UNIVERSITY OF TORONTO
Department of Computer and Science

December 2013 Examinations

CSC373H1 F

Instructor: Kaveh Ghasemloo

Duration - 3 hours

No Aids Allowed.

PLEASE COMPLETE THE SECTION BELOW AND THE SECTION BEHIND THIS PAGE:

First Name: ______________________________

Last Name: ______________________________

Exam Instructions

- **Check that your exam book has 32 pages** (including this cover page and 5 blank pages at the end). The last 5 pages are for rough work only, *they will* **not** *be marked.* Please bring any discrepancy to the attention of an invigilator.
- There are 12 questions worth a total of 180 points. Answer all questions on the question booklet. You need to get at least 40% in this exam to pass the course.
- In question 9–12, if you don't know the answer you can leave the question blank *and* write "I DON'T KNOW." to receive 20% of the points of the question.
- Please read all questions right now and ask any clarification questions you have during the first 60 minutes to minimize distractions to other students.

Course Specific Notes

- Unless stated otherwise, you can use standard data structures and algorithms discussed in CSC263 and CSC373 by simply stating their standard name (e.g. min-heap, merge-sort, Dijkstra) without describing their implementation or proving their properties. If you modify a data structure or an algorithm, you must describe the modifications and their effects.
- In some questions you will be given a computational problem and you should design an efficient algorithm for it. Unless stated otherwise, for data structures and algorithms that you design you should provide a short high-level explanation of how your algorithm works in plain English, *and* the pseudo-code for your algorithm in a style similar to those we have seen in the lectures. If you miss any of these the answer might not be marked. Your answers will be marked based on the efficiency of your algorithms and the clarity of your explanations. State the running time of your algorithm with a brief argument supporting your claim and prove that your algorithm works correctly (e.g. finds an optimal solution).

1	2	3	4	5	6	7	8	9	10	11	12

PLEASE PRINT YOUR STUDENT NUMBER AND YOUR NAME

Student Number: ____________

First Name: ____________

Last Name: ____________

The section below is for marker's use only. Do NOT use it for answering or as scratch paper.

Question #	Score/Points
1	/ 20
2	/ 20
3	/ 10
4	/ 10
5	/ 10
6	/ 10
7	/ 10
8	/ 10
9	/ 20
10	/ 20
11	/ 20
12	/ 20
Total	/ 180

A. External View of Algorithms

1. **Computational Problems** [20]

   Define the following computational problems. State what is given as their input and what is their output.

   For each problem state if (the decision problem corresponding to) the problem is in P or if it is in NP-complete. Do not justify your answers.

   a. Shortest Path (between two vertices):

   b. Max Flow:

The question continues on the next page.

The question continues here.

c. Integer Programming:

d. Clique:

e. SAT:

2. Algorithms

a. Fill in the following table for algorithms. For each algorithm provide a good asymptotic upper-bound on its worst-case running time and state the name of the problem solved by the algorithm. [12]

Algorithm	Running Time	Problem
Dijkstra		
Bellman-Ford		
Floyd-Warshall		
Ford-Fulkerson		
Edmonds-Karp		
Simplex		

The question continues on the next page.

The question continues here.

[4] b. Explain the essential ideas behind the greedy algorithm design paradigm.
(When can we use the greedy paradigm? What are the important steps in designing a greedy algorithm? What makes greedy algorithms efficient?)

[4] c. Explain the essential ideas behind the dynamic programming algorithm design paradigm.
(When can we use the dynamic programming paradigm? What are the important steps in designing a dynamic programming algorithm? What makes dynamic programming algorithms efficient?)

3. **Complexity** [10]

Define the following terms:

a. Polynomial-time black-box reductions ($\leq_T^{\mathsf{P}}$):

b. Polynomial-time many-one reductions ($\leq_m^{\mathsf{P}}$):

The question continues on the next page.

