Final Exam: Practice

Name: ____________

Fall 2019, CSCE629 Analysis of Algorithms — XX/XX/2019

Texas A&M U — Prof. Fang Song

Instructions (please read carefully before start!)

- This exam contains 14 pages (including this cover page) and 4 questions, for the total of 100 points and 0 bonus points.
- You will have two hours (xx:xx - xx:xx) to finish the exam. Be strategic and allocate your time wisely.
- You may bring two double-sided letter size (8.5-by-11″) study sheet. Any other resources and electronic devices are NOT permitted.
- Your work will be graded on correctness and clarity. Make sure your hand writing is legible.
- Don't forget to write your name on the top.

Grade Table (for instructor use only)

Question	Points	Score
1	30	
2	25	
3	20	
4	25	
Total:	100	

1. *Short answers*. Joker and Batman agreed to give up violence and compete peacefully on algorithmic problems. Answer the following, and **always** justify your answers briefly.

   (a) (5 points) Joker goes first. He gives n equal-sized intervals, and asks for the maximum number of compatible intervals. Batman proposes a greedy algorithm that picks the intervals based on earliest start time. Is Batman's algorithm correct?

   (b) (5 points) Batman's turn. He describes a *directed acyclic graph* where the edges have arbitrary weights and ask Joker to find the simple path from node s to node t of *maximum* total weight. Joker decides to negate the weight on each edge and running a shortest path algorithm. Is Joker's algorithm correct?

   (c) (5 points) In this round they change the rule. Each will make a statement and the other needs to decide if the statement is true or false. Joker states that if all capacities in a flow network are integers, then the value of the maximum flow must also be an integer. Help Batman tell true or false.

(d) (5 points) With your help, Batman won on last question. Now he makes the following statement. If a graph has a unique shortest path P from node s to node t, and has a unique minimum spanning tree T, then every edge in P must also be in T. Help Joker this time.

(e) (5 points) In the last round, Joker challenges Batman with a tax-collecting task at the city council of Gotham, and asks Batman to resign forever unless he manages to design an $O(n^2 \log n)$ algorithm. Batman comes up with a divide-and-conquer algorithm where he can divide a problem into two subproblems both of size half of the original, and the combining step takes $O(n^2)$ time. Does Batman need to resign?

(f) (5 points) Likewise, Batman wants to stop Joker from crimes, and Joker agrees not to commit any crime as long as Batman's question keeps him busy. After thinking hard for a while, Batman tells Joker the 3-SAT problem, the perfect bipartite matching problem, and the notion of Karp reductions. Batman asks Joker to find a Karp reduction from 3-SAT to perfect bipartite matching. Given the state-of-art, how long do you think that Joker will be kept busy?

2. (Network flow)

   (a) (8 points) Given a maximum (s,t)-flow for a flow network G, how to find a minimum (s,t)-cut and how much time does it take?

   (b) (5 points) Suppose G is a directed graph with integer edge capacities. Let (A,B) be a minimum (s,t)-cut. Prove or disprove: if G' is obtained by adding 1 to the capacity of every edge in G, then (A,B) is a minimum (s,t)-cut in G'.

(c) (12 points) You are given maximum (s,t)-flow f for a flow network G. Subsequently, the capacity of a specific edge $e^* \in G$ is reduced by 1.

i) Given an algorithm that finds the maximum flow in the resulting graph G' in $O(m+n)$ time. You may assume the flow f is *acyclic*: there is no cycle in G on which all edges carry positive flow.

ii) Prove the correctness and the $O(m+n)$ running time of your algorithm.

3. (Coin changing) Consider the problem of making change for n cents using the fewest number of coins. Assume that each coin's value is an integer.

   (a) (5 points) Prof. Greedy proposes an algorithm that always uses coins of the largest value as many as possible and then moves to the next largest one. Prove that if the coins available consist of quarters (25 cents), dimes (10 cents), nickels (5 cents), and pennies (1 cent), then Prof. Greedy's algorithm yields an optimal solution.

   (b) (5 points) Consider other sets of coin denominations. Does this algorithm always yield an *optimal* solution? We assume the set always contains pennies so that there is a solution for every value of n.

(c) (10 points) Give an $O(nk)$-time algorithm that makes change for any set of k different coin denominations. Again we assume pennies are always included.

4. (k-spanning tree) In a (minimum) spanning tree of a graph, it could occur that some vertex is a bottleneck in the sense that its degree is high. In *k-spanning tree* problem, we intend to keep the degrees low in a spanning tree. More precisely, we are given an undirected graph $G = (V, E)$, and the goal is to decide if there is a spanning tree in G in which each vertex has degree at most k.

