CSE101: Design and Analysis of Algorithms

Ragesh Jaiswal, CSE, UCSD
Administrative Information
Course Instructor:
- Ragesh Jaiswal
- Office: 4122, CSE
- Email: raja@ucsd.edu
- Office hours: 10:00 - 11:00am, Monday

Course Timing:
- Lectures: Mon, Wed, Fri, 8:00 - 8:50am
- Discussion 1: Mon 3-4, WLH 2111
- Discussion 2: Mon 4-5, WLH 2111
- Discussion 3: Mon 5-6, WLH 2111

Teaching Assistants:
- Qiao Fang
- Scott Fernandez
- Sanjeev Shenoy

Tutors: TBA
Grading Scheme

1. **Homework**: 10%
2. **Quizzes**: 20%
3. **Midterms**: 30% (2 midterms in class, 15% each)
4. **Final**: 40%

Policy on cheating:

- Students using unfair means will be severely penalised.
Textbook: Algorithm Design by Jon Kleinberg and Eva Tardos.

I will be following this book very closely. So, it will be a good idea to get a copy of this book.

Other reference books:

- Algorithms by Sanjoy Dasgupta, Christos Papadimitriou, and Umesh Vazirani.

Course webpage:
http://www.cs.ucsd.edu/~rajaiaiswal/cse101/.

The site will contain course information, references, homework, course slides, and announcements. Please check this page regularly.
Recap. of Data Structures and Algorithms
What is an algorithm?
Recap.

- What is an algorithm?
- How do we measure the performance of an algorithm?
- How do we know whether there is an efficient algorithm for a given problem.
• What is an algorithm?
  • A step-by-step way of solving a problem.
• How do we measure the performance of an algorithm?
• How do we know whether there is an efficient algorithm for a given problem.
• Notion of Efficiency:
  • Worst-case analysis: Largest possible running time over all input instances of a given size $n$ and then see how this function scales with $n$.
  • Polynomial time: The worst-case running time is some polynomial in the input size (e.g., $T(n) = 5n^3 + 3n^2 + 2n + 10$)
Recap.

- **What is an algorithm?**

- How do we measure the performance of an algorithm?

- How do we know whether there is an efficient algorithm for a given problem.

- **Notion of Efficiency:**
  - **Worst-case analysis:** Largest possible running time over all input instances of a given size \( n \) and then see how this function scales with \( n \).
  - **Polynomial time:** The worst-case running time is some polynomial in the input size (e.g., \( T(n) = 5n^3 + 3n^2 + 2n + 10 \))

![Plot of \( n^2 \) and \( 2n + 2 \)](image.png)

**Figure:** Plot of \( n^2 \) and \( 2n + 2 \)
What is an algorithm?

How do we measure the performance of an algorithm?

How do we know whether there is an efficient algorithm for a given problem.

Notion of Efficiency:
- Worst-case analysis: Largest possible running time over all input instances of a given size $n$ and then see how this function scales with $n$.
- Polynomial time: The worst-case running time is some polynomial in the input size (e.g., $T(n) = 5n^3 + 3n^2 + 2n + 10$)

Asymptotic order of growth ($O$, $\Omega$, $\Theta$):
- $T(n)$ is $O(f(n))$ (or $T(n) = O(f(n))$) iff there exists constants $c > 0, n_0 \geq 0$ such that for all $n \geq n_0$, we have $T(n) \leq c \cdot f(n)$. 
Growth rates:

Arrange the following functions in ascending order of growth rate:

- $n$
- $2\sqrt{\log n}$
- $n^{\log n}$
- $n^{\log n}$
- $2^{\log n}$
- $n/\log n$
- $n^n$