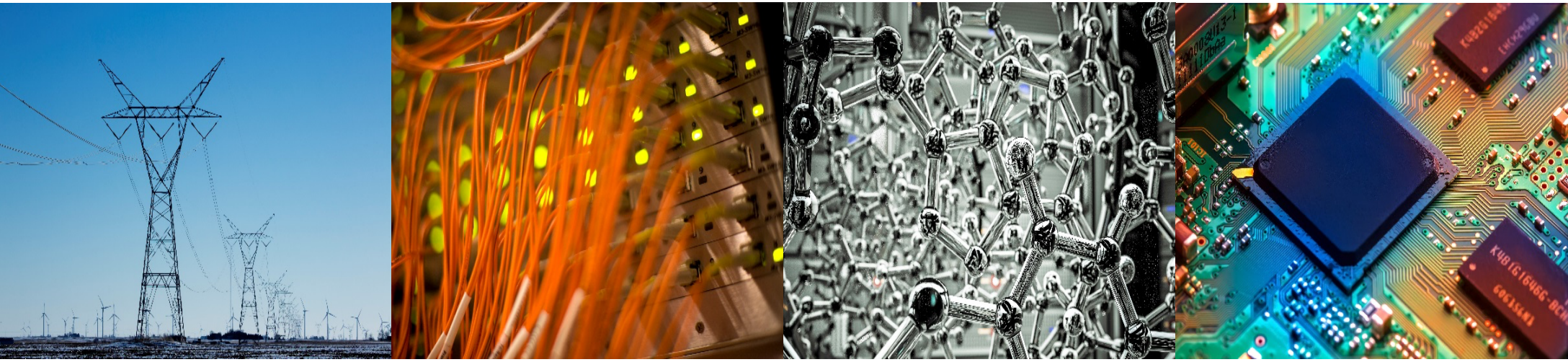
ECE 220 Computer Systems & Programming

Lecture 19 – Sorting Algorithms & Recursion

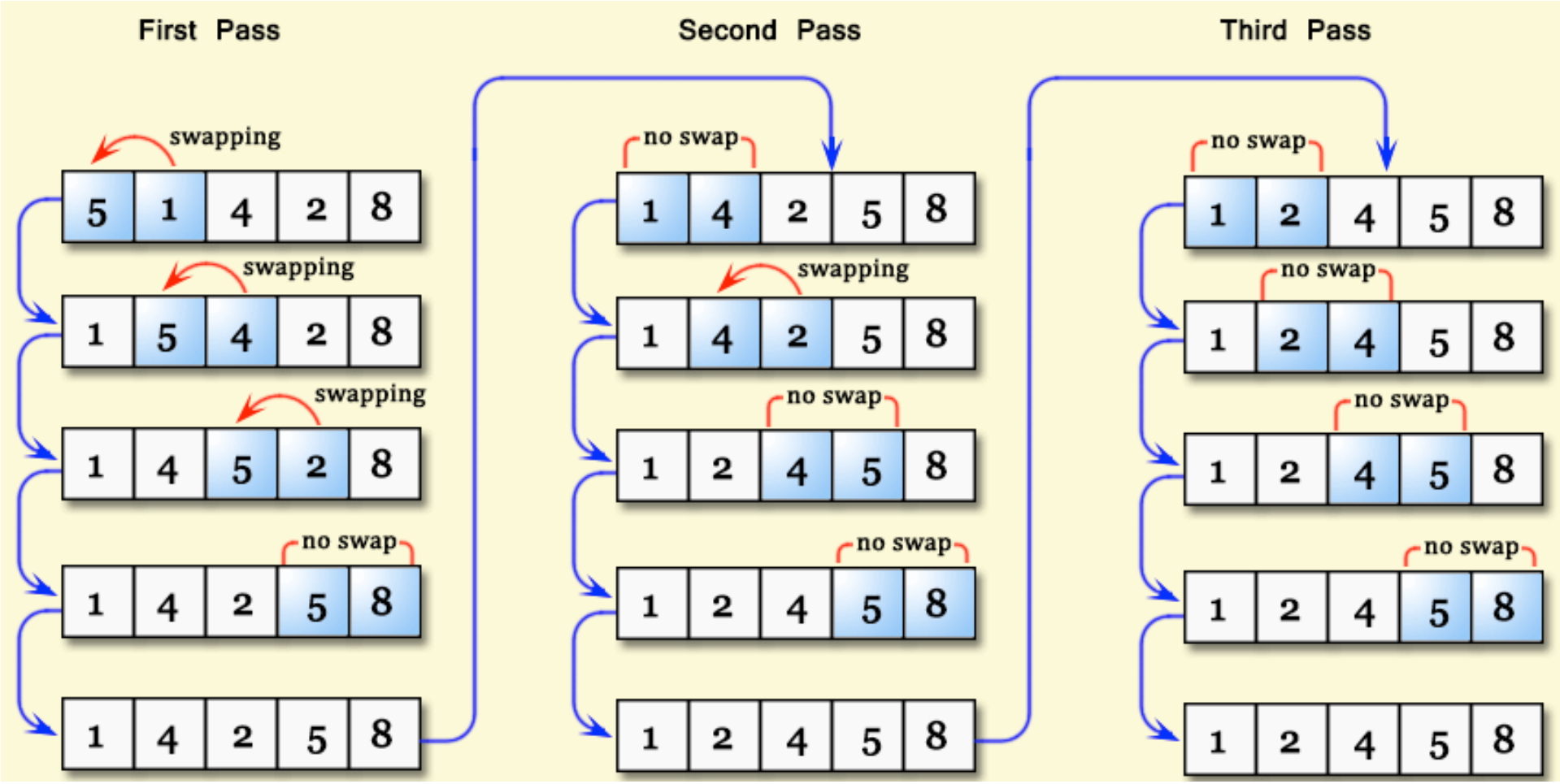
July 13, 2020



ILLINOIS
