Data Structures*

*Note: Sub-titles are not captured in Xplore and should not be used

1st Sarvesh Anant Kulkarni
Computer Engineering
COEP Technological University
Pune India
email : kulkarnisa22.comp@coep.ac.in

3rd Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

4th Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

***Abstract*—This document is a model and instructions for LaTeX. This and the IEEEtran.cls file define the components of your paper [title, text, heads, etc.]. *CRITICAL: Do Not Use Symbols, Special Characters, Footnotes, or Math in Paper Title or Abstract.**

***Index Terms*—component, formatting, style, styling, insert**

I. Introduction

This report is for those enthusiastics who are eager to explore data structures.

II. Choosing a Data Structure/Algorithm

A. Maintaining the Integrity of the Specifications

a) The choice of data structure or algorithm to cover is largely open ended. But you must send a short paragraph explaining your plan before Thanksgiving break. This should include the topic you plan to cover and what you aim to discuss in your paper. :

b) You can use the book to find example data structures that we did not cover, or explore resources online. While Wikipedia is not a primary source, it is a reliable repository of data structures that can be easily explored and there are usually good examples.:

III. Writeup Requirements

Your writeup should be well-structured and follow scientific writing principles. You can structure as you see fit, but at a minimum, your paper should include:

1) An introduction - this can include a history and a preview of the central features of the data structure/algorithm
2) Motivation - what is the problem that needs to be solved? It is recommended that you come up with real-world application and describe it here.
3) Background/related work - what does this relate to from class? What are the conceptual differences? Are there other algorithms/data structures that generally compete with your choice?
4) Interface - what are the major operations of the data structure? This may not apply for an algorithm, but you discuss assumptions for the structure of the data.
5) Illustration - walk through an example (e.g., for BFS, going from Parrish to the Ville). You should have some sort of diagram that illustrates how the data structure/algorithm works.
6) Analysis - provide pseudocode for a few key operations and analyze their run time. This does not need to be exhaustive, and how many operations you cover depends on the complexity of the problem. For example, inserting/removing from linked lists was fairly simple so it would be good to cover most of the methods. Removing/inserting into an AVL tree is quite complex, so be sure to limit your discussion to one or two key operations. You should then summarize the other operations.

A. Presentation

You will present your findings in a short presentation (maximum 12 minutes) during lecture on Tuesday, December 10. You will be divided into groups of 10 students with one discussion leader (Prof. Soni, Prof. Brody, a ninja, or Frances). Each student in your group as well as the discussion leader will evaluate your presentation on the criteria defined.

B. Logistics and Tips

- You are permitted to use powerpoint/slides if you like. This has been rare in the past as making slides is time consuming. But if you choose to use any visualization requiring a projector, you must let me know ASAP. Also, you should show up a few minutes early to set up your laptop.
- Some students in the past have provided a worksheet to help in their presentation, or to supplement their presentation with more details that had tobe omitted due to time.
- Be sure to practice! 12 minutes goes by much faster than you'll expect
- Do not try to cover every aspect of your topic. Specifically, the presentation does not need to cover all the aspects of your paper.
- Moreover, a good strategy is to concentrate on one subproblem and give a high-level intuition of other topics.

Identify applicable funding agency here. If none, delete this.

C. Basics of Data strucutre

A data structure is a storage that is used to store and organize data. It is a way of arranging data on a computer so that it can be accessed and updated efficiently.

- Linear data structure: Data structure in which data elements are arranged sequentially or linearly, where each element is attached to its previous and next adjacent elements, is called a linear data structure.
  - Static data structure: Static data structure has a fixed memory size. It is easier to access the elements in a static data structure. An example of this data structure is an array.
  - Dynamic data structure: In dynamic data structure, the size is not fixed. It can be randomly updated during the runtime which may be considered efficient concerning the memory (space) complexity of the code. Examples of this data structure are queue, stack, etc.

