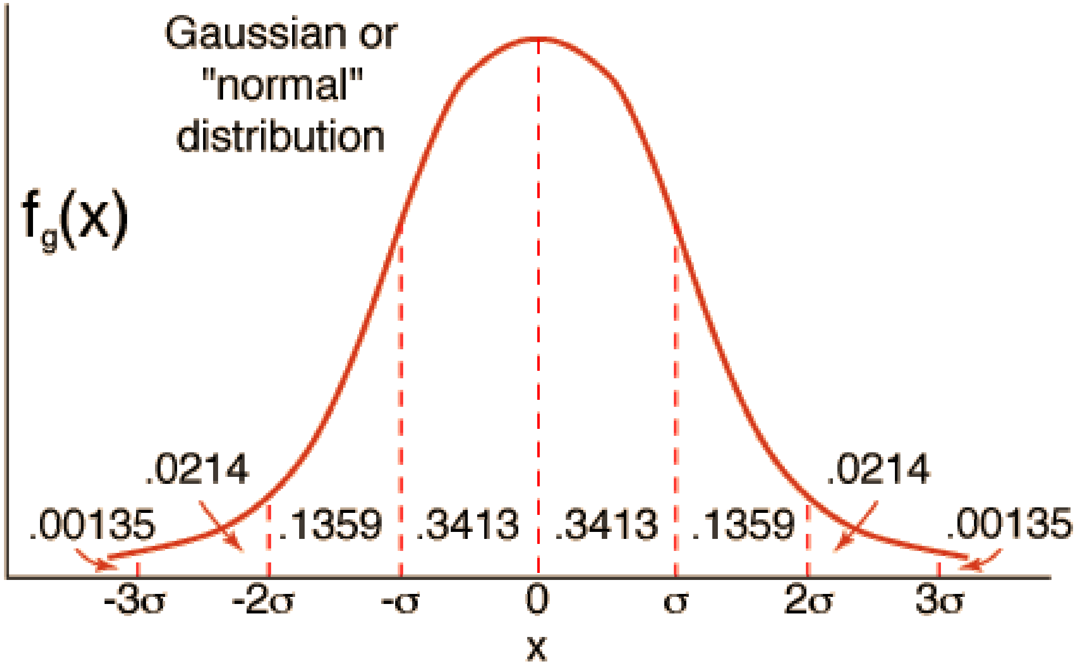
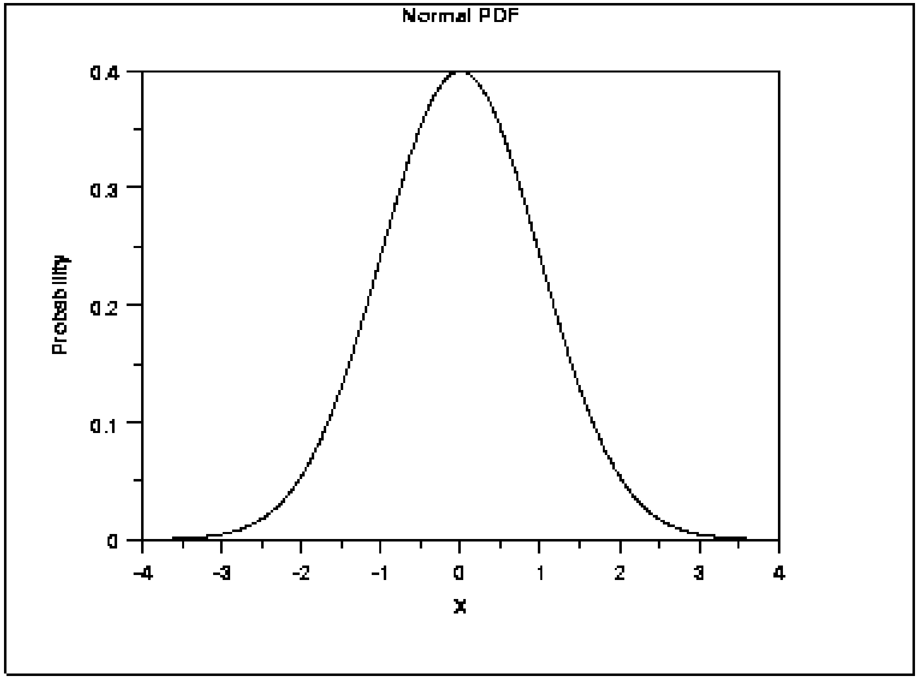
Why is Statistics important in Bioinformatics?

