1. (a) Define the minimum cost spanning tree(MST).

(b) Describe the Kruskal’s Algorithm and give the runtime of that.

(c) Prove the correctness of Kruskal’s Algorithm.

(d) What is the strategy for the Kruskal’s algorithm.

Ref: CS300, 2017 Spring

1. (a) What is greedy strategy?

(b) Give some examples where greedy doesn’t work.

Ref: CS300, 2017 Spring

1. (a) State the UNION-FIND Problem.

(b) Give the way to implement UNION-FIND and way of improvement.

(c) Prove that the way is improved way.

(d) Give the runtime of improved algorithm.

(e) Suggest improvement of UNION-FIND using “path-compression”

Ref: CS300, 2017 Spring

1. Briefly define alphabet, character, strings(or words), length, empty word, prefix, suffix, substring.

Ref: CS300, 2017 Spring

1. State the brute forcing algorithm for string matching and the runtime of that.

Ref: CS300, 2017 Spring

1. (a) Describe Rabin-Karp Algorithm.

(b) Give the runtime of Rabin-Karp Algorithm.

(c) What can be problem for the Rabin-Karp Algorithm?

(d) Give the remedy to solve the problem and the problem of the remedy and solve again it.

(e) Give the runtime of improved Rabin-Karp Algorithm.

Ref: CS300, 2017 Spring

1. (a) Describe Boyer-Moore Algorithm.

(b) Give the runtime of Boyer-Moore Algorithm.

(c) Suggest “Suffix Heuristic” for Boyer-Moore Algorithm. Explain how it works.

Ref: CS300, 2017 Spring

1. (a) Define Edit Distance.

(b) Describe some properties of Edit Distance.

(c) Give the algorithm for finding Edit Distance.

Ref: CS300, 2017 Spring

1. (a) Define Trie, data structure for String.

(b) Give the way of implementing vertices.

(c) Describe the properties of Trie.

Ref: CS300, 2017 Spring

1. Give algorithm and state the runtime of search operation of Trie.

Ref: CS300, 2017 Spring

1. Give algorithm and state the runtime of insert operation of Trie.

Ref: CS300, 2017 Spring

1. Give algorithm and state the runtime of delete operation of Trie.

Ref: CS300, 2017 Spring

1. (a) Define PARTRICIA-Tree

(b) Define Suffix Tree

(c) Give the way and runtime for construction.

(d) Give the space complexity for construction.

Ref: CS300, 2017 Spring

1. Give the way and runtime for efficient string matching using suffix tree.

Ref: CS300, 2017 Spring

1. Give the way and runtime for searching for many strings in one text using suffix tree.

Ref: CS300, 2017 Spring

1. Give the way and runtime for searching one string P in many texts

Ref: CS300, 2017 Spring

1. Give the way and runtime for searching for the longest common substring in

Ref: CS300, 2017 Spring

1. (a) Why is data compression important?

(b) State two types of data compression.

(c) Define ‘code’ for text compression and property of that.

(d) Give some avoidance when we define code.

Ref: CS300, 2017 Spring

1. (a) Describe Huffman Coding.

(b) How to encode and decode in Huffman Coding?

(c) Give the runtime for construction and algorithm technique.

(d) Give the runtime for encoding and decoding.

(e) Why is Huffman Coding optimal?

Ref: CS300, 2017 Spring

1. (a) What is decision problem?

(b) Define Problem in general.

Ref: CS300, 2017 Spring

1. (a) Define P and give some example of P.

(b) Define NP and give some example of NP.

Ref: CS300, 2017 Spring

1. Why and Is P=NP?

Ref: CS300, 2017 Spring

1. State the satisfiability problem. (SAT Problem)

Ref: CS300, 2017 Spring

1. State the Clique problem.

Ref: CS300, 2017 Spring

1. State the Vertex Cover problem.

Ref: CS300, 2017 Spring

1. State the Partition problem.

Ref: CS300, 2017 Spring

1. State the Traveling Salesman problem (TSP)

Ref: CS300, 2017 Spring

1. (a) State the coPrimes problem.

(b) Is coPrimes NP?

Ref: CS300, 2017 Spring

1. (a) Define Graph Isomorphism.

(b) Define .

Ref: CS300, 2017 Spring

1. (a) Define SUBSET SUM.

(b) Define PARTITION.

(c) PARTITOIN SUBSET SUM.

Ref: CS300, 2017 Spring

1. (a) Define VERTEX COVER

(b) Define CLIQUE

(c) VERTEX COVER CLIQUE.

