CSE 3101E Design and Analysis of Algorithms

Prof. J. Elder
Textbooks

• Required Text:
    • Available in the York University Bookstore
    • Also available from [www.amazon.ca](http://www.amazon.ca) for $68.01

• Optional Text:
Assignments

• You will learn best if you try to tackle each problem by yourself initially.

• You are encouraged to discuss the problems and work in groups to overcome roadblocks.

• You are encouraged to team up with a partner and write up a single assignment report (maximum 2 per group).

• Make the reports as concise and organized as possible. Marks may be taken off for excess verbosity or lack of clarity.

• Late assignments are not excepted (except for medical emergencies – please see syllabus).
Teaching Assistant

- Eduardo Corral Soto

- Eduardo will be marking all assignments.

- Please direct any questions regarding the marking of assignments to Eduardo.

- If you believe there was an error in marking your assignment, please bring it to Eduardo’s attention promptly (within a week).

- Office hour: Mon 9:30-10:30 (Location TBA)
Please ask questions!

Help me know what people are not understanding!
On the slides

• These slides:
  – are available from the website
    www.elderlab.yorku.ca/~elder/teaching/cse3101
  – may change up to the last minute as I polish the lecture.
  – Include slides originally created by J. Edmonds: Thanks Jeff!
Lectures

- I will teach the first 18 lectures
- Jeff Edmonds will teach the last 3 lectures on dynamic programming
Lecture 1. What is this course about?
Course Content

- A list of algorithms.
  - Learn their code.
  - Trace them until you are convinced that they work.
  - Implement them.
  - Worry about details.

```java
class InsertionSortAlgorithm extends SortAlgorithm {
    void sort(int a[]) throws Exception {
        for (int i = 1; i < a.length; i++) {
            int j = i;
            int B = a[i];
            while ((j > 0) && (a[j-1] > B)) {
                a[j] = a[j-1];
                j--; }
            a[j] = B;
        }
    }
}
```
Innovation

Abraham Lincoln

Thomas Edison

Steve Wozniak and Steve Jobs
The future belongs to the computer scientist/engineer who has

- **Knowledge**: An up to date grasp of fundamental problems and solutions
- **Ability**: Principles and techniques that can be adapted to solve new problems
Course Content

- A survey of algorithmic design techniques.
- Abstract thinking.
- How to develop new algorithms for any problem that may arise.
A survey of fundamental ideas and algorithmic design techniques

For example . . .
Mathematical Tools

Summations
\[ \sum_{i=1} f(i). \]

Classifying Functions
\[ f(i) = n^{\Theta(n)} \]

Recurrence Relations
\[ T(n) = a \cdot T(n/b) + f(n) \]

Time Complexity
\[ t(n) = \Theta(n^2) \]
Iterative Algorithms
Loop Invariants

<preCond>
codeA
loop
    <loop-invariant>
    exit when <exit Cond>
    codeB
    codeC
<postCond>

Code  One step at a time  Relay Race

COSC 3101, PROF. J. ELDER
Recursive Algorithms
Greedy Algorithms

Example: Making Change
Dynamic Programming
Useful Learning Techniques
Read Ahead

You are expected to read the lecture notes before the lecture.

This will facilitate more productive discussion during class.
Explaining

- We are going to test you on your ability to explain the material.
- One good way to study is to explain the material over and over again to yourself or to each other.
Be Creative

• Ask questions.

• Why is it done this way and not that way?
Guesses and Counter Examples

• Guess at potential algorithms for solving a problem.
• Look for input instances for which your algorithm gives the wrong answer.
• Treat it as a game between these two players.
Refinement:
The best solution comes from a process of repeatedly refining and inventing alternative solutions
End