Electrical & Computer Engineering
GRAINGER COLLEGE OF ENGINEERING

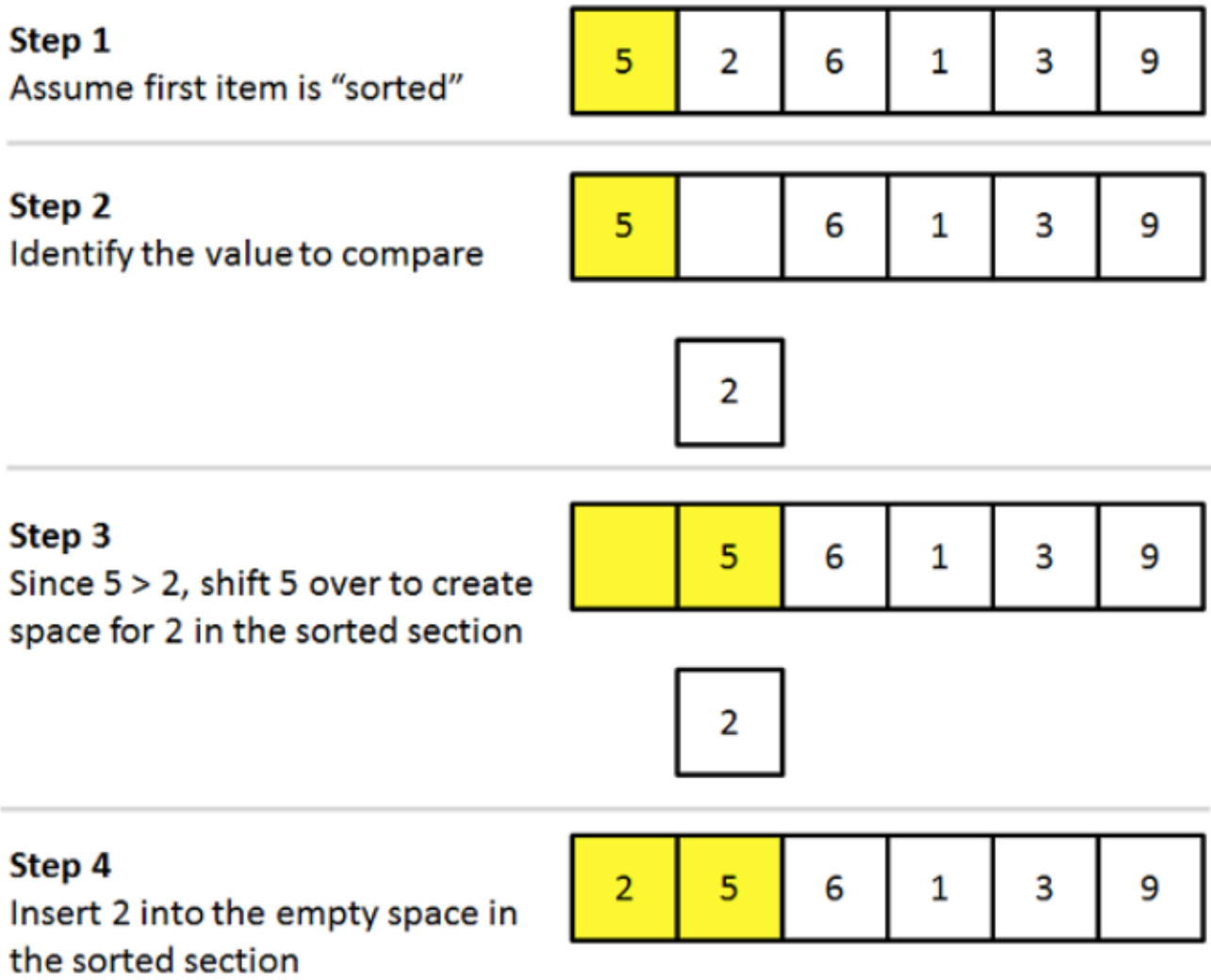
Sorting Algorithms (http://visualgo.net/sorting)

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:

