COT3100 Exam 3 review

Carson Mulvey

Mar 26, 2024

Topics

Exam details

- Time: Thursday, March 28th, 8:20 to 10:20 PM
- Topics:
  - 3.1 to 3.3
  - 4.1 to 4.3; 4.6
  - 5.1 to 5.4
- Things to bring:
  - Writing utensils
  - Handwritten reference sheet (8.5x11)
  - **4 function** calculator
  - ID (UF ID, state ID, or ID on phone)

Topics

Complexity review

Search algorithms:

- Linear search - $O(n)$
- Binary search - $O(\log_2 n)$

Sorting algorithms:

- Bubble sort - $O(n^2)$
- Selection sort - $O(n^2)$
- Insertion sort - $O(n^2)$
- Merge sort - $O(n \log n)$
- Quick sort - $O(n \log n)$ (worst case $O(n^2)$)

Note

The following order represents the growth rates of functions from slowest to fastest:

$$1 \ll \log n \ll n \ll n \log n \ll n^2 \ll (\text{polynomials}) \ll 2^n \ll n! \ll n^n$$

*new algorithms, be sure to review!

Big O, big Ω, and big Θ

Definition

A function $f(x)$ is $O(g(x))$ if there are C and k such that for all $x > k$,

$$|f(x)| \leq C|g(x)|.$$

Definition

A function $f(x)$ is $\Omega(g(x))$ if there are C and k such that for all $x > k$,

$$|f(x)| \geq C|g(x)|.$$

Definition

A function $f(x)$ is $\Theta(g(x))$ if it is both $O(g(x))$ and $\Omega(g(x))$

Remember that big O is an **upper bound**, big Ω is a **lower bound**, and Θ is grows at the **same rate** (asymptotically).

Practice

Problem

For $f(n) = 4n^3 + \log(n^3)$, prove that $f(n)$ is $\Theta(n^3)$.

Practice

Problem

Find the optimal big O for the following functions:

- $f(x) = x^2 + \log_{2024}(x^{2024})$
- $g(x) = 2^{2024!}$
- $h(x) = 2024^x + x!$

Topics

Number bases

Skills to know:

- Convert from decimal to another base
- Convert from another base to decimal
- **Convert between bases of similar powers***
- Adding and multiplying within a base

*Note

We can convert between non-decimal bases easily if one they share the same primes. For example, from binary to hexadecimal, because $2^4 = 16$, we group by 4s. For example:

$$(1001\ 0101\ 0111\ 1010)_2 = (957A)_{16}$$

Practice

Problem

Convert $(11101010)_2$ to decimal and hexadecimal. Then go backwards to verify your work.

Practice

Problem

Find $(1010)_2 \cdot (110)_2$ without converting to decimal.

Division and modular arithmetic

Skills to know:

- Division algorithm
- Divides ('$|$')
- Basic modular arithmetic

Practice

Problem

Find the quotient and remainder of the following:

- $23 \div 5$
- $-15 \div 4$
- $0 \div 3$

Practice

Problem

Show that $a|b$ is equivalent to $b \equiv 0 \pmod{a}$.

Practice

Problem

Prove that if $a \equiv b \pmod{n}$ and $c \equiv d \pmod{n}$, then $ac \equiv bd \pmod{n}$.

Factorization and primes

Skills to know:

- Definitions: prime and relatively prime
- Prime factorization
- gcf and lcm
- Euclidian algorithm

Practice

Problem

Find the prime factorizations of 228 and 126. Then find $\gcd(228, 126)$ and $\mathrm{lcm}(228, 126)$. Are 228 and 126 relatively prime? Finally, verify your answer to $\gcd(228, 126)$ using the Euclidian algorithm.

Topics

Induction and strong induction

Skills to know:

- Writing a proof by induction
- Writing a proof by strong induction

Recursion

Skills to know:

- Understanding a recursive definition
- Proving an explicit form by induction
- Coding a recursive solution to a problem
- Analyzing time complexity of a recursive algorithm (see handout)

Practice

Problem

Code an algorithm min that finds the minimum element of $a_1, a_2, \ldots, a_n$ recursively using the first $n-1$ elements.

Practice (harder)

Problem

Let pyr(n) be a function taking a positive integer that finds the amount of triangles in a 2D pyramid of height n made of equilateral triangles.

1. Find a recursive definition.
2. Code a recursive algorithm pyr(n) that finds pyr(n)
3. Find pyr(1), pyr(2), and pyr(3). Conjecture an explicit formula.
4. Prove your explicit formula by induction.