McGill

April 2018
Final Examination

Algorithm Design

COMP 360 SEC 001

2:00 PM April 27, 2018

EXAMINER:	Hamed Hatami	ASSOC. EXAMINER:	Adrian Vetta

STUDENT NAME:		**McGILL ID:**									

INSTRUCTIONS

EXAM:	CLOSED BOOK ☒ OPEN BOOK ☐
	SINGLE-SIDED ☒ PRINTED ON BOTH SIDES OF THE PAGE ☐
	MULTIPLE CHOICE ☐ Note: The Examination Security Monitor Program detects pairs of students with unusually similar answer patterns on multiple-choice exams. Data generated by this program can be used as admissible evidence, either to initiate or corroborate an investigation or a charge of cheating under Section 16 of the Code of Student Conduct and Disciplinary Procedures. ANSWER IN BOOKLET ☐ EXTRA BOOKLETS PERMITTED: YES ☒ NO ☐ ANSWER ON EXAM ☒
	SHOULD THE EXAM BE: RETURNED ☒ KEPT BY STUDENT ☐
CRIB SHEETS:	NOT PERMITTED ☐ PERMITTED ☒ e.g. one 8 1/2X11 handwritten double-sided sheet Specifications: One handwritten double-sided letter size sheet
DICTIONARIES:	TRANSLATION ONLY ☐ REGULAR ☒ NONE ☐
CALCULATORS:	NOT PERMITTED ☒ PERMITTED (Non-Programmable) ☐
ANY SPECIAL INSTRUCTIONS:	

1	2	3	4	5	6	7	**Total**
/15	/15	/15	/15	/10	/15	/15	**/100**

15

1. Write a maximization linear program in two variables such that the boundary of its feasible region is the square with vertices $(-1,0),(0,1),(1,0),(0,-1)$, and whose optimal solution has value 5 and is given only by the point $(0,-1)$.

   Solution:

$$\begin{aligned} \text{max} \quad & 5x_2 \\ \text{s.t.} \quad & x_1 + x_2 \leq 1 \\ & x_1 + x_2 \geq -1 \\ & x_1 - x_2 \geq -1 \\ & x_1 - x_2 \leq 1 \end{aligned}$$

   Scheme: 3 points for each of the 5 parts in LP. Deduct 3 points if they write $x_1, x_2 \geq 0$.

2. Either prove that the following problem is NP-complete or show that it belongs to P by giving a polynomial time algorithm:

   - Input: An undirected graph G, two nodes s, t and a positive integer k.
   - Question: Is there a path of length at least k between s and t?

   Solution: The problem is NP-complete. To show that it is in NP note that an efficeint certifier can take a path in the graph and check to see if it is of length at least k and starts at s and ends at t.

   To prove completeness, we can reduce the Hamiltonian Path problem to this. Given an input G to the hamiltonian path problem, we can run the oracle for the above problem for $k = n$, and all choices of s and t and $s \neq t$. If the oracle returns yes to any of these instances then we know that G has a Hamiltonian path and otherwise, we know that it does not.

   Scheme: 0 if they say the problem is in P. 2 points for proving that the problem is in NP. Another 3 points for mentioning that one needs to reduce Hamiltonian path to this problem, and the other 10 points for the details of the reduction.

15

3. Either prove that the following problem is NP-complete or show that it belongs to P by giving a polynomial time algorithm:

   - Input: A flow network and a positive integer k.
   - Question: Is there a maximum flow f such that the total sum of the flow on all of the edges is at most k? That is $\sum_{e \in E} f(e) \leq k$, and f is a maximum flow?

   Solution: The problem is in P. We have seen in the class that the maximum flow problem can be formulated as a linear program. Solve this linear program, and then solve it again but with the extra constraint $\sum_{e \in E} f(e) \leq k$. If the solutions match then the answer is YES, otherwise it is NO.

   Scheme: 1 or 2 Points if they just say that they find a max-flow and see if the condition $\sum_{e \in E} f(e) \leq k$ is satisfied. 10 Points if they mention linear programming. 13 Points if the just say that they solve the linear program with the new constraint (without comparing it to the original solution). If they try to solve it using Ford-Fulkerson via shortest path augmentations 5/15.

15

4. Either prove that the following problem is NP-complete or show that it belongs to P by giving a polynomial time algorithm:

   - Input: An undirected graph G.
   - Question: Does G have a proper colouring with three colours R, G, B that assigns the colour B to exactly one vertex?

   Solution: See Homework 5.

10

5. Given a set P of n points on the plane, consider the problem of finding the smallest circle containing all the points in P. Show that the following is a 2-factor approximation algorithm for this problem. Pick a point x in P, and set r to be the distance of the farthest point in P from x. Output the circle centered at x with radius r.

   Solution: Let y be the farthest point from x. Any circle that contains both x and y (including the optimal solution) has radius at least $\mathrm{dist}(x, y)/2 = r/2$, thus the algorithm is a 2-factor algorithm.

   Scheme: 2 to 3 points if they say the worst case is when all the points are in the circle whose center is in the middle of x and y without any proof.

6. Consider the triangle elimination problem. We are given an undirected graph $G = (V, E)$, and want to find the smallest possible set of vertices $U \subseteq V$ such that deleting these vertices removes all the triangles (i.e. cycles of length 3) from the graph. Prove that one of the following algorithms is a 3-factor approximation algorithm, and the other one is not.

   Algorithm I:

   - While there is still a triangle C left in G:
   - Delete all the three vertices of C from G
   - EndWhile
   - Output the set of the deleted vertices

   Algorithm II:

   - While there is still a triangle C left in G:
   - Delete one arbitrarily chosen vertex of C from G
   - EndWhile
   - Output the set of the deleted vertices

   Solution: See Homework 5.

7. In the MAX-CUT problem, given an undirected graph G we want to partition the vertices of G into two parts (A, B) such that the number of edges between A and B is maximized. Prove that the following is a $\frac{1}{2}$-factor approximation algorithm for this problem:

   - Let $v_1, \ldots, v_n$ be all the vertices of G.
   - Initially set $A = B = \emptyset$.
   - For $i = 1, \ldots, n$ do
   - If v_i has more neighbours in A than in B then add v_i to B, otherwise add v_i to A.
   - EndFor

   Solution: Let α_i be the number of edges that go between A and B when the vertex v_i is added to one of the parts, and similarly let β_i be the edges that go inside A or B when v_i is added. The algorithm guarantees that $\alpha_i \geq \beta_i$. Note that in the end the total number of edges crossing between A and B is $\sum_{i=1}^{n} \alpha_i$, and the total number of edges is $\sum_{i=1}^{n} \alpha_i + \beta_i$ We have

$$\text{Output} = \sum_{i=1}^{n} \alpha_i \geq \sum_{i=1}^{n} \frac{\alpha_i + \beta_i}{2} = |E|/2 \geq \frac{\text{Opt}}{2}.$$

   Scheme: 4 points if they mention that the optimal is at most $|E|$ and that the algorithm picks at least half of the edges. 5 to 7 points if they say that more than half the neighbours of every vertex is going to be in the other part (which is not true). At least 10 points if they define α_i and β_i as above.