(d) Prove that G has a Vertex Cover V’ of size k if and only if has a clique of size n-k.

Ref: CS300, 2017 Spring

1. (a) Prove that if A B and , then

(b) Prove that if A B and , then

Ref: CS300, 2017 Spring

1. Prove that is transitive, i.e., if and then

Ref: CS300, 2017 Spring

1. (a) State CNF form of SAT.

(b) State k-SAT Problem.

(c)

(d) Prove that if is satisfiable if and only if G has a clique of size k, k is the number of clauses of

Ref: CS300, 2017 Spring

1. (a) Define NP-Hard

(b) Define NP-Complete

(c) Give some remark for some relationship of NP-Hard Problems.

(d) What is directed proved problem as NP-Hard?

Ref: CS300, 2017 Spring

1. (a) State the Optimization Problem.

(b) How to solve the optimization problem?

Ref: CS300, 2017 Spring

1. (a) Give the state of the UNION-FIND data structure after the following sequence of operations, staring from singleton sets . Use path compression. In case of ties, always make the lower numbered root point to the higher numbered one.

(b) Suppose that starting from the singleton partition of some set first some UNION- and then some FIND-Operations are carried out, altogether m operations. Suppose that union-by-height and path-compression is used. Show that the amortized runtime per operation is , i.e., the total runtime of these operations is   
Hint: Let k be the number of times the parent pointer of some vertex is accessed if it is pointing to the root of some tree. Find an upper bound on k? How often is a parent pointer accessed otherwise before it is pointing to the root again.

Ref: 8th Homework Assignment of CS300, prof. Helmut Alt, 2017 Spring

1. (a) Suppose that a long straight stretch of a highway has to be equipped with mobile phone base stations. The company has investigated that possible positions for stations are at kilometers from the beginning of the highway stretch. No point of the highway should be more than 5 km away from the nearest station. Give an algorithm to ﬁnd the minimum number of stations to achieve that goal, if possible.

(b) A server has n customers waiting to be served. The service time required by each customer is known in advance: it is minutes for customer i. So if, for example, the customers are served in order of increasing i, then the ith customer has to wait minutes. We wish to minimize the total waiting time . Give an eﬃcient algorithm for computing the optimal order in which to process the customers.

Ref: 8th Homework Assignment of CS300, prof. Helmut Alt, 2017 Spring

1. Consider the brute-force algorithm for string-matching on alphabet of size d. Assume a random text of length n where each character occurs in each position with the same probability 1/d. let be the pattern searched for of length m. Analyze exactly the expected number of comparisons between characters in terms of d, n and m.

Ref: 9th Homework Assignment of CS300, prof. Helmut Alt, 2017 Spring

1. In two-dimensional pattern recognition an -array T and an -array P of characters of an alphabet are given and the task is to find all occurrences of P in T.

(a) Design and analyze a brute-force algorithm.

(b) How would you modify the Rabin-Karp-algorithm to solve this problem? Analyze the runtime in the worst case.

Ref: 9th Homework Assignment of CS300, prof. Helmut Alt, 2017 Spring

1. (a) Consider the following pattern :

babcbaabccababc

Apply the Boyer-Moore algorithm to P and a text of length at least 50 of your choice, where both the bad-character- as well as the good-suffix-rule can be successfully applied at least once.

(b) Give and analyze an algorithm to determine the array used in the algorithm of Boyer-Moore.

Ref: 10th Homework Assignment of CS300, prof. Helmut Alt, 2017 Spring

1. Describe and analyze efficient algorithms for search, insertion, and deletion in PATRICIA-trees. Suppose that edges have the space efficient labelling as shown in class.

Ref: 11th Homework Assignment of CS300, prof. Helmut Alt, 2017 Spring

1. Use suffix trees to design efficient algorithms for the following problems:

(a) Find the longest substrings of a string w that occur more than once.

(b) Find the shortest substrings of w that occur only once.

(c) Find the longest substrings of w which are palindromes, i.e., read the same forward and backward.

Ref: 11th Homework Assignment of CS300, prof. Helmut Alt, 2017 Spring

1. Show that the following problems are in NP. Of which can you show that they are in P?

(a) Given a graph, does it contain a simple cycle of length 4?

(b) Set cover (SC):

Given sets (of integers) and . Are there k of the given sets whose union equals the union of all sets?

(c) Is a given natural number a the product of 2 prime numbers?