The question continues here.

c. P:

d. NP:

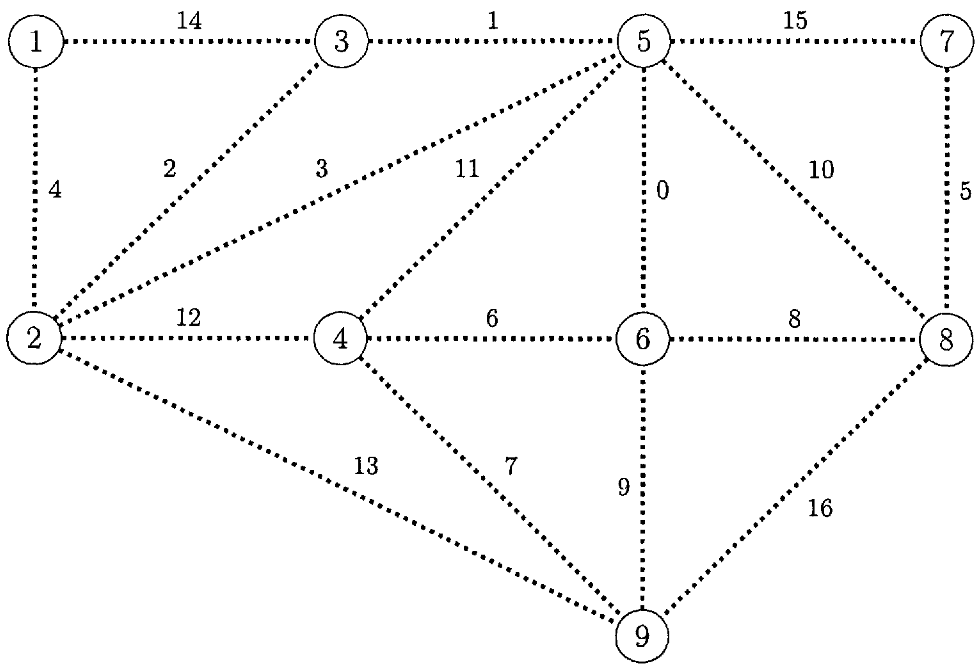
e. NP-complete:

B. Basic Understanding

4. **MST: Prim Algorithm** [10]

The following undirected graph has 17 edges, and the edge weights are the consecutive integers $0, 1, 2, 3, \ldots, 16$. Execute the Prim algorithm on this graph.

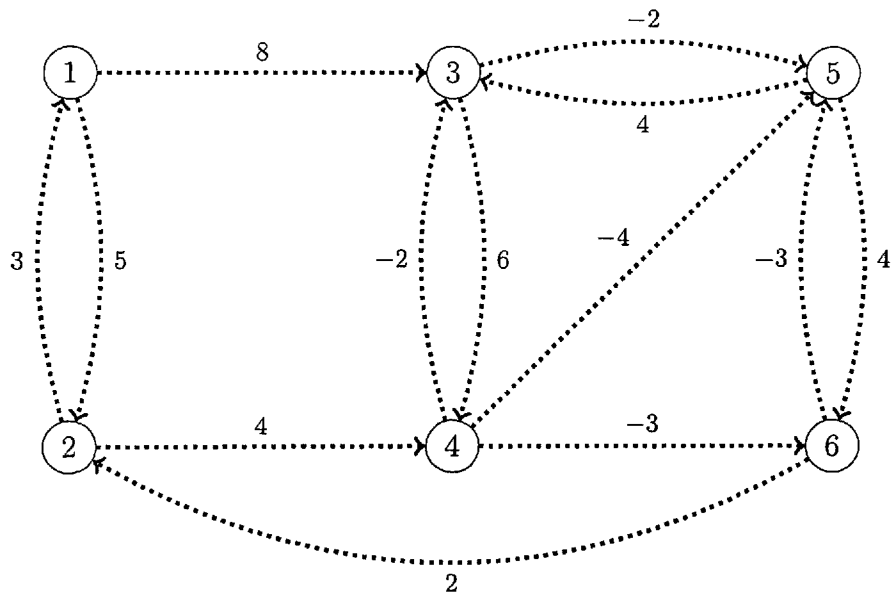
- Thicken the edges that belong to the MST.
- List the edges of the MST in the order they are found by the algorithm.