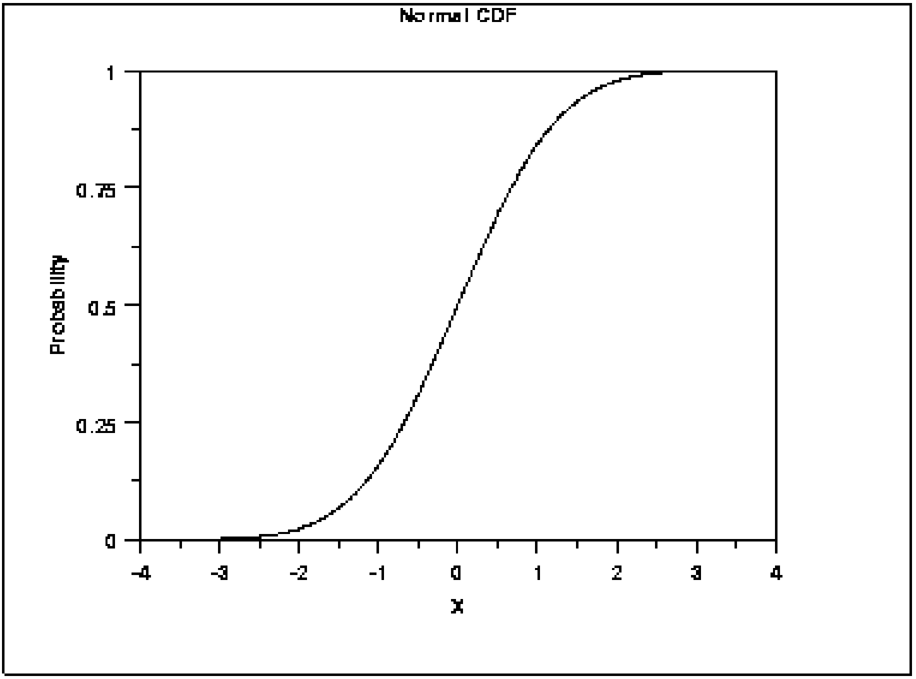
- Random processes are inherent in evolution and in sampling (data collection).
- Errors are often unavoidable in the data collection process.
- Statistics helps in studying *trends, interpolations, extrapolations, categorizations, classifications, inferences, models,* ...

