# Algorithms and complexity

week 13

# Topics covered

- Graphs and graph algorithms (review)
- Greedy algorithms
- Divide and conquer
- Dynamic programming
- Flow problems
- Reductions
- NP-completeness

- *this is not a complete list*

# Main topics

- Graphs
  - graph representations
  - graph traversals (BFS DFS)
  - directed graphs
  - topological sort
- Combinatorics and other background
  - big Oh notation
  - solving recurrence relations, the master theorem
  - sets and counting, combinations, length of representations
- Data structures
  - priority queues, sets (union-find for advanced)

# Main topics

- Greedy algorithms
  - interval scheduling
  - interval partitioning
  - shortest paths
  - minimum spanning trees
- Divide and conquer
  - mergesort and counting inversions
  - solving recurrences, the master theorem
  - closest pair of points (advanced)
  - Strassen's matrix multiplication algorithm

# Main topics

- Dynamic programming
  - weighted interval scheduling
  - memoization and recursion
  - subset sums and knapsack problems
  - sequence alignment
  - shortest paths (Bellman Ford)
  - segmented least squares

# Main topics

- Network flow
  - max flow and min cut
    - flow value lemma, weak duality
  - max flow and Ford Fulkerson (augmenting paths)
    - scaling for the Ford Fulkerson algorithm
  - applications of max flow
    - bipartite matching via max flow
    - edge disjoint paths
    - disconnecting a network
    - circulations with demands (and lower bounds)
    - survey design
    - project selection

# Main topics

- NP and intractability
  - reductions
  - P and NP, certificates and verification
  - NP-complete, NP-hard and NP-complete problems
    - list of NP-complete problems: vertex cover, independent set, 3SAT and satisfiability, set cover, hamiltonian cycle, longest path, traveling salesman problem, clique
  - co-NP (asymmetry of NP)

# Main topics

- dealing with intractability
  - small vertex covers
  - independent set on trees
  - weighted independent sets on trees
  - randomized algorithm for global minimum cut (advanced)
  - approximation algorithm for makespan scheduling (advanced)

# and some advanced topics

- testing bipartiteness, 2-coloring
- the union find data structure
- closest pair of points
- sequence alignment in linear space
- finding negative cycles in a graph
- randomness, probabilistic analysis of algorithms
- processes accessing a database (contention resolution)
- randomized algorithm for global minimum cut (Karger's algorithm)
- randomized algorithm for MAX3SAT (tutorial)
- approximation algorithm for the traveling salesman problem

# Marks and webCT and tutorial sessions

- all progressive marks will go on webCT
- please make sure you have a mark for every component you have submitted at the end of the semester (at the time of the exam)
- if there are any problems contact the unit coordinator

- tutorial sessions this week (week 13)
  - **no tutorials**, but
  - tutor(s) will be available to answer any questions you might have
    - thursday 11am-12noon and
    - thursday 2pm-3pm
  - You can also have a look at your quiz 2 marking during that time

# Assessment components

- assessment for this unit:
  - 10% Assignment 1
  - 10% Quiz 1
  - 10% Assignment 2
  - 10% Quiz 2
  - 60% Final exam
- following the University/School of IT policy, you need at least 40% in the final exam and 50% overall to get a passing mark for this unit.

# Final exam

- 2hours, five questions to answer, 2 pages of notes allowed
- each question is 20% of the exam
- Q1
  - true false questions (similar to the quiz)
  - wrong answers have negative marks, no answer ok
  - also multiple choice question
- Q2
  - short tracing questions, several short subquestions
- Q3
  - algorithmic problems with relatively short answers
  - several subquestions

# Final exam

- Q4
  - algorithmic problems that require longer answers
  - similar to the assignment and tutorial questions for example
  - this is the "difficult one" (i think)
- Q5 normal stream only (COMP2007/5211)
  - algorithmic problems mostly longer answers
  - possibly a mix with tracing algorithms and short answers

- Q6 advanced stream only (COMP2907)
  - algorithmic questions with longer answers

# Examples of hard questions from previous years

- Prove that a new problem (defined in the question) is NP-complete.
- Give a dynamic programming algorithms for the longest rising trend (part of assignment 2)
- Give a polynomial time algorithm for maxflow in networks with capacities on the vertices
- All tutorial hard questions are good examples as well

# Hints

- there will be tracing question(s)
- there will be questions on general graph problems
- there are questions about reductions, NP-completeness and NP-hardness
- the exam usually has plenty of flow related questions
  - for example you could be asked to design a greedy algorithm for a flow related problem
- you might get questions to describe how a certain algorithm that we have covered in the lectures (or tutorials) works
- the questions are chosen randomly from all the material
- the hard questions (in Q4) may also be "longer" to solve

# Algorithmic units of study

- Some "algorithmic" units you might be interested in
  - Algorithms for the life sciences - bioinformatics (3rd year)
  - Artificial Intelligence (3rd year)
  - Computational geometry (honors)
  - Information visualization, graph drawing (honors)
  - Data mining
- Advanced topics (4th year / honours)
  - topics change from year to year. Possible topics that may appear in the future (depending on interest)
    - algorithmic game theory (2008)
    - approximation algorithms
    - computational complexity theory (2010)

good luck with all your exams