[10] 5. **Shortest Paths: Bellman-Ford**

Compute the shortest paths from vertex 1 in the following graph using the Bellman-Ford algorithm.

- Thicken the edges that belong to the shortest paths from vertex 1 on the graph.
- What is the *distance* of the vertex 5 from vertex 1?



You can use the following table for your computations.

	1	2	3	4	5	6
0	0	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$
1						
2						
3						
4						
5						
6						

You may continue your solution on the next page.

Continue your solution here.

[10] 6. **Huffman Code**

What is the optimal Huffman code for the following set of frequencies?

Draw the tree corresponding to the optimal code.

Symbol	a	b	c	d	e	f	g
Frequency	29	15	26	15	23	17	35

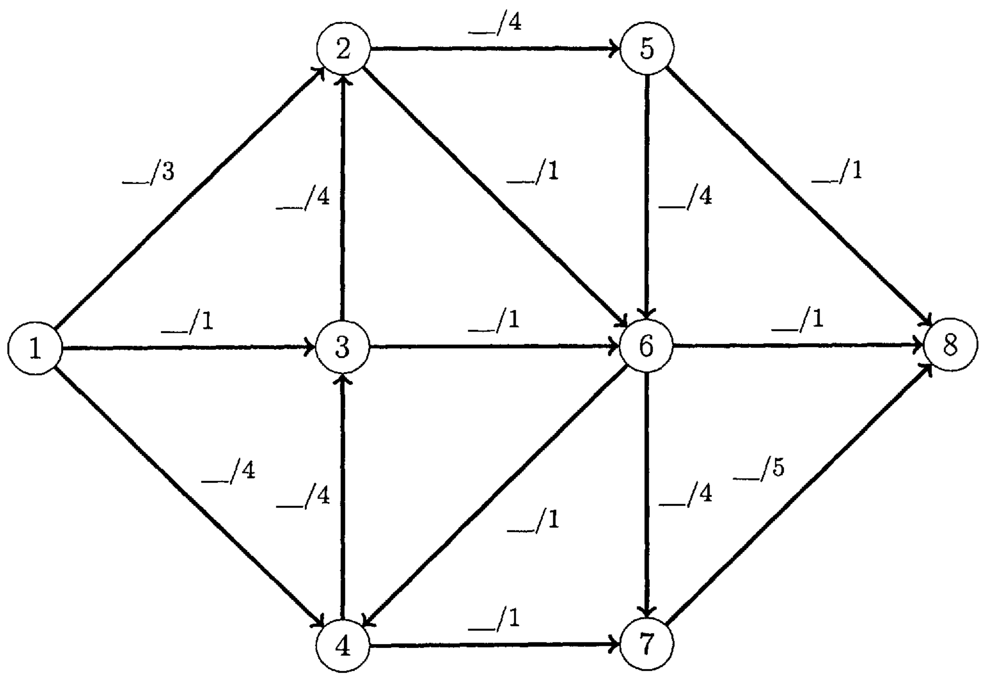
You may continue your solution on the next page.

Continue your solution here.

[10] 7. **Max-Flow/Min-Cut**

Consider the network G given below. Let $s = 1$ and $t = 8$.

- Find a max-flow f from s to t in G. Write the amount of flow on each edge.
- Find a min-cut (S, T) in G. Draw a line around S.



You may continue your solution on the next page.