Normal/Gaussian Distribution

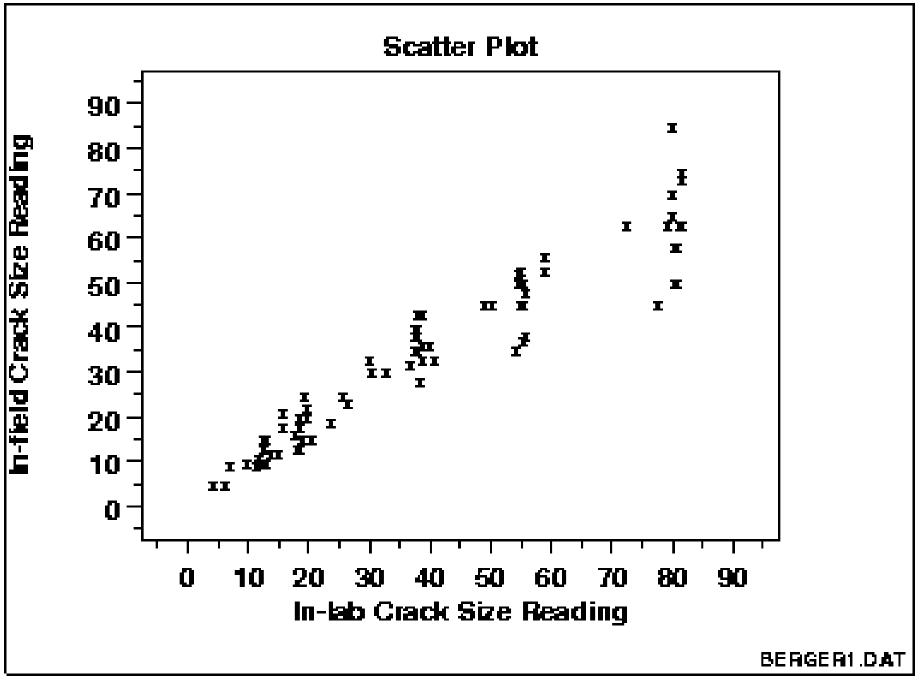


Density & Cumulative Distribution Functions

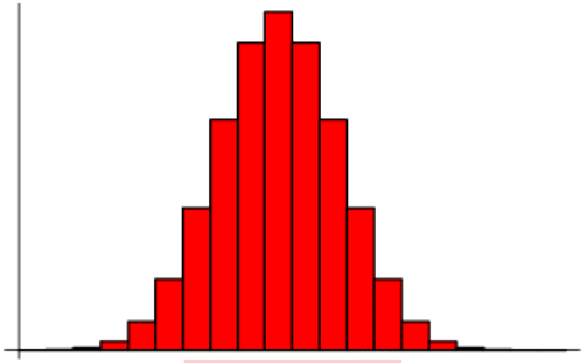




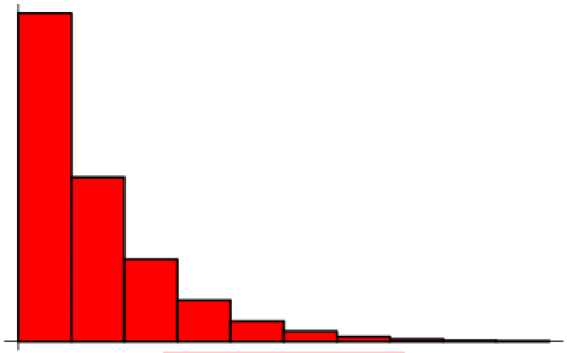
Graphical Techniques: Scatter Plot



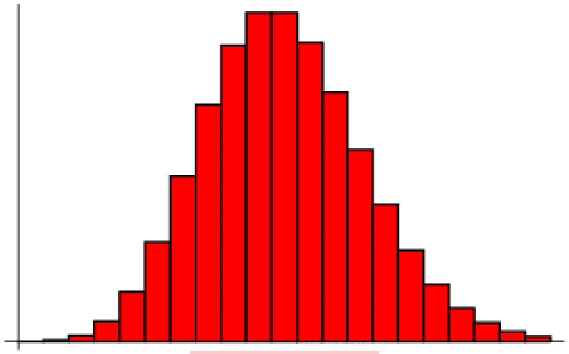
Common Discrete Distributions



Binomial



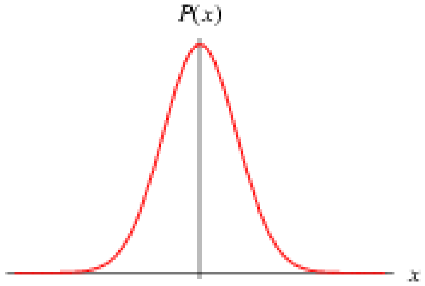
Geometric

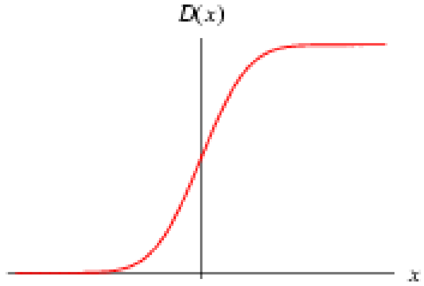


Poisson

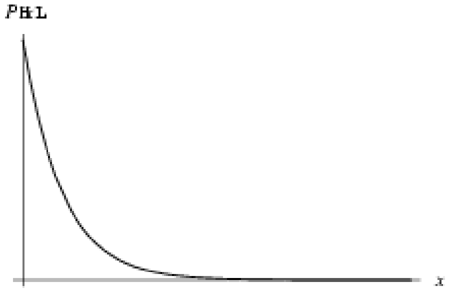
Common Continuous Distributions

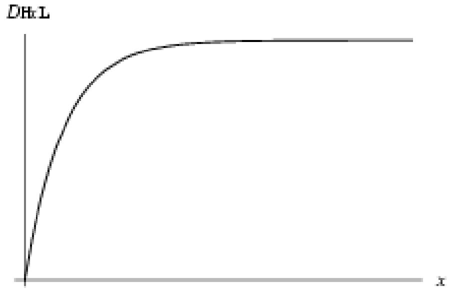