Ref: 12th Homework Assignment of CS300, prof. Helmut Alt, 2017 Spring

1. Show the following polynomial time reductions:

(a) subset sum PARTITION

Hint:

(b) Let SGI be the following problem: Given two graphs . Is isomorphic to a subgraph of . Show that SGI where A is one of the NP-problems given in class.

(c) SAT3SAT

Hint: replace any SAT-clause by 3SAT-clauses where are new additional variables.

Ref: 12th Homework Assignment of CS300, prof. Helmut Alt, 2017 Spring

1. A Boolean formula in CNF is obviously satisfiable if or are satisfiable by some truth assignment to . Here, means the formula simplified by omitting clauses containing the literal and taking out the literal from all clauses where it occurs. is defined analogously.

Once some clause contains no more literals the formula is not satisfiable, if the formula is empty (no more clauses) it is satisfiable.

Use these properties to implement a recursive SAT-solver, i.e., a problem, that decides whether the input formula is satisfiable and, if so, produces a satisfying assignement.

Ref: 12th Homework Assignment of CS300, prof. Helmut Alt, 2017 Spring

1. Give True or False for each of the following statements. Justify your answers.

(1) A graph algorithm with running time is asymptotically better than an algorithm with running time for a connected, undirected graph

(2) If the depth-first search of a graph G yields no back edges, then the graph G is acyclic.

(3) If some of the edge weights in a graph are negative, the shortest path from s to t can be obtained using Dijkstra’s algorithm by first adding a large constant C to each edge weight, where C is chosen large enough that every resulting edge weight will be nonnegative.

(4) Let G=(V,E) be a weighted graph and let M be a minimum spanning tree of G. The path in M between any pair of vertices v1 and v2 must be a shortest path in G.

Ref: CS300 Final Sample Problems

1. You have an exam with n questions. Each question I has integral point value that requires minutes to solve. Suppose that no partial credit is awarded in this exam. Your goal is to come up with an algorithm which, given computes the minimum number of minutes required to earn at least C points on the exam.

(1) Let M(i,c) denote the minimum number of minutes needed to earn c points when you are restricted to selecting from questions 1 through i. Give a recurrence expression for M(i, c). (The base cases: for all i, and )

(2) Give an algorithm to compute the minimum number of minutes required to earn at least C points on the exam and analyze the running time.

(3) Explain how to extend your solution from the previous part to output a list S of the questions to solve such that the total score is at least C and the total number of minutes is minimized.

(4) Suppose now that partial credit is given so that the number of points you receive on a question is proportional to the number of minutes you spend working on it. That is, you earn points per minute on question i (up to a total of points), and you can work for fractions of minutes. Give an -time algorithm to determine which questions to solve (and how much time to devote to them) in order to receive C points the fastest. Prove the correctness of your algorithm.

Ref: CS300 Final Sample Problems

1. Given 2 decision problems and in NP and for each of the following statements, give one of T(true), F(false) or O(open question), and briefly justify your answer.

(1) If

(2)

(3)

(4)

(5)

(6) Suppose is solvable in . Then is also solvable in

Ref: CS300 Final Sample Problems

1. Give True or False.

(1) We can use a 2-approximation algorithm for the minimum vertex cover problem as a 2-approximation algorithm for the maximum clique problem.

(2) If one NPC problem can be solved in polynomial time, all of the NPC problem can be solved in polynomial time.

(3) Suppose Q is in NP, but not necessarily NPC. Then, a polynomial time algorithm for 3SAT would necessarily imply a polynomial time algorithm for Q.

(4) NPC problems can be reduced to any other NPC problem.

(5) If SAT problem is in P, then co-NP P.

Ref: CS300 Final Sample Problems

1. Let G=(V,E) be an undirected graph. A strongly independent set is a subset S of vertices such that for any two vertices, there is no path of length 2 between u and v.

Consider the following Strongly Independent Set (SIS) problem:

Given an undirected graph G=(V,E) and an integer k, does G have a strongly independent set of size k?

For each following question, give Yes or No or Unknown.

(1)

(2)

(3)

(4)

(5)

Ref: CS300 Final Sample Problems

1. Suppose that a certain country has the coins with the following denominations (all integers). Given an integer n, we want to find the minimum number of coins to make n. (You can assume that to make any amount n)

(1) Consider the greedy algorithm which repeatedly takes the largest coin possible. For the set of denominations , prove that the greedy algorithm always gives the minimum number of coins.

(2) Give the set of denominations for which the greedy algorithm does not give the minimum number of coins.

(3) Give an efficient algorithm to solve the problem for any set of k different denominations. What is the running time of your algorithm?

Ref: CS300 Final Sample Problems