Final Exam Review

Last Updated: December 10th, 2024

Algorithm Fundamentals

Definitions. computational problem, problem instance, size parameters, big-O notation (O, Ω, Θ, o, ω), recurrence relation, divide-and-conquer recurrence (uniform versus non-uniform), recursion trees (sub-problem trees versus complexity trees), optimal substructure, memoization.

Algorithms. Questions to ask: what is the computational problem that is solved by this algorithm? What is the "big-picture" idea of how the algorithm works? What is its worst-case running time? What conditions cause the worst-case to occur? How to prove that the algorithm always returns a correct answer?

Fundamental Algorithmic Paradigms. Divide and Conquer, Greedy, Dynamic Programming, Mapping Reductions

Divide and Conquer Algorithms. Mergesort, Quicksort, Find-Statistic, Strassen's algorithm, Karatsuba's Integer Multiplication algorithm, Maximum Subsequence Sum, and Minimum Positive Subsequence Sum, Fast Fourier Transform. Questions to ask: what is the divide-and-conquer recurrence associated with the algorithm?

Greedy Algorithms. Kruskal, Prim, and Dijkstra's algorithms, Unit Task Scheduling (UTS), Task Selection, Fractional Knapsack, Fuel Reloading. Questions to ask: how can a data structure be used to improve the algorithm's running time?

Data Structures for Greedy Algorithms. Disjoing Set data structure (Kruskal and UTS), Binary Heap (Prim and Kruskal).

Dynamic Programming Algorithms. 0-1 Knapsack, Edit Distance, Longest Common Subsequence, Optimal Binary Search Tree, Single-source Shortest and Longest Paths in an Acyclic Graph, Floyd-Warshall, Traveling Salesperson. Questions to ask: what is the dynamic-programming recurrence associated with the algorithm.

Substitution Method. inductive assumption, substitution procedure

Theorems. Master Theorem

Complexity Theory

Definitions. Turing reducibility, polynomial-time Turing reducibility, oracle, oracle query, mapping reducibility, polynomial-time mapping reducibility, problem domain, computational complexity theory, decision problem, optimization problem, problem instance, positive/negative instances, size parameter, P, NP, NP-completeness, P =? NP open problem, predicate function

Turing Reductions. Multiply $\leq_T$ Add, 2SAT $\leq_T^p$ Reachability

Logic Problems and Related Definitions. SAT, 2SAT, 3SAT, Boolean formula, clause, implication, contrapositive, variable, literal, consistent/inconsistent literal set, variable assignment, satisfiable, unsatisfiable

Mathematical Problem Domains. Sets, Logic, Graph Theory, Number Theory

Three Kinds of Mapping Reductions. Embeddings, Contractions, Interdomain

Examples of Mapping Reductions.

- Even $\leq_m^p$ Odd
- Independent Set $\leq_m^p$ Clique
- Clique $\leq_m^p$ Independent Set
- Independent Set $\leq_m^p$ Vertex Cover
- Set Partition $\leq_m^p$ Subset Sum
- Hamilton Path $\leq_m^p$ LPath
- Subset Sum $\leq_m^p$ Set Partition
- Vertex Cover $\leq_m^p$ Half Vertex Cover
- Clique $\leq_m^p$ Half Clique
- SAT $\leq_m^p$ 3SAT
- 3SAT $\leq_m^p$ Clique
- 3SAT $\leq_m^p$ Subset Sum

- 3SAT $\leq_m^p$ Directed Hamilton Path
- Directed Hamilton Path $\leq_m^p$ Hamilton Path
- Hamilton Path $\leq_m^p$ Hamilton Cycle
- Hamilton Cycle $\leq_m^p$ Traveling Salesperson
- Max Matching $\leq_m^p$ Max Flow

Notable Problems in P. Reachability, 2SAT, Prime, Distances in Graphs, Boolean Formula Evaluation, Max Flow, Max Matching

How to Prove that a problem is in NP? certificate set, certificate, valid certificate

How to Prove that an NP problem is NP-complete? NP-completeness reductions

Notable NP-complete problems. SAT, 3SAT, Clique, Half Clique, Independent Set, Subset Sum, 3-Dimensional Matching, Set Partition, Vertex Cover, Half Vertex Cover, Directed Hamilton Path, Hamilton Path, Hamilton Cycle, Traveling Salesperson

Theorems. Cook's Theorem, Transitivity of both Turing and Mapping reducibility

Practice Problems

1. Which divide-and-conquer recurrence best describes the running time of Strassen's algorithm.
   (a) $T(n) = 4T(n/2) + n^3$
   (b) $T(n) = 8T(n/2) + n^2$
   (c) $T(n) = 7T(n/2) + n^2$
   (d) $T(n) = 2T(n/2) + n$
2. The polynomial-time mapping reduction from Maximum Matching (in a bipartite graph) to Maximum Flow in a network is most efficiently computable in ____ time with respect to the size of the bipartite graph.
   (a) cubic
   (b) quadratic
   (c) log-linear
   (d) linear
3. All of the following problems have been established to be solvable in worst-case polynomial time, *except* for

(a) finding a minimum vertex cover in a simple graph

(b) finding a maximum spanning tree in a weighted graph

(c) finding a maximum matching in a bipartite graph

(d) determining if an instance of `2SAT` is satifiable

4. Based on the `NP`-completeness reductions provided in lecture, proving that `TSP` is `NP`-complete relies on each of the following reductions except for

   (a) `3SAT` $\leq_m^p$ `Directed Hamilton Path`

   (b) `3SAT` $\leq_m^p$ `Clique`

   (c) `SAT` $\leq_m^p$ `3SAT`

   (d) `Hamilton Path` $\leq_m^p$ `Hamilton Cycle`

5. In computer science, an oracle is defined as

   (a) an entity that is capable of proving answers to instances of some specified problem.

   (b) any company that develops database software.

   (c) an entity that is capable of providing instructions for any URM program.

   (d) an entity that is capable of computing a universal function.

6. The mapping reduction from `Independent Set` to `Clique` is an example of a(n) ____ reduction.

   (a) embedding

   (b) contraction

   (c) interdomain

   (d) none of the above