Normal



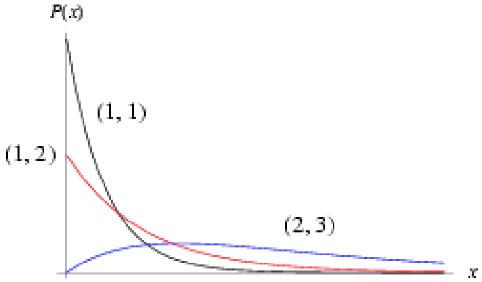


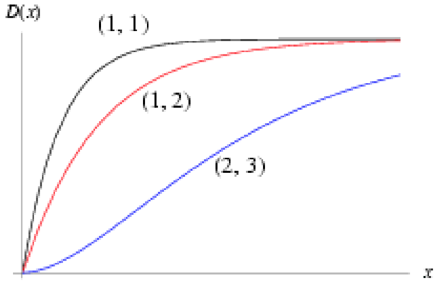
Exponential





Gamma





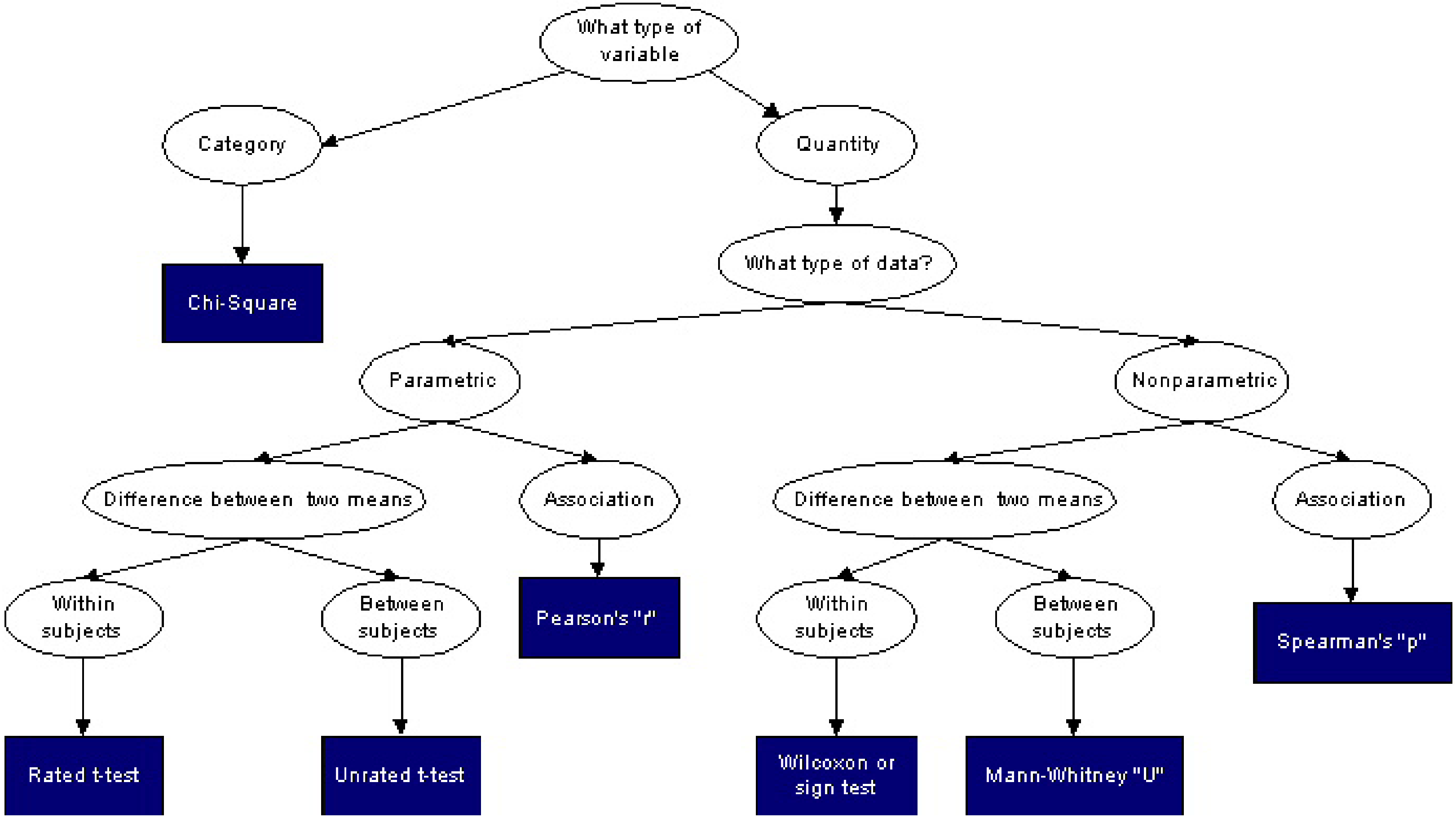
Monte Carlo methods

- Numerical statistical simulation methods that utilize sequences of random numbers to perform the simulation.
- The primary components of a Monte Carlo simulation method include the following:
  - *Probability distribution functions (pdf's)* --- the system described by a set of pdf's.
  - *Random number generator* --- uniformly distributed on the unit interval.
  - *Sampling rule* --- a prescription for sampling from the specified pdf's.
  - *Scoring (or tallying)* --- the outcomes must be accumulated into overall tallies or scores for the quantities of interest.
  - *Error estimation* --- an estimate of the statistical error (variance) as a function of the number of trials and other quantities must be determined.
  - *Variance reduction techniques* --- methods for reducing the variance in the estimated solution to reduce the computational time for Monte Carlo simulation.

