Lecture 1: Introduction

Agenda:

- Welcome to CMPUT 204 - Algorithms I
- Official course information
- Course overview
- Announcements
- Basic concepts & Math induction review

Reading:

- Textbook pages $5 - 14$

Official course information

- Course webpage:
  http://www.cs.ualberta.ca/~bowling/classes/cmput204

- Mark distribution:
  - 5 assignments 35% $(6\% + 8\% + 8\% + 8\% + 5\%)$
  - 3 midterms at 10% each (in class, 50 minutes)
  - Final 35% (3 hours)

- Lectures follow textbook: Introduction to Algorithms, by Cormen/Leiserson/Rivest/Stein (2nd Edition, 2001)

- Michael Bowling (Section A1)
  - ATH339, 492-1766, bowling@cs.ualberta.ca
  - Office hours: TBA

- Mohammad Salavatipour (Section A2)
  - ATH303, 492-1759, mreza@cs.ualberta.ca
  - Office hours: Monday & Friday 2:00-3:00

Course overview

- 204 **Algorithm I**, Topics covered:
  - Introduction to algorithms
  - Algorithms: sorting, matrices, graphs, sets
  - Design: divide-and-conquer, dynamic programming, greedy
  - Analysis: model assumptions, worst/average/best case, asymptotic, reduction, complexity classes P and NP, hard problems
  - Calendar and outline with reading list on course webpage

- 304 **Algorithms II**
  - More advanced algorithms, and their design and analysis

- 474 **Formal Languages, Automata and Computability**
  - More formal approach to models, complexity, and computability

- Other related courses
  - * 419 **Algorithmic Graph Theory** (occasionally)
    * 497 **Combinatorial Algorithms** (occasionally)
  - * 366 **Intelligent Systems**
    * 466 **Machine Learning**

Announcements

- No seminars first week
- Assignment #1 available (from course webpage)
- Attend your registered section and seminar
- Some of the materials covered in seminars

Role of algorithms in computing

- Algorithm
- Typical problems
  - Optimizations (minimizing cost, maximizing profit, etc.)
  - Internet (routing, search, mining, etc.)
  - E-commerce
  - Operations research (resource management, scheduling, etc.)
- Easy and hard problems
- Issues while designing Algorithms : time/space management

Basic concepts

- Problem
- Instance
- Algorithm
- Issues for a given algorithm
  - Correctness — often via **Loop Invariants**, proved by induction
  - Analysis — measuring resource requirement
    - Running time
    - Space
  - Optimality
- Algorithm design concepts
  - Divide and conquer
  - Data structure
  - Dynamic programming
  - Exhaustive enumeration
  - Greedy
  - Reduction (useful for showing problems are *hard*)
  - Branch and bound
  - Others (probably won't cover)

Mathematical Induction:

- Steps in a proof by mathematical induction:
  1. Define the statement to be proved precisely
  2. Statement holds for the base case — figure out what is the base case
  3. Assuming the statement holds for some intermediate case, prove that it holds for the *next* case

     — often, the assumption should be used in the proof

- Examples:
  1. Prove by induction that $\sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6}$.

     Proof. Base case $n = 1$: LHS is 1; RHS is 1 too. So the equality holds.

     Assuming the equality holds for $n = k$ ($k \geq 1$). That is, $\sum_{i=1}^{k} i^2 = \frac{k(k+1)(2k+1)}{6}$. For $n = k+1$, LHS is

     $$\begin{aligned}\sum_{i=1}^{k+1} i^2 &= \sum_{i=1}^{k} i^2 + (k+1)^2 \\ &= \frac{k(k+1)(2k+1)}{6} + (k+1)^2 \\ &= \frac{(k+1)(2k^2+k+6(k+1))}{6} \\ &= \frac{(k+1)(k+2)(2k+3)}{6}.\end{aligned}$$

     RHS is $\frac{(k+1)(k+2)(2k+3)}{6}$.

     So the equality holds when $n = k+1$. Therefore, the equality holds for all $n \geq 1$.
  2. Prove by induction that $\sum_{i=1}^{n} \frac{1}{i(i+1)} = \frac{n}{n+1}$.

Have you understood the lecture contents?

well	ok	not-at-all	topic
☐	☐	☐	coverage, marking scheme !!!
☐	☐	☐	basic concepts in algorithmics
☐	☐	☐	proof by math induction