# 13. Sorting
# ENEE 140

Prof. Tudor Dumitraș
Associate Professor, ECE
University of Maryland, College Park


http://ter.ps/enee140

1

## Today's Lecture

- Where we've been
  - Scalar data types (`int, long, float, double, char`)
  - Vector data types (arrays and strings)
  - Multidimensional arrays
  - Control flow
  - Functions
  - Random number generation
  - File I/O
- Where we're going today
  - Sorting
- Where we're going next
  - Final exam review session

2

2

## A Few Reminders

- Project 2 is due on **Friday, December 5, at midnight**
  - P2 review session on **Wednesday, Dec 3, in the lab**
- Review all the course material for next week
  - Final exams from prior years posted on the class web page
  - Review session **next Tuesday, in class**
- Final exam: **Thursday, Dec 18, 10:30 am–12:30 pm, in CSI 1115**

3

3

## Incremental Maintenance of Aggregates

- Sometimes, you must compute values that summarize multiple numbers (aggregates)
  - Examples: count, maximum, average
  - You can compute many aggregates incrementally, by updating a variable at each iteration of a loop

```
int a, count = 0, max = INT_MIN;
while (scanf("%d", &a) > 0) {
  count++;

}
```

must initialize the aggregates

increment count

update max

- **How should you initialize the aggregate?**

4

4

## Swapping Two Variables

- How to swap the values of two variables `a` and `b`?
  - `a` must take the old value of `b`
  - `b` must take the old value of `a`

```
int a=1, b=2;
a=b;                    a is 2
b=a;                    b is 2 incorrect!

int a=1, b=2, tmp;
tmp = a;                tmp is 1
a = b;                  a is 2
b = tmp;                b is 1
```

- **How would you swap 2 rows of a matrix? Or 2 matrices?**

5

5

## Sorting

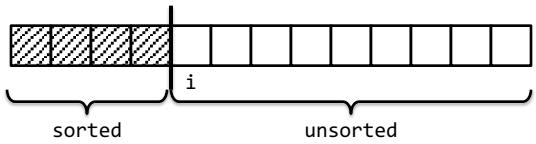
- Rearrange the elements of array `a[N]` so that they are ordered
  - Ascending order: $a[0] \le a[1] \le a[2] \le \ldots \le a[N-1]$
  - Descending order: $a[0] \ge a[1] \ge a[2] \ge \ldots \ge a[N-1]$
- There are many sorting algorithms
  - Some use techniques not covered in ENEE 140 (e.g. recursion)
- We focus on a few simple algorithms
  - Selection sort
  - Insertion sort

6

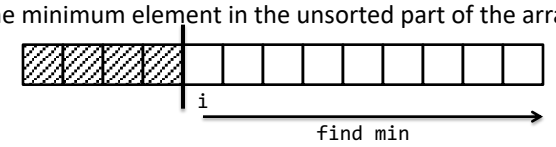
6

## Selection Sort

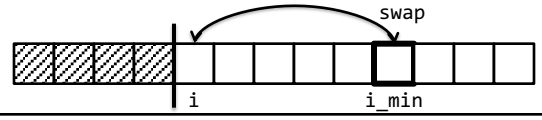
- Key idea: gradually build up the sorted array
- At each iteration:
  - The beginning part of the array contains the lowest elements, in sorted order




  - Find the minimum element in the unsorted part of the array




  - Add it to the end of the sorted part




7

7

## Demonstration

8

8

## What Programming Principles Did We Learn in ENEE140?

- Learning objective: write high-quality code
  - **Correctness**: the code should do what it's supposed to do (and nothing else!)
  - **Maintainability**: other programmers should find the code easy to read and to modify
- Other code-quality attributes that we did not emphasize
  - Efficiency
  - Robustness
  - Security

9

9

## What Programming Concepts Did We Learn in ENEE140?

- C is a low-level language
  - No operations for manipulating composite types (e.g. strings, lists, arrays), no memory management, no input/output facilities
    - The standard library provides some of these facilities
  - A small language
    - Can be learned quickly
- Concepts **covered** in ENEE 140:
  - Variables
  - Integer & floating-point division
  - …
- Concepts **not covered** in 140:
  - Recursion
  - Pointers

10

10

## What is Programming (Revisited)

- Becoming fluent in the language that computers understand
  - Humans are better than computers at doing certain things
  - Computers are better than humans at other things
  - **If you can program, you can do both!**
- Programming stimulates a **way of thinking**
  - Helps you acquire aptitudes and skills applicable in many situations
  - Examples: top-down problem solving, thinking at multiple levels of abstraction, thinking of worst-case scenarios to avoid failures
- Programming is a **creative** process
  - Within the bounds of what computers and programming languages can do

11

11

## Next Steps

- Assignments for this week
  - No weekly challenge
  - Homework: `lab12.pdf` (on http://ter.ps/enee140), due on Friday at 11:59 pm
  - **Project 2:** complete implementation due on Friday at 11:59 pm
    - P2 review on Wednesday, in the lab
- Next week
  - Review all the material; **final-exam review session in class**

12