Data Structures, Algorithms, and ADT's

CS240 Spring 19

What are Data Structures?

- What is an unsigned integer value of 1 on a 32 bit computer?
- What can you do with a 1?
- A data structure is any data representation and its associated operations.

Why do Data Structures matter?

- We must figure out a way to organize data in a way that is manageable
- The most obvious example is an unsorted array.

What problem do Data Structures solve?

- Assume you have an unsorted array with an unstable data set
    - For example, you have an array of 10,000 movies on netflix, and the current number of people watching each of these movies.
    - The Movie ID corresponds to an array index which holds the number of people watching that movie.

```
int net_movies_watchers[10000] = {0};
```

What problem do Data Structures solve?

- You are writing an application that displays the top 5 watched movies on netflix at any given moment
- The application allows the user to check how many people are watching any given movie.
  - What operations are important? i.e. What needs to be fast, and what doesn’t have to be?
- Using the proper data structure can make the difference between a program running quickly or not at all.

Resource constraints

- Data Structures require what two resources?
- All data structures and algorithms are measured according to their use of these two resources
- The effectiveness of a Data Structure or Algorithm is considered its ‘Cost’

Selecting a data Structure

- When selecting a data structure to solve a problem:
  - Determine the basic operations that are required.
  - Determining the best ADT for a problem always depends on resource constraints.
  - Select the data structure that best meets these requirements.

Additional Constraints

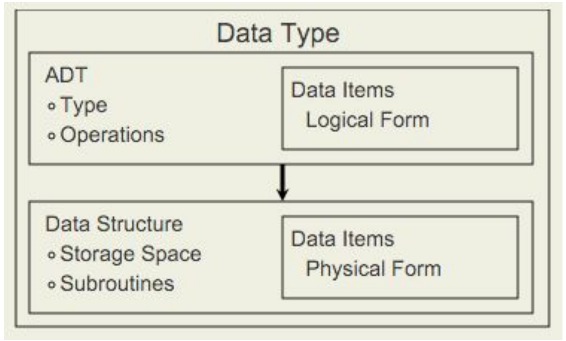
- When is data inserted?
    - **Stable data**
    - **Unstable data**
- How is data accessed?
    - Are all data items processed in the same order every time?
    - "Random access" search generally requires more complex data structures.

Abstract Data Types

- So a Data Structure is just a concrete collection of data and operations.
- A data structure is an implementation of an Abstract Data Type
- ADTs are language independent

How is an ADT different from a Data Structure?

- Important: An ADT does not specify how the data type is implemented.
- An ADTs interface is defined in terms of:
  - a type name
  - a set of operations on that type



ADT vs Data Structure

- Suppose a particular programming environment provides a library that includes a list class.
- A variety of implementations for lists is possible. The particulars of the implementation is the Data Structure.

Course Topics

- In this course we will do 3 things:
  - Add a collection of data structures and algorithms to our toolkit.
  - Explore the idea of tradeoffs
  - Learn how to measure the effectiveness of a data structure or algorithm.
- With each ADT we will explore an algorithm along with that ADT
  - The algorithm will sometimes require that particular ADT, but not always

How to choose?

- Primarily, we will strive for two goals:
  - Design data structures and algorithms that are
    - easy to understand
    - easy to code
    - easy to debug
  - Design data structures and algorithms that makes efficient use of the computer's resources.
- Simplicity vs Simplistic

Classwork

ADT