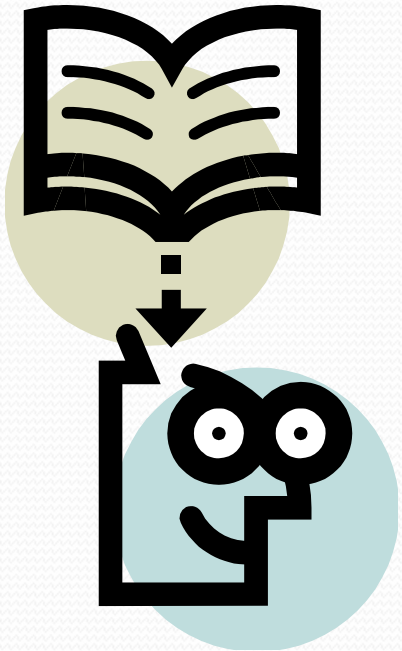
Algorithm Analysis: Introduction

Prepared by Hyrum D. Carroll
(based on slides from Suk Jin Lee)

Algorithm

- What is an algorithm?



Algorithm in General

- An **algorithm** is a finite set of well-defined instructions for accomplishing some task which, given an initial state, will terminate in a defined end-state.
- To develop any algorithm, it is necessary to know, how the corresponding problem can be solved.

Algorithm in Computer Science

- An **algorithm** is any well-defined computational procedure that takes some value, or set of values, as input and produces some value, or set of values, as output. An algorithm is a sequence of computational steps (instructions) that transform the input into the output.

Algorithm in Computer Science

- An **algorithm** is a tool for solving a well-specified computational problem.
- The statement of the problem specifies in general terms the desired input/output relationship.
- The algorithm describes a specific computational procedure for achieving that relationship.

Solving Computational Problem

- Problem setting
- Design of an algorithm
- Analysis of the algorithm (proof of its correctness, analysis of its efficiency, comparison to other algorithms solving the same problem)
- Software implementation of the algorithm
- Solving the problem

So, what is this course about?

Solving Interesting Problems

1. Design of algorithms (Methods and tools).

2. Correctness of algorithms
   - Buggy algorithms are worthless!

3. Efficiency of algorithms
   - Complexity, simplicity, running time, space needed, …

4. Examples of efficient algorithm design
   - Graph algorithms, shortest path algorithms, …

Properties of Algorithms (1)

- Time efficiency
  - As a function of its input size, how long does it take?
- Space efficiency
  - As a function of its input size, how much additional space does it use?

Most interested in asymptotic efficiency:

worst-case & average-case

Properties of Algorithms (2)

- Simplicity
  - Informal notion
  - Easier to code correctly
  - Lower constant factors in efficiency (probably)
  - Easier to explain
  - Easier to prove the correctness
  - More mathematically “elegant”

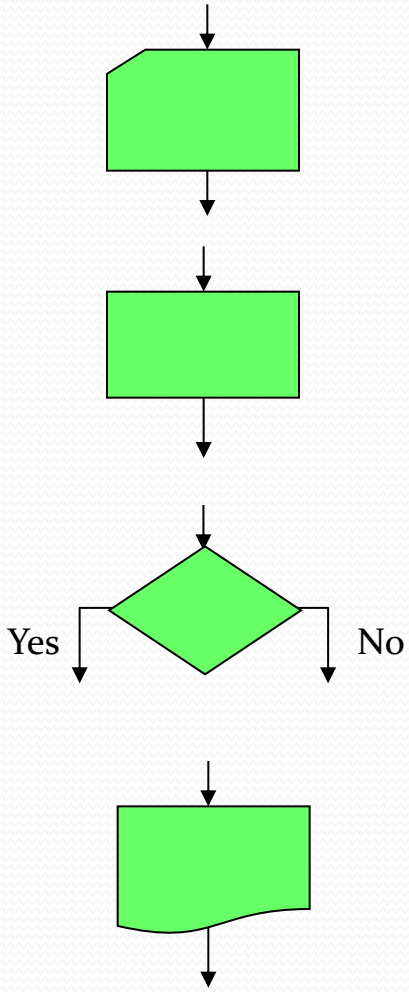
Basic Types of Algorithms

- Linear algorithm consists of a sequence of unconditional straightforward steps.
- Loop is a group of steps that are repeated until some condition will not be satisfied.
- Nested Loop is a loop containing another loop (loops).
- Branching algorithm consists of a number of subsequences that can be taken depending on some condition (conditions).

Modern computer technologies

- Hardware (including pipelining and superscalar architecture)
- Graphical User Interfaces (GUI)
- Object-oriented programming
- Networking (local-area and wide-area)
- **ALGORITHMS**

Algorithms Representation: Flowchart



- Data input
- Operations (Arithmetic) and Assignments
- Logical block (Branching)
- Data output

Algorithms Representation: Pseudocode

←	Assignment	$i \leftarrow j + 1$, $i \leftarrow i - 1$
for…do	**for** loop	**for** $i \leftarrow 1$ to n **do** **loop body** i is a loop counter (loop variable)
while…do	**while** loop	**while** (logical condition) **do** **loop body**
[1..*n*]	a range within an array	*a*[1..*n*]
a[*i*]	the i^{th} element of the array	
Length[*a*]	length of array *a*	
//	the reminder of the line is a comment	
if then else	conditional branching	if <logical condition> then <statement(s)> else <statement(s)>