Sorting

ENEE 140

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

http://ter.ps/enee140

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
  - P3 review
- Where we're going next
  - Final exam review

2

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
```

3

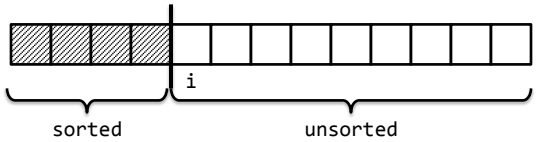
Sorting

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

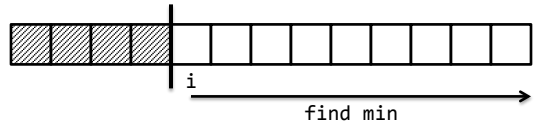
4

Selection Sort

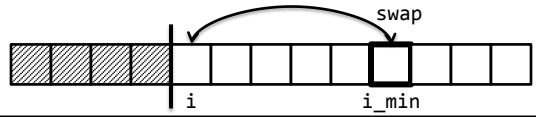
- Key idea: gradually build up the sorted array
- At each iteration:
  - The beginning part of the array contains the lowest elements, in sorted order



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



  - Add it to the end of the sorted part



5

What is Programming? (revisited)

- Becoming fluent in the language that computers understand
  - Define branches, loops, encapsulate tasks in functions and modules
  - Work with computer `int`s and `float`s, understand data types
  - Input / output
  - Your programs must explicitly handle all inputs and corner cases
  - Can use a debugger to figure out why your program isn't working
- Programming stimulates a way of thinking
  - Top-down problem solving: break a program into multiple modules
  - Use abstraction: loop invariants, use formatted I/O (w/o knowing low-level I/O)
  - Think of worst-case scenarios: handle incorrect inputs to avoid failures
- Programming is a creative process
  - Come up with algorithms to solve problems

6

Course Evaluations

- Do not forget to submit your course evaluation for ENEE 140!
  - Deadline: Wednesday, May 11 (**before final exam week**)
  - Let us know about how we could improve how this course is taught
    - Challenges you've encountered, so that we can improve those areas
    - What worked well, so that we don't change it
- https://www.CourseEvalUM.umd.edu

7

Review of Lecture

- What did we learn?
  - Swapping two variables
  - Selection sort
- Next lecture
  - Review session for the final exam
- Reminder: Project 3 due on Monday
- Assignments for this week
  - Review all the material for the final exam
  - No weekly challenge
  - Homework: `lab13.pdf` (on http://ter.ps/enee140), due on Friday at 11:59 pm