  Non-linear data structure: Data structures where data elements are not placed sequentially or linearly are called non-linear data structures. In a non-linear data structure, we can't traverse all the elements in a single run only. Examples of non-linear data structures are trees and graphs.

D. Why Learn Data Structures and Algorithms?

This article is for those who have just started learning algorithms and wondered how impactful it will be to boost their career/programming skills. It is also for those who wonder why big companies like Google, Facebook, and Amazon hire programmers who are exceptionally good at optimizing Algorithms.

E. Asymptotic Analysis: Big-O Notation and More

In this tutorial, you will learn what asymptotic notations are. Also, you will learn about Big-O notation, Theta notation and Omega notation.

The efficiency of an algorithm depends on the amount of time, storage and other resources required to execute the algorithm. The efficiency is measured with the help of asymptotic notations.

An algorithm may not have the same performance for different types of inputs. With the increase in the input size, the performance will change.

The study of change in performance of the algorithm with the change in the order of the input size is defined as asymptotic analysis.

F. Authors and Affiliations

The class file is designed for, but not limited to, six authors. A minimum of one author is required for all conference articles. Author names should be listed starting from left to right and then moving down to the next line. This is the author sequence that will be used in future citations and by indexing services. Names should not be listed in columns nor group by affiliation. Please keep your affiliations as succinct as possible (for example, do not differentiate among departments of the same organization).

G. Asymptotic Notations

symptotic notations are the mathematical notations used to describe the running time of an algorithm when the input tends towards a particular value or a limiting value.

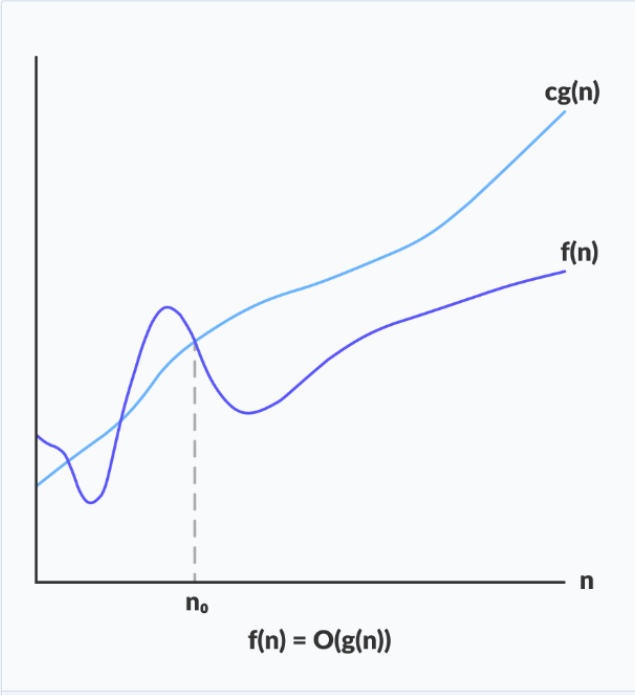
For example: In bubble sort, when the input array is already sorted, the time taken by the algorithm is linear i.e. the best case.

But, when the input array is in reverse condition, the algorithm takes the maximum time (quadratic) to sort the elements i.e. the worst case.

When the input array is neither sorted nor in reverse order, then it takes average time. These durations are denoted using asymptotic notations.

There are mainly three asymptotic notations:

- Big-O notation
- Omega notation
- Theta notation



The above expression can be described as a function f(n) belongs to the set O(g(n)) if there exists a positive constant c such that it lies between 0 and cg(n), for sufficiently large n.

For any value of n, the running time of an algorithm does not cross the time provided by O(g(n)).

Since it gives the worst-case running time of an algorithm, it is widely used to analyze an algorithm as we are always interested in the worst-case scenario.

REFERENCES

"Data Structures using C", Y. Langsam, M. Augenstin and A. Tannenbaum, first edition"