Course overview and Administrative matters[1]

Myrto Arapinis and **Colin Stirling**
School of Informatics
University of Edinburgh

September 15, 2014

[1] Slides mainly borrowed from Richard Mayr

Administrative matters

Teaching staff

Lecturers:

- Dr. Myrto Arapinis, first half of the term
- Pr. Colin Stirling, second half of the term

Course TA:

- TBA

Course Secretary (ITO):

- Kendall Reid (kr@inf.ed.ac.uk)

Course web page

http://www.inf.ed.ac.uk/teaching/courses/dmmr/

Contains important info:

- Lecture slides
- Tutorial sheet exercises
- Course organization
- etc.

Tutorials

- You should receive email from the ITO informing you of preliminary allocation of tutorial groups

Tutorials

- You should receive email from the ITO informing you of preliminary allocation of tutorial groups
- See link on course web page for current assignment of tutorial groups

Tutorials

- You should receive email from the ITO informing you of preliminary allocation of tutorial groups
- See link on course web page for current assignment of tutorial groups
- If you cant make the time of your allocated group, please email Kendall suggesting some groups you can manage

Tutorials

- You should receive email from the ITO informing you of preliminary allocation of tutorial groups
- See link on course web page for current assignment of tutorial groups
- If you cant make the time of your allocated group, please email Kendall suggesting some groups you can manage
- If you change tutor groups for any reason, you must let Kendall and the ITO know (Because your marked coursework is returned at the tutorial groups)

Tutorials

- You should receive email from the ITO informing you of preliminary allocation of tutorial groups
- See link on course web page for current assignment of tutorial groups
- If you cant make the time of your allocated group, please email Kendall suggesting some groups you can manage
- If you change tutor groups for any reason, you must let Kendall and the ITO know (Because your marked coursework is returned at the tutorial groups)
- Tutorial attendance is mandatory. If you miss two tutorials in a row, your PT (DoS) will be notified

Tutorials and (marked) exercises

- Weekly exercise sheets, available every Wednesday 2pm on the course web page

Tutorials and (marked) exercises

- Weekly exercise sheets, available every Wednesday 2pm on the course web page
- The last question on every sheet will be graded. The coursework grade contributes 15% to the total course grade, and every one of the 9 exercise sheets counts 1/9th of the coursework grade

Tutorials and (marked) exercises

- Weekly exercise sheets, available every Wednesday 2pm on the course web page
- The last question on every sheet will be graded. The coursework grade contributes 15% to the total course grade, and every one of the 9 exercise sheets counts 1/9th of the coursework grade
- Starting in week 2, deadline for submission of each tutorial sheet is Wednesday at 4:00pm at the ITO (they also have a collection box)

Tutorials and (marked) exercises

- Weekly exercise sheets, available every Wednesday 2pm on the course web page
- The last question on every sheet will be graded. The coursework grade contributes 15% to the total course grade, and every one of the 9 exercise sheets counts 1/9th of the coursework grade
- Starting in week 2, deadline for submission of each tutorial sheet is Wednesday at 4:00pm at the ITO (they also have a collection box)
- Solutions will be discussed in tutorials the following week. Graded sheets are returned in tutorials (or collected later from the ITO)

Tutorials and (marked) exercises

- Weekly exercise sheets, available every Wednesday 2pm on the course web page
- The last question on every sheet will be graded. The coursework grade contributes 15% to the total course grade, and every one of the 9 exercise sheets counts 1/9th of the coursework grade
- Starting in week 2, deadline for submission of each tutorial sheet is Wednesday at 4:00pm at the ITO (they also have a collection box)
- Solutions will be discussed in tutorials the following week. Graded sheets are returned in tutorials (or collected later from the ITO)
- Exception: no tutorial on Week 1

Textbook

- Kenneth Rosen, **Discrete Mathematics and its Applications**, 7th Edition, (Global Edition) McGraw-Hill, 2012
- Available at Blackwells
- For additional material see the course webpage

Grading

- Written Examination: 85%
- Assessed Assignments: 15%. Each one of the 9 exercise sheets counts equally, *i.e.* 1/9th

Course overview

Intended learning outcomes

Intended learning outcomes

- Reason mathematically about basic (discrete) structures (such as numbers, sets, graphs, and trees) used in computer science

Intended learning outcomes

- Reason mathematically about basic (discrete) structures (such as numbers, sets, graphs, and trees) used in computer science
- Use of mathematical and logical notation to define and formally reason about mathematical concepts such as sets, relations, functions, and integers, and discrete structures like trees, graphs, and partial orders

Intended learning outcomes

- Reason mathematically about basic (discrete) structures (such as numbers, sets, graphs, and trees) used in computer science
- Use of mathematical and logical notation to define and formally reason about mathematical concepts such as sets, relations, functions, and integers, and discrete structures like trees, graphs, and partial orders
- Evaluate elementary mathematical arguments and identify fallacious reasoning

Intended learning outcomes

- Reason mathematically about basic (discrete) structures (such as numbers, sets, graphs, and trees) used in computer science
- Use of mathematical and logical notation to define and formally reason about mathematical concepts such as sets, relations, functions, and integers, and discrete structures like trees, graphs, and partial orders
- Evaluate elementary mathematical arguments and identify fallacious reasoning
- Construct inductive hypothesis and carry out simple induction proofs

Intended learning outcomes

- Reason mathematically about basic (discrete) structures (such as numbers, sets, graphs, and trees) used in computer science
- Use of mathematical and logical notation to define and formally reason about mathematical concepts such as sets, relations, functions, and integers, and discrete structures like trees, graphs, and partial orders
- Evaluate elementary mathematical arguments and identify fallacious reasoning
- Construct inductive hypothesis and carry out simple induction proofs
- Use graph theoretic models and data structures to model and solve some basic problems in Informatics (e.g., network connectivity, etc.)