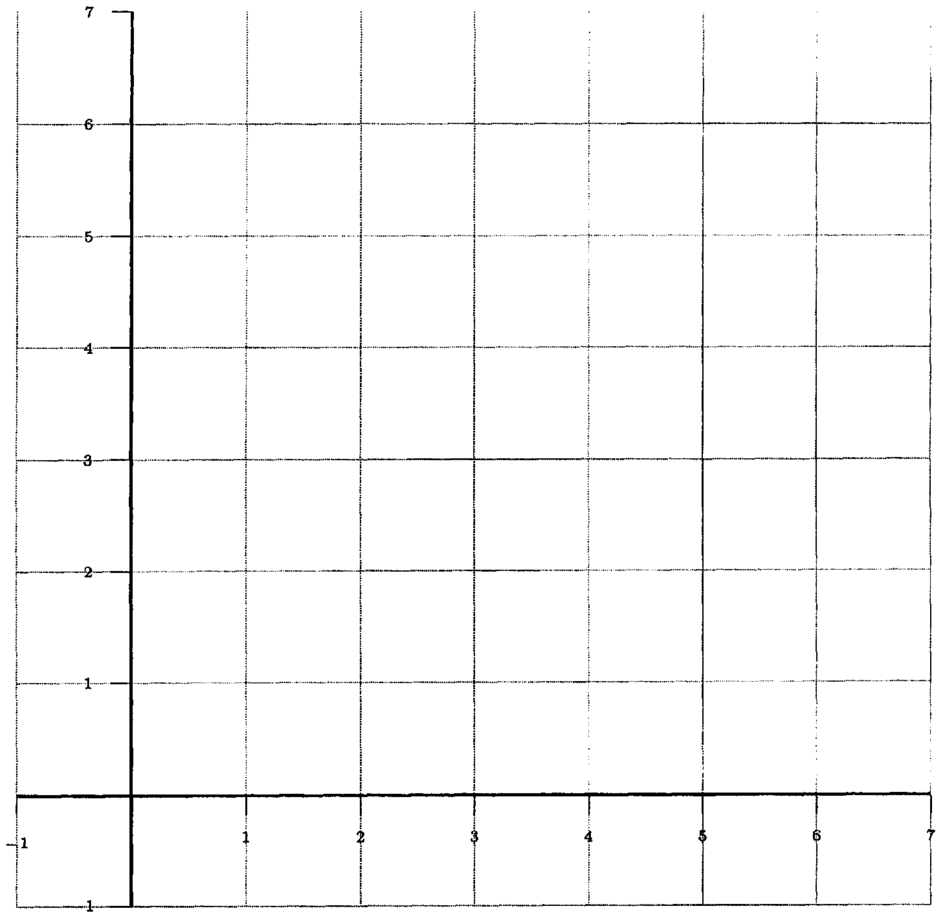
Continue your solution here.

8. **Linear Programming**

Consider the following linear program:

- variables: x, y
- constraints: I. $0 \leq x, y$ II. $-2x + y \leq 0$ III. $x - 2y \leq 1$ IV. $x + 2y \leq 5$
- objective function: $\max 2x + 3y$

[6] a. Draw the lines corresponding to the constraints.
Dash the region corresponding to the feasible solutions.



The question continues on the next page.

The question continues here.

b. The optimal answer is achieved in the intersection of two lines. Which constraints correspond to those lines? What is the optimal value of the objective function? [4]

C. Designing Algorithms

[20] 9. **Patrick The Gold Collector**

Patrick likes to play the following game. There is a 2 by n board. Each square on the board contains some amount of gold. Patrick has a pawn that starts at the left side of the table and has to cross to the other side. The goal is to collect the maximum possible amount of gold.

The amount of gold in each square is given in a 2 by n table $T[1..2, 1..n]$. At each turn the pawn can move one square to the right. The move can be horizontal or diagonal. In other words, if the pawn is at position $(1, i)$ it can move either to square $(1, i+1)$ or square $(2, i+1)$. In the first move the pawn can move either to $(1, 1)$ or $(2, 1)$.

