CS 3345: List of topics

- Mathematical background
- Running time analysis of programs; order notation
- Lists, Stacks, Queues
- Trees, Binary search trees
- Hashing
- Binary heaps, Priority Queues
- Sorting and Searching
- Graphs
- DFS and topological ordering
- Shortest paths, BFS, Dijkstra's algorithm
- Maximum flow problem
- Minimum spanning trees: Prim/Kruskal's algorithms, Disjoint sets
- Algorithm design techniques; advanced data structures

Core of CS

- The following classes are part of core CS, and the knowledge you get from them will serve you during your entire career:
    - CS 2, Discrete Math, Data structures, Algorithms
- Not enough if you understand what has been done ...
- … need to be able to apply the knowledge to solve new problems ...
- … and know that you applied it the right way
- An easy "A" in any of these classes is a disservice to you
- Time spent on these classes is an investment in your future
- Learning these subjects on your own, without guidance, is hard

What is covered in this class?

- Discuss the data structures from CS 2336 more formally, and include running time analyses, along with introduction to proof of correctness.
- Study a few more advanced data structures.
- Study more algorithms, including additional graph algorithms.
- This is a bridge course between CS 2336 and CS 4349.

CS 2336	CS 3345	CS 4349
Programming	Programming + Theory	Theory

Skill levels

1. Knowledge of data structures available in libraries (e.g., ArrayList, TreeMap, HashSet), their operations, and their use cases.
2. How are these data structures implemented?
3. Which operations are efficient and which ones should be avoided (e.g., contains operation of iterators, vs contains operation of lists)?
4. How are well-known algorithms implemented? What data structures do they use?
5. Is a given implementation close to optimal? Can its speed be improved by a better choice of data structures and algorithms?
6. Design and implement new data structures and algorithms.

Standard track vs Intensive track

- Standard track (default):
  - Weekly assignments (short, written) + 4-5 programming projects

- Intensive track (by choice):
  - Weekly assignments + 4-5 programming projects + weekly programming problems
  - Enrollment survey will come on elearning soon
  - Need this level of training to prepare for a career in the top 10 companies

Problems in homework assignments

- Get understanding of inner workings of data structures and algorithms:
  - Sort the following list: …
  - Insert 3 into the following AVL tree and delete 17 from the resulting tree.
  - Find the length of a shortest path from s to t in the following graph.
- This knowledge is needed to debug implementations.

Problems in exams and job interviews

- < 20%: As in homeworks (inner workings of ds&a).
- > 80%: Design, applications. Examples:
  - Design a data structure for the following operations: …
  - Design an algorithm for the following problem: …
  - How does merge sort work?
  - How are priority queues implemented using binary heaps?
  - Write algorithm for add operation of binary search trees.
  - Show run-time analysis of the following algorithm.
  - Prove correctness of a theorem.

CS 3345.004 Data Structures Lecture #1

Example: Fibonacci numbers:

$$F(n) = F(n-1) + F(n-2), \quad n > 1$$

$$F(0) = 0, \quad F(1) = 1$$

```
fib(n):
  if n > 1 then
    return fib(n-1)
         + fib(n-2)
  else
    return n
```

RT = exponential in n

$= O(c^n)$

$c = \frac{\sqrt{5}+1}{2} \approx 1.617$

Dynamic Program:

```
F[0] ← 0
F[1] ← 1
for i ← 2 to n do
    F[i] ← F[i-1]
         + F[i-2]
return F[n].
```

$RT = O(n)$.

$O(\log n)$ time?

Yes — to be discussed later.

Next class: RT analysis, Order notation, Recurrences, Math background.