Final Review

Discrete Structures
Wednesday, April 28, 2021

You must justify all your answers to recieve full credit

---

1. **Binary Relations.** Properties, functional relations, equivalence and order.
   (a) Given a desription of a binary relation as a bipartite graph, matrix, list, a set-builder expression (or just verbal explanation), find its description in another form.
   (b) Given a binary relation determine if it is a function, an injective, surjective or bijective function.
   (c) Given a description of a binary relation between countable sets, find images of particular values in its domain (including values of functional relations).
   (d) Given some relations find which pairs belong to them.
   (e) Given a binary relation determine if it is reflexive, symmetric, antisymmetric or transitive.
   (f) Given a relation, check if it is an equivalence relation, get the underlying set's partition into equivalence classes (sometimes find one representative in every equivalence class).
   (g) Given a relation, check if it is a partial or total order relation.
2. **Manipulating Relations.** Closures, n-ary relations and relational algebra.
   (a) Given some relations or functions, find their composition (represent with bipartite graphs or as matrix multiplications).
   (b) Given a binary relation on a single set, compute its powers.
   (c) (Warshall Algorithm) Given a binary relation, compute its transitive closure; show the steps.
   (d) Given a relation, find its reflexive, symmetric, transitive closure (also multiple closures).
   (e) Given n-ary relations, apply the 6 relational algebra operations; show relation tables or just determine their size.
   (f) Given n-ary relations, compute inner join, left outer join, right outer join and full outer join of relations.
3. **Number Theory.** Divisibility, GCM and LCM, congruences, multiplicative inverses, exponentiation.
   (a) Given an arithmetic progression find when will it repeat (modulo m).
   (b) Given two positive integers a, b, solve their Bezout identity.
   (c) Given m and x, compute multiplicative inverse $\overline{x}$ modulo m. (Also check that it exists.)
   (d) Given a set of 2 or 3 mutual primes and a system of congruences (plus a set of inverses for them), write a solution for the system using Chinese Remainder Theorem.

(e) Given a prime number p and a number a not divisible by p, check if a is a primitive root or find when $a^k \equiv 1 \pmod{p}$; also check congruences involving powers and use Little Fermat theorem.

(f) Given a primitive root a modulo p (and a list of its powers), solve some congruences involving powers (inverses, roots and/or discrete logarithms).

4. **Recurrent Sequences.** Proving periodicity, 1st and 2nd order recurrences, divide-and-conquer recurrences, Master theorem.

   (a) Given a definition for a recurrent sequence, prove some property (such as congruence) by induction or use periodicity arguments.

   (b) Given a definition for a recurrent sequence and a closed formula, prove the correctness of its closed formula.

   (c) Given a 1st order non-homogeneous recurrence, solve it.

   (d) Given a 2nd order homogeneous recurrence, solve it. (Assume that the characteristic equation has two real roots or one double root, but no complex roots.)

   (e) Given a word problem (strings following some rules, the Tower of Hanoi, tilings, etc.) build a recurrence and/or solve it.

   (f) Given a divide-and-conquer type algorithm, write the recurrence for its time complexity and solve with Master theorem.

5. **Counting.** Permutations, combinations, binomial coefficients, inclusion-exclusion, pigeonhole principle.

   (a) Given a word problem, map bijectively to a set expression (including unions, Cartesian products, etc.) and count variants using the product, sum, difference rules.

   (b) Given a set of restrictions and symmetries inherent for the task, count variants using also the division rule.

   (c) Given a selection task, count variants using combinations and permutation formulas with or without repetition.

   (d) Given a polynomial, find coefficients using binomial and multinomial rules.

   (e) Given a word problem, estimate the "worst case" using the pigeonhole principle.

   (f) Given a word problem, count variants using inclusion-exclusion principle.

6. **Probability.** Single events, multiple events, complements, independence, conditional probability, Bernoulli trials.

   (a) Define the sample space of an experiment, describe the events and compute their probabilities using Laplace's definition.

   (b) Compute probabilities of derived events (complementary, intersection, union, etc.).

   (c) Analyze the probabilities of the outcomes of a probabilistic 2-player game (such as Monty Hall).

   (d) Identify where Bayes' theorem should be applied and apply it.

7. **Random Variables.** Expected value, variance, distributions, independence, Chebyshev's inequality.

   (a) Identify the geometric, binomial, and Bernoulli distributions.

(b) Given a problem description, define random variables for its probabilistic model, define its distribution as a function.

(c) Given a distribution for a discrete random variable X, compute $E(X)$ and $V(X)$.

(d) Given a random variable X, estimate probabilities of X being in some interval by Chebyshev's inequality.

8. **Graphs.** Vertex, edge, subgraph, (un)directed, simple, regular, bipartite, connectedness, path, cycle, complement, isomorphisms.

   (a) Given a property of a graph (edge / vertex number, regularity), estimate the number of graphs with that parameter.

   (b) Given a special graph (complete, cycle, wheel, n-cube, complete bipartite), check some property of the graph.

   (c) Given a graph, check if it is bipartite, complete, or connected.

   (d) Justify whether or not a graph has a particular subgraph (such as a path, cycle, complete graph, etc)

   (e) Convert between different representations of graphs (pair of sets, diagram, adjacency list, adjacency matrix).

   (f) Given a tree, check the condition for a Euler circuit (or path) and find it, if it exists.

   (g) Given two graphs prove or disprove they are isomorphic.

9. **Trees.** Leaf, root, height, traverse trees, BFS, DFS.

   (a) Given some parameters of a rooted tree (count of vertices, internal vertices, leaves, edges, height), estimate other parameters.

   (b) Given a (full and/or balanced) n-ary tree and some parameters, estimate other parameters.

   (c) Convert between different representations of a tree (diagram, syntax tree, prefix / infix / postfix notation, preordered / inordered / postordered vertices with children)

   (d) Given an undirected graph, do a DFS and BFS traversal, indicating all steps.