Intended learning outcomes

- Reason mathematically about basic (discrete) structures (such as numbers, sets, graphs, and trees) used in computer science
- Use of mathematical and logical notation to define and formally reason about mathematical concepts such as sets, relations, functions, and integers, and discrete structures like trees, graphs, and partial orders
- Evaluate elementary mathematical arguments and identify fallacious reasoning
- Construct inductive hypothesis and carry out simple induction proofs
- Use graph theoretic models and data structures to model and solve some basic problems in Informatics (e.g., network connectivity, etc.)
- Prove elementary arithmetic and algebraic properties of the integers, and modular arithmetic, explain some of their basic applications in Informatics, *e.g.* to cryptography

Intended learning outcomes (cont.)

- Compare the asymptotic growth rates of basic functions; derive asymptotic bounds, and limits, for simple series and recurrence relations

Intended learning outcomes (cont.)

- Compare the asymptotic growth rates of basic functions; derive asymptotic bounds, and limits, for simple series and recurrence relations
- Use these to derive bounds on the resource consumption (e.g., running time) of simple iterative and recursive algorithms

Intended learning outcomes (cont.)

- Compare the asymptotic growth rates of basic functions; derive asymptotic bounds, and limits, for simple series and recurrence relations
- Use these to derive bounds on the resource consumption (e.g., running time) of simple iterative and recursive algorithms
- Calculate the number of possible outcomes of elementary combinatorial processes such as permutations and combinations

Intended learning outcomes (cont.)

- Compare the asymptotic growth rates of basic functions; derive asymptotic bounds, and limits, for simple series and recurrence relations
- Use these to derive bounds on the resource consumption (e.g., running time) of simple iterative and recursive algorithms
- Calculate the number of possible outcomes of elementary combinatorial processes such as permutations and combinations
- Be able to construct discrete probability distributions based on simple combinatorial processes, and to calculate the probabilities and expectations of simple events under such discrete distributions

Foundations: Logic and Proof

- Review of propositional logic (INF1A): logical connectives, truth tables, basic laws for Boolean logic (double negation, commutativity, De Morgans laws, law of excluded middle, ...), normal forms (disjunctive and conjunctive normal form)

- Rudimentary predicate (first-order) logic: existential and universal quantification, basic algebraic laws of quantified logic (duality of existential and universal quantification)

- The structure of a well-reasoned mathematical proof; Proof strategies: proofs by contradiction, proof by cases; examples of incorrect proofs (to build intuition about correct mathematical reasoning)

Foundations: Sets, Functions and Relations, Counting

- Sets (naive): operations on sets: union, intersection, set difference, laws of set operations (Boolean algebra: same as Boolean logic), Venn diagrams, the powerset operation, examples of finite and infinite sets (the natural numbers), examples of set-builder notation (set comprehension notation). Ordered pairs, n-tuples, and Cartesian products of sets
- Relations: (unary, binary, and n-ary) properties of binary relations (symmetry, reflexivity, transitivity). partial orders, total orders functions viewed as relations
- Functions: injective, surjective, and bijective functions, inverse functions, composition of functions
- Rudimentary counting: size of the Cartesian product of two finite sets, number of subsets of a finite set, (number of n-bit sequences), number of functions from one finite set to another

Basic algorithms

- Concept and basic properties of an algorithm
- Basics of growth of function, and complexity of algorithms: asymptotic, Big-O-notation, (and little-o, Big-Omega, Big-Theta, etc.), comparison of growth rate of some common functions

Basic number theory and cryptography

- Integers and elementary number theory (divisibility, GCDs and the Euclidean algorithm, prime decomposition and the fundamental theorem of arithmetic)
- Modular arithmetic (congruences, Fermats little theorem, the Chinese remainder theorem)
- Application: RSA public-key cryptography (explain that primality testing is "easy" (polynomial), while factoring is thought to be "hard" (exponential))

Induction and recursion

- Principle of mathematical induction (for positive integers)
- Examples of proofs by (weak and strong) induction
- Recursive definitions, and Structural induction
- Examples of recursive definitions: well-formed propositional logic syntax, well-formed regular expressions (Inf1-C&L), Fibonacci numbers, and other, recursively defined functions
- Examples of structural induction proofs for: propositional logic, regular expressions, Fibonacci, etc.

Counting

- Basics of counting
- Pigeon-hole principle
- Permutations and combinations
- Binomial coefficients, binomial theorem, and basic identities on binomial coefficients
- Generalizations of permutations and combinations (e.g., combinations with repetition/replacement)
- Stirling's approximation of the factorial function

Graphs

- Directed and undirected graph: definitions and examples in Informatics
- Adjacency matrix representation
- Terminology: degree (indegree, outdegree), and special graphs: bipartite, complete, acyclic, ...
- Isomorphism of graphs; subgraphs
- Paths, cycles, and (strong) connectivity
- Euler paths/circuits, Hamiltonian paths (brief)
- Weighted graphs, and shortest paths (Dijkstra's algorithm)
- Bipartite matching: Hall's marriage theorem
- Planar graphs (brief), graph coloring (brief)

Trees

- Rooted and unrooted trees
- Ordered and unordered trees
- (Complete) binary (k-ary) tree
- Subtrees
- Examples in Informatics
- Spanning trees (Kruskal's algorithm, Prim's algorithm.)

Discrete probability

- Discrete (finite or countable) probability spaces
- Events
- Basic axioms of discrete probability
- Independence and conditional probability
- Bayes' theorem
- Random variables
- Expectation; linearity of expectation
- Basic examples of discrete probability distributions birthday paradox, and other subtle examples in probability