CST 311
Algorithm Analysis & Design

Al Lake
Oregon Institute of Technology

Chapter 1

Introductory

Course Outline

- This class is an introduction to design, analysis and implementation of algorithms.
- The course topics will include the following:
  - mathematical foundations for algorithm analysis,
  - a review of elementary data structures, sorting, searching, and graph algorithms,
  - string matching,
  - algorithmic design models and
  - an introduction to NP-completeness.

Algorithm

- **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 that transform the input into the output.**
- **An algorithm describes a specific computational procedure for achieving the input/output relationship.**

Correct

- **An algorithm is said to be correct if, for every input instance, it halts with the correct output.**
- **A correct algorithm solves the given computational problem.**
- **An incorrect algorithm might not halt at all on some input instances, or it might halt with an answer other than the desired one.**

Data Structure

- **A data structure is a way to store and organize data in order to facilitate access and modifications.**
- **No single data structure works well for all purposes, and so it is important to know the strengths and limitations of several of them.**

Insertion-Sort

INSERTION-SORT(A)

```
for j ← 2 to length[A]
    do key ← A[j]
       ▷ Insert A[j] into the sorted sequence A[1 . . j − 1].
       i ← j − 1
       while i > 0 and A[i] > key
           do A[i + 1] ← A[i]
              i ← i − 1
       A[i + 1] ← key
```

Loop invariants and the correctness of insertion sort

Merge Sort

MERGE-SORT(A, p, r)

if $p < r$

  then $q \leftarrow \lfloor (p + r)/2 \rfloor$

    MERGE-SORT(A, p, q)

    MERGE-SORT$(A, q + 1, r)$

    MERGE(A, p, q, r)

Efficiency

- Insertion sort takes approximately c_1n^2 to sort n items
- Merge sort takes approximately $c_2n\lg n$ to sort n items

- Note: Insertion sort normally has a smaller constant factor than merge sort, so $c_1 < c_2$.

Comparing Insertion-sort & Merge Sort

c	n	$c_1 n^2$	c_2 nlgn
2	10	200	20
2	100	20,000	400
2	1000	2,000,000	6,000
2	10000	200,000,000	80,000
2	100000	20,000,000,000	1,000,000
20	10	2,000	200
20	100	200,000	4,000
20	1000	20,000,000	60,000
20	10000	2,000,000,000	800,000
20	100000	200,000,000,000	10,000,000
1000	10	100,000	10,000
1000	100	10,000,000	200,000
1000	1000	1,000,000,000	3,000,000
1000	10000	100,000,000,000	40,000,000
1000	100000	10,000,000,000,000	500,000,000