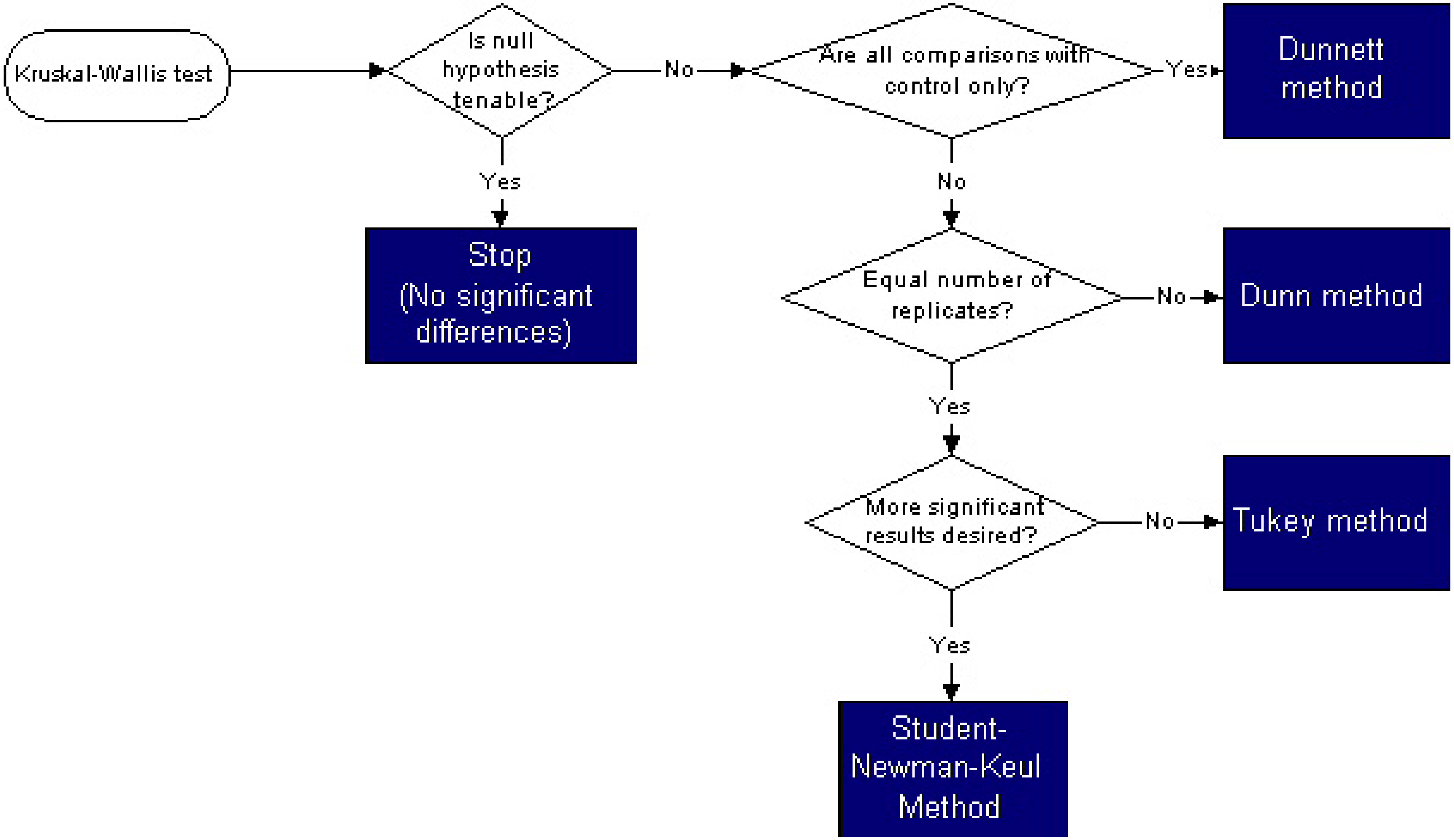
Parametric & Non-parametric equivalents

Type of test	Parametric test	Nonparametric test
2-sample	t-test	Mann-Whitney U-test
Paired sample	Paired t-test	Wilcoxon
Distribution	Chi-square	Kolmogorov-Smirnov
>2 samples	1-way ANOVA	Kruskal-Wallis
Correlation	Pearson’s correlation	Spearman’s correlation
Crossed comparisons	Factorial ANOVA	Friedman’s Quade
Multiple comparisons	Tukey, SNK, Dunnett’s, Scheffe’s	Nonparametric version of its parametric equivalents

Selection of Statistical Tests



Selection of Multiple Comparison Tests



Kruskal-Wallis test
Is null hypothesis tenable?
No
Are all comparisons with control only?
Yes
Dunnett method
Yes
Stop (No significant differences)
No
Equal number of replicates?
No
Dunn method
Yes
More significant results desired?
No
Tukey method
Yes
Student-Newman-Keul Method

Computer Science Fundamentals

- Specify an input-output description of the problem.
- Design a conceptual algorithm and analyze it.
- Design data structures to refine the algorithm.
- Write the program in parts and test the parts separately.

Evolution of Data Structures

- Complex problems often required complex data structures.
- Simple data types: Lists. Applications of lists include: students roster, voters list, grocery list, list of transactions,
- Array implementation of a list. Advantage – random access.
- Need for list “operations” arose – “Static” vs. “dynamic” lists. “Storing” items in a list vs. “Maintaining” items in a list.
- Lot of research on “Sorting” and “Searching” resulted.
- “Inserting” in a specified location in a list caused the following evolution: Array implementation → Linked list implementation.
- Other linear structures e.g., stacks, queues, etc.

Evolution of Data Structures (Cont’d)

- Trees made hierarchical organization of data easy to handle. Applications of trees: administrative hierarchy in a business set up, storing an arithmetic expression, organization of the functions calls of a recursive program, etc.
- Search trees (e.g., BST) were designed to make search and retrieval efficient in trees. A BST may not allow fast search/retrieval, if it is very unbalanced, since the time complexities of the operations depended on the height of the tree. Hence the study of “balanced” trees and “nearly balanced” trees. Examples: AVL trees, 2-3 trees, 2-3-4 trees, RB trees, Skip lists, etc.
- Graphs generalize trees; model more general networks.
- Abstract data types. Advantages include: Encapsulation of data and operations, hiding of unnecessary details, localization and debugging of errors, ease of use since interface is clearly specified, ease of program development, etc.