   (a) (7 points) Show that k-spanning tree is in NP for any $k \geq 2$.

   (b) (10 points) Show that 2-spanning tree is NP-complete. [Hint: consider the relation with Hamiltonian cycle.]

(c) (8 points) Show that, in fact, k-spanning tree is NP-complete for any $k \geq 2$.

Scratch paper – no exam questions here.

Scratch paper – no exam questions here.

Guideline on grading rubrics

We will refer to these rubrics for certain types of questions. They are general guidelines and are subject to change. They are stated on a 10-point basis, and will be scaled accordingly.

Dynamic Programming

- **6 points for a correct recurrence**, described either using functional notation or as pseudocode for a recursive algorithm.
  - + 1 point for a clear English description of the function you are trying to evaluate. (Otherwise, we don't even know what you're trying to do.) Automatic zero if the English description is missing.
  - + 1 point for stating how to call your recursive function to get the final answer.
  - + 1 point for the base case(s).
  - + 3 points for the recursive case(s).
- 4 points for iterative details
  - + 1 point for describing the memoization data structure; a clear picture may be sufficient.
  - + 2 points for describing a correct evaluation order; a clear picture may be sufficient. If you use nested loops, be sure to specify the nesting order.
  - + 1 point for running time
- Proofs of correctness are not required for full credit on exams, unless the problem specifically asks for one.
- For problems that ask for an algorithm that computes an optimal structure—such as a subset, partition, subsequence, or tree, an algorithm that computes only the value or cost of the optimal structure is sufficient for full credit, unless the problem says otherwise.

Graph reductions For problems solved by reducing them to a standard graph algorithm covered in class (for example: shortest paths, topological sort, minimum spanning trees, maximum flows, bipartite maximum matching):

- 1 point for listing the vertices of the graph. (If the original input is a graph, describing how to modify that graph is fine.)
- 1 point for listing the edges of the graph, including whether the edges are directed or undirected. (If the original input is a graph, describing how to modify that graph is fine.)

- 1 point for describing appropriate weights and/or lengths and/or capacities and/or costs and/or demands and/or whatever for the vertices and edges.
- 2 points for an explicit description of the problem being solved on that graph. (For example: "We compute the shortest path in G from u to v.")
- 3 points for other algorithmic details, assuming the rest of the reduction is correct.
  + 1 point for describing how to build the graph from the original input (for example: "by brute force").
  + 1 point for describing the algorithm you use to solve the graph problem (for example: "Dijstra's algorithm")
  + 1 point for describing how to extract the output for the original problem from the output of the graph algorithm.
- 2 points for the running time, expressed in terms of the original input parameters.
- If the problem explicitly asks for a proof of correctness, divide all previous points in half and add 5 points for proof of correctness.

NP-Complete reductions For problems that ask "Prove that X is NP-Complete":

- **4 points for the polynomial-time reduction**:
  + 1 point for explicitly naming the NP-hard problem Y to reduce from. You may use any of the problems listed in the lecture notes; a list of NP-hard problems will appear on the back page of the exam.
  + 2 points for describing the polynomial-time algorithm to transform arbitrary instances of Y into inputs to instances of black-box algorithm for X
  + 1 point for describing the polynomial-time algorithm to transform the output of the black-box algorithm for X into the output for Y.
  + Reductions that call a black-box algorithm for X more than once are acceptable. You do not need to analyze the running time of your resulting algorithm for Y, but it must be polynomial in the size of the input instance of Y.
- **6 points for the proof of correctness**. This is the entire point of the problem. These proofs always have two parts; for example:
  + 3 points for proving that your reduction transforms positive instances of Y into positive instances of X.
  + 3 points for proving that your reduction transforms negative instances of Y into negative instances of X.
  + These proofs do not need to be as detailed as in the book or homework; however, it must be clear that you have at least considered all possible cases. We are really just looking for compelling evidence that you understand why your reduction is correct.

  + It is still possible to get partial credit for an incorrect reduction. For example, if you describe a reduction that sometimes reports false positives, but you prove that all False answers are correct, you would still get 3 points for half of the correctness proof.

- **2 points for showing** $X \in \mathbf{NP}$, if this has not been shown in a previous problem. Relocate 1 point each from the reduction and the correctness proof.
- Zero points for reducing X to some NP-hard problem Y.
- Zero points for attempting to solve X.