CS 3305

Syllabus

Syllabus: CS 3305.HON Syllabus (Spring 2021)

Textbook: Mathematics for Computer Science

Meeting times: T/Th 11:30am-12:45pm

Last day of lecture: Thursday, May 6

Grading Criteria:

Note: Honorlock will be used for all exams, 24 hour window available.

1. Midterm 1: 20%
2. Midterm 2: 20%
3. Final exam: 20%
4. Homework: 25%
5. Paper review: 10%
6. Class participation: 5%

See syllabus (linked above) for more information.

Day 1: 1/19

Why Discrete?

Rigorous understanding of algorithms: Proving the correctness of algorithms and verify that their performance is as expected.

Time complexity and space complexity: space complexity is also important because RAM can only hold so much information before slowing down (due to accessing hard drive for swapping)

Asymptotic Bound

1. **Big-O**: Upper bound of complexity
2. **Big-Omega**: Lower bound
3. **Big-Theta**: Tight bound, set intersection of **Big-O** and **Big-Theta**

Modular Arithmetic

$a \equiv b \pmod{m} \iff a \mod m = b \mod m \iff (a-b) \equiv 0 \pmod{m}$

Representation of Numbers

- **Decimal**: 0, 1, 2, . . . , 9
- **Binary**: 0, 1
- **Hexadecimal**: 0, 1, 2, . . . , 9, A, B, C, D, E, F

Binary multiplying by power of 2

$5_{10} = 101_2 \implies 10_{10} = 1010_2 \implies 20_{10} = 10100_2$

Bit shift left for multiplying by 2, bit shift right for dividing (truncate) by 2.

Prime and Composite Numbers

Prime: only 2 factors, 1 and itself

Composite: more than 2 factors

Definition

Sieve of Eratosthenes

Discard multiples of prime numbers to *sieve* through set of all numbers less than n.

```
def primes_less_than_n(n):
    primes = set([i for i in range(2, n + 1)])
    for i in range(int(sqrt(n))):
        if i in primes:
            j = 2
            while i * j <= n:
                primes.discard(i * j)
                j += 1
    return primes
```

Induction and Recursion

- **Recursion**: express problem as smaller instances
- **Induction**: used to prove recursion solutions
- **Strong Induction**: induction based on all previous steps rather than just the one previous step