**Razinul Karim**

**18CSE-002**

**INTRODUCTION TO ALGORITHM**

**Chapter 1**

**Algorithms:**

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.

**Problems solved by algorithms:**

1. The Human Genome Project has made great progress
2. The Internet enables people all around the world to quickly access and retrieve large amounts of information
3. Manufacturing and other commercial enterprises often need to allocate scarce resources in the most beneﬁcial way.
4. They have many candidate solutions

**Data structures:**

A data structure is a way to store and organize data in order to facilitate access and modiﬁcations.

**Technique:**

Different chapters address different aspects of algorithmic problem solving. Some chapters address speciﬁc problems, such as ﬁnding medians and order statistics, computing minimum spanning trees, and determining a maximum ﬂow.

**Hard problems:**

The hard the problem is, more efficient algorithm is needed.

**Parallelism:**

This model has advantages from a theoretical standpoint, and it forms the basis of several successful computer programs, including a championship chess program.

**Algorithms as a technology:**

If computers were inﬁnitely fast, any correct method for solving a problem would do. You would probably want your implementation to be within the bounds of good software engineering practice (for example, your implementation should be well designed and documented), but you would most often use whichever method was the easiest to implement.

**Efﬁciency:**

For example Insertion sort’s running time as c1n.n and merge sort’s running time as c2n.logn.

**Algorithms and other technologies:**

1. Advanced computer architectures and fabrication technologies
2. Easy-to-use, intuitive
3. Graphical user interfaces (guis)
4. Object-oriented systems
5. Integrated web technologies
6. Fast networking, both wired and wireless.

**Chapter 2**

**Insertion sort:**

Insertion-sort(A)

1 for j = 2 to A.length

2 key = A[j]

3 // Insert A[j] into the sorted sequence A[..j - 1].

4 i = j - 1

5 while i>0 and A[i] > key

6 A[i+1] = A[i]

7 i = i -1

8 A[i +1]= key

**Initialization:**

It is true prior to the ﬁrst iteration of the loop.

**Maintenance:**

If itis true before an iteration of the loop, itremains true before the next iteration.

**Termination:** When the loop terminates, the invariant gives us a useful property that helps show that the algorithm is correct.

**Analyzing algorithms:**

Analyzing an algorithm has come to mean predicting the resources that the algorithm requires. Occasionally, resources such as memory, communication bandwidth, or computer hardware are of primary concern, but most often it is computational time that we want to measure

**The divide-and-conquer approach:**

Divide: Divide the n-element sequence to be sorted into two subsequences of n=2 elements each. Conquer: Sort the two subsequences recursively using merge sort. Combine: Merge the two sorted subsequences to produce the sorted answer.

**Chapter 3**

**Asymptotic notation:**

Asymptotic Notations are languages that allow us to analyze an algorithm's running time by identifying its behavior as the input size for the algorithm increases.

**Types of Data Structure Asymptotic Notation:**

1. Big-O Notation (Ο) – Big O notation specifically describes worst case scenario.  
2. Omega Notation (Ω) – Omega(Ω) notation specifically describes best case scenario.  
3. Theta Notation (θ) – This notation represents the average complexity of an algorithm.

**Monotonic:**

In calculus, a function. defined on a subset of the real numbers with real values is called monotonic if and only if it is either entirely non-increasing, or entirely non-decreasing.

**Polynomials:**

an expression of more than two algebraic terms, especially the sum of several terms that contain different powers of the same variable(s).

**Exponentials:**

In mathematics, an exponential function is a function of the form. where b is a positive real number, and in which the argument x occurs as an exponent.

**Logarithm  :**

n mathematics, the logarithm is the inverse function to exponentiation.

**Factorials:**

the product of a series of factors in an arithmetical progression