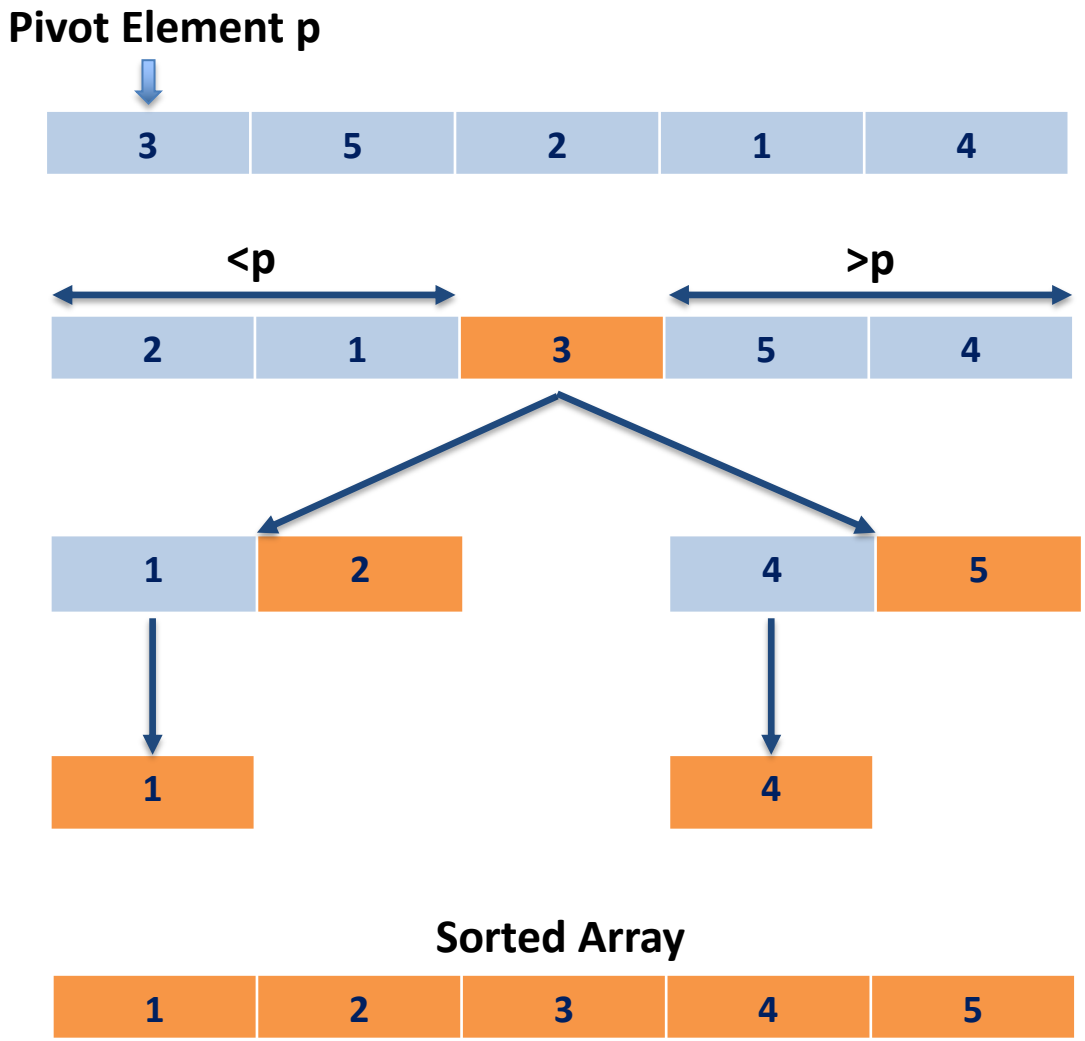
1) remove item from array, insert it at the proper location in the sorted part by shifting other items;
2) repeat this process until the end of array is reach.



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.



Recursion

A **recursive function** is one that solves its task by **calling itself** on smaller pieces of data.

- Similar to recurrence function in mathematics.
- Like iteration -- can be used interchangeably; sometimes recursion results in a simpler solution
- Must have at least 1 **base case** (terminal case) that ends the recursive process

Example: Running sum ($\sum_1^n i$)

Mathematical Definition:

RunningSum(1) = 1

RunningSum(n) =

n + RunningSum(n-1)

Recursive Function:

```
int RunningSum(int n) {
  if (n == 1)
    return 1;
  else
    return n + RunningSum(n-1);
}
```

Recursive Fibonacci

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, …

$$\begin{cases} F_n = F_{n-1} + F_{n-2} \\ F_0 = 0 \\ F_1 = 1 \end{cases}$$

```
int Fibonacci(int n){ /* assume n is non-negative */
```

```
}
```

Recursive Fibonacci with Look-up Table

```
int table[100];
/* assume each element will be initialized to -1 in main */

int fibonacci(int n){ /* assume 0<=n<100 */
   /* if fibonacci(n) has been calculated, return it */



   /* otherwise, perform the calculation, save it to table
     and then return it */




}
```