Input: The table $T[1..2, 1..n]$.

Output: The maximum amount of gold that Patrick's pawn can collect.

Example:

Input:

T	1	2	3	4
1	5	4	10	14
2	6	10	8	11

Output: 40

Explanation:
The following moves given $40 = 6 + 10 + 10 + 14$:
Start at $(2, 1)$, then move horizontal, horizontal, diagonal, horizontal.

a. Design an algorithm to solve this problem. Briefly explain the high-level idea behind your algorithm and provide its pseudo-code.

The question continues on the next page.

The question continues here.

b. Give a good upper-bound on the worst-case running time of your algorithm. Briefly justify your answer.

c. Briefly argue that your algorithm correctly solves the problem.

[20] 10. **Hospitable Steve**

Steve is an experienced sailor. He has invited some of his friends to his yacht. All guests are standing on the deck in a row to watch the sunset. Steve knows that some of his guests do not get along with each other. He decides to invite some of the guests to see the interior of his yacht so the remaining guests will get along with the people next to them. He wants to minimize the number of the guests he has to invite.

You task in this question is to help Steve find the minimum number of people he has to invite.

Input: An n by n matrix A where $A[i,j]$ is 1 iff the ith and jth guests in the row do not get along.

Output: The minimum number of guests to be invited so the remaining guests will get along with the people next to them.

Example:

Input: $n = 4$, $A = \begin{array}{c|cccc|} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 0 & 1 & 1 \\ 2 & 0 & 0 & 1 & 0 \\ 3 & 1 & 1 & 0 & 0 \\ 4 & 1 & 0 & 0 & 0 \\ \hline \end{array}$ Output: 1

Explanation: If we remove 3, the guests in the row will be $1, 2, 4$. The guests 1 and 2 get along as well as guests 2 and 4.

a. Design an algorithm to solve this problem. Briefly explain the high-level idea behind your algorithm and provide its pseudo-code.

The question continues on the next page.

The question continues here.

The question continues on the next page.

The question continues here.

b. Give a good upper-bound on the worst-case running time of your algorithm. Briefly justify your answer.

c. Briefly argue that your algorithm correctly solves the problem.

You may continue your solution on the next page.

Continue your solution here.

D. Applications & Reductions

[20] 11. **Integer Program for Minimum Spanning Tree**

a. Express the minimum spanning tree problem as an integer program.

b. Briefly argue that your reduction is correct.

You may continue your solution on the next page.

Continue your solution here.

E. Complexity Theory

[20] 12. **NP-completeness of Directed Hamiltonian Path Problem**

Recall that a Hamiltonian path is a path where every vertex in the graph appears exactly once in it. The dHamPath problem is defined as follows:

Input: A directed graph G.

Output: YES if there is a Hamiltonian path in G, NO otherwise.

Let st-dHamPath be the following decision problem:

Input: A directed graph G and two distinct vertices s and t.

Output: YES if there is a Hamiltonian path from s to t in G, NO otherwise.

Assume that st-dHamPath is NP-hard.

a. Show that dHamPath is in NP by providing a polynomial-time certificate verifier and a polynomial bound on the size of the certificates.

b. Briefly argue for the correctness of your verifier algorithm.

The question continues on the next page.

The question continues here.

c. Show that dHamPath is NP-hard by providing a polynomial-time many-one reduction algorithm from st-dHamPath to dHamPath.

d. Briefly argue for the correctness of your reduction algorithm.

This page is for rough work only, it will **not** be graded.

More space for rough work is available on the next page.

This page is for rough work only, it will **not** be graded.

More space for rough work is available on the next page.

This page is for rough work only, it will **not** be graded.

More space for rough work is available on the next page.

This page is for rough work only, it will **not** be graded.

More space for rough work is available on the next page.

This page is for rough work only, it will **not** be graded.