Welcome to CIT 5940

What You Will Learn in CIT 5940

- Find the syllabus here.
- You will learn:
  - Commonly used data structures and algorithms and their guarantees and tradeoffs
  - How to measure the effectiveness of a data structure or algorithm

Data Structures & Algorithms

- **Data Structures** are abstract containers for data that are designed to enable efficient storage, retrieval, and modification of data.
- **Algorithms**, which you are studying in great detail in CIT 5960, are problem-solving procedures that can be applied over data structures.

Our goal in this course is to help build your programmer's basic toolkit of data structures and some algorithms.

Tradeoffs

There is no "ideal" data structure—each comes with its own advantages and compromises.

- Our job will be to study the costs & benefits of using each data structure.
- Each data structure can be analyzed in terms of:
  - The amount of space required to maintain it
  - The amount of time required to execute its typical operations

Effectiveness

The way that data is stored has a significant impact on the strategies that we use to interact with it!

Example: Searching in an Array

An array is a simple data structure that stores an ordered—but not necessarily sorted—sequence of values.

```
[54, 74, 31, 53, 38, 9, 34, 90, 60, 42, 24, 7, 3, 99, 7, 50, 44, 55]
```

A reasonable procedure to search over this array:

```java
public static boolean contains(int[] array, int target) {
  for (int i = 0; i < array.length; i++) {
    if (array[i] == target) {
      return true;
    }
  }
  return false;
}
```

Example: Searching in an Array

Our approach requires us to do one iteration of the for loop to confirm that `54` is present, three iterations to confirm that `31` is present, and twenty iterations to confirm that `55` is present or that `-15` is not present.

```
[54, 74, 31, 53, 38, 9, 34, 90, 60, 42, 24, 7, 3, 99, 7, 50, 44, 55]
```

Example: Searching in an Array

What if we knew that our array was **sorted?**

- Challenges: we have to sort the array, and then we have to be careful about how we add new elements to a sorted array
- Advantages: we can use a **binary search** to find elements in the array much more quickly!

Example: Searching in a Sorted Array

```java
public static boolean contains(int[] array, int target) {
  int low = 0;
  int high = array.length - 1;
  while (low <= high) {
    int mid = (low + high) / 2;
    if (array[mid] == target) {
      return true;
    } else if (array[mid] < target) {
      low = mid + 1;
    } else {
      high = mid - 1;
    }
  }
  return false;
}
```

Example: Searching in an Array

- Both an Array and a Sorted Array are reasonable structures for storing data in a sequence.
- Both an Array and a Sorted Array allow for a user to search for a given element.
- Finding an element in a Sorted Array can be much faster than finding an element in an Array since we can use a binary search to rule out half of the positions in the Sorted Array at each step.

Effectiveness

- Some data structures support certain operations that others do not.
- Among the data structures that support operations we need, we can evaluate which are more suitable for speed & space based on the data we're working with.

Goals

We want you to be able to:

- design algorithms that are easy to understand, code, & debug by using data structures
- design software that makes efficient use of the computer's resources

Administrative Stuff

Pre-requisites and Co-requisites

- Programming: CIT 5900/5910 or CIS 1200
  - Comfort with writing & testing medium size programs in an object-oriented language
  - Java experience is very helpful
- Math: CIT 5920 or CIS 1600
- Algorithms: CIT 5960 (co-requisite)

Homeworks

- HW1: Catch a Plagiarist
- HW2: Algorithm analysis (written)
- HW3: File compression
- HW4: Blockly
- HW5: Autocomplete
- HW6: Search Engine
- HW7: Graphs
- HW8: Group project

Assignment Expectations

- Assignments are largely autograded
  - Instant feedback on submission! 🎉
  - Transparent grading criteria! 🎉
  - Defines a narrow specification that must be conformed to! 🙃
- You will likely need help during office hours
  - We have several OH per week, but there are more of you than there are of us.
  - You are expected to be **writing your own test cases!**
    - (it's part of your grade)
    - it helps streamline office hours questions & keep queues short.

Assignment Expectations

1. Understand the Problem
    i. What are the relevant concepts and how do they relate?
2. Formalize the Interface
    i. How should the program interact with its environment?
3. Write Test Cases
    i. How does the program behave on typical inputs?
    ii. How does the program behave on **unusual** inputs, or invalid ones?
4. Implement the Required Behavior
    i. Decompose the problem into simpler ones & apply this process to each.

Questions & Website Tour