CPSC 320
Intermediate Algorithm Design and Analysis
Course Information and Introduction

Jan Manuch

2015W1

Course Information CPSC 320

1. **Instructor : Jan Manuch**
   - Email : jmanuch@cs.ubc.ca
   - Office : ICCS 247
2. **Teaching Assistants:**
   - Dennis Park (dennispk@cs.ubc.ca)
   - Ehsan Kermani (ehsanmok@cs.ubc.ca)
   - Shiwen He (shiwenhe@cs.ubc.ca)
3. **Course Webpage :**
   https://connect.ubc.ca
4. Discussions through Piazza (linked through Connect) — register as soon as possible!

Course Information CPSC 320

Tutorials:

1. T1B Mondays 3p-4p (Ehsan) DMP 201
2. T1C Tuesdays 1p-2p (Dennis) DMP 101
3. T1E Mondays 9a-10a (Ehsan) DMP 201
4. T1F Wednesdays 12p-1p (Shiwen) DMP 201
5. register to one, attend whichever you like

Office hours:

1. instructor 3 hours a week (Mondays and Fridays)
2. each TA (2 hours a week)
3. You will be able to vote for a couple of options on Piazza soon!

Course Information CPSC 320

Review: summations of sequences; proof techniques; mathematical induction

Textbook:

Algorithm Design, J. Kleinberg, É. Tardos, Addison Wesley, 2006.

Grading Scheme:

- Assignments (20%):
  - submit to the assignment box (X235?)
  - -33% per day late (late assignments must be scanned and submitted by email to me)
  - 7–8 assignments
  - must submit at least 5 of them
- 2 Midterms ($2 \times 15\%$)
  - In class. The first around Thanksgiving and the second around Remembrance Day.
- Final Exam (50%)

You must get at least 50% on the weighted average of the exams

Course Information CPSC 320

Objective: Introducing concepts and problem-solving techniques that are used in the design and analysis of efficient algorithms.
Topics:

1. Searching and sorting;
2. Data structures, such as skip lists;
3. Mathematical tools, like O notation, recurrence relations, and amortized analysis;
4. Algorithm design techniques, for instance randomization, greediness and dynamic programming.

Much of the material is formal (mathematical) in nature, and hence proofs will constitute an important part of the course.

Course Information CPSC 320

At the end of the course, you will be able to:

1. Recognize which algorithm design technique(s), such as divide and conquer, prune and search, greedy strategies, or dynamic programming was used in a given algorithm.
2. Select and judge several promising paradigms and/or data structures (possibly slightly modified) for a given problem by analyzing the problem's properties.
3. Implement a solution to a problem using a specified algorithm design paradigm, given sufficient information about the form of that problem's solution.
4. Select, judge and apply promising mathematical techniques (such as asymptotic notations, recurrence relations, amortized analysis and decision trees) to establish reasonably tight upper and lower bounds on the running time of algorithms.
5. Recognize similarities between a new problem and some of the problems you have encountered, and judge whether or not these similarities can be leveraged towards designing an algorithm for the new problem.

Problem Solving

Steps:

1. formulate the problem exactly (sometimes it's useful to abstract from details or simplify the problem first, and then revisit those later)
2. propose an algorithm
3. analyze the algorithm and prove its correctness OR go back to (2)
4. analyze the time (space) complexity of the algorithm