COS 226 Algorithms and Data Structures Spring 2003

Final

This test has 11 questions worth a total of 84 points. You have 150 minutes. The exam is closed book, except that you are allowed to use a one page cheatsheet. No calculators or other electronic devices are permitted. Give your answers and show your work in the space provided. **Write out and sign the Honor Code pledge before turning in the test.**

"I pledge my honor that I have not violated the Honor Code during this examination."

Problem	Score
1	
2	
3	
4	
5	
6	
Sub 1	

Problem	Score
7	
8	
9	
10	
11	
Sub 2	

Total	

Name:

Login ID:

Precept:

1	12:30	Kevin
2	1:30	Adriana
4	3:30	Kevin

1. **Analysis of algorithms. (6 points)**

   Precisely define what it means for an algorithm to have *worst-case* running time $O(N^2)$ where N is the size of the input.

2. **String searching. (4 points)**

   This question is about the problem of searching for any occurrence of an M-character pattern in an N-character text string. Suppose that student X uses the brute-force method and student Y uses the right-left scan. The students are each given the option of picking from among one of the three string generators listed below for both programs to be tested on (both pattern and text to be generated from the same generator, then both programs invoked on the same input).

   A. random string of As and Bs

   B. random ASCII characters

   C. all As except last character is B

   Fill in the blanks to give the best choice and expected results from each students' point of view. (If the asymptotic number of character compares is the same, answer "1".)

   (a) For X's choice _____, brute-force is a factor of _____ faster than right-left.

   (b) For Y's choice _____, right-left is a factor of _____ faster than brute-force.

3. **Geometric algorithms. (8 points)**

The computer graphics in *Matrix Reloaded* require fast geometric algorithms for various operations dealing with polygons. Let A and B be two simple polygons, represented by their counter-clockwise ordering of their vertices. Let N be the total number of vertices For simplicity, you may assume that there are no degeneracies, e.g., A and B do not share any vertices, and there are no 3 vertices that are collinear.

(a) Give a *brief* description of a fast algorithm for determining whether A lies completely within B, and circle the choice below that best describes its worst-case performance. (For full credit, you need to describe an optimal algorithm.)

i. $\log N$ ii. $\sqrt{N}$ iii. N iv. $N \log N$ v. N^2

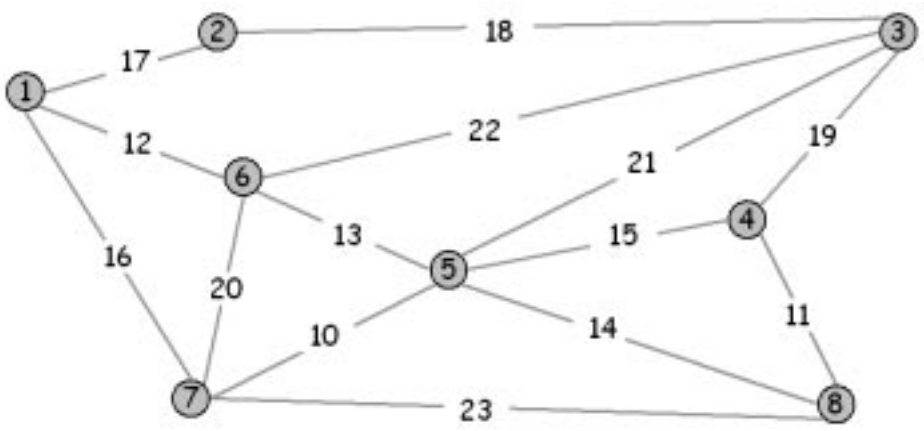
You may use any of the subroutines from lecture and the book, including:

Subroutine	Primitive	Running time
S1	do 2 line segments intersect?	1
S2	do N line segments intersect?	$N \log N$
S3	is point inside simple N-sided polygon?	N
S4	are 2 points on the same side of a line?	1
S5	convex hull of N points	$N \log N$
S6	Voronoi diagram of N points	$N \log N$

(b) Repeat (a) when A and B are convex.

4. **Minimum spanning tree. (8 points)**

Consider the following undirected network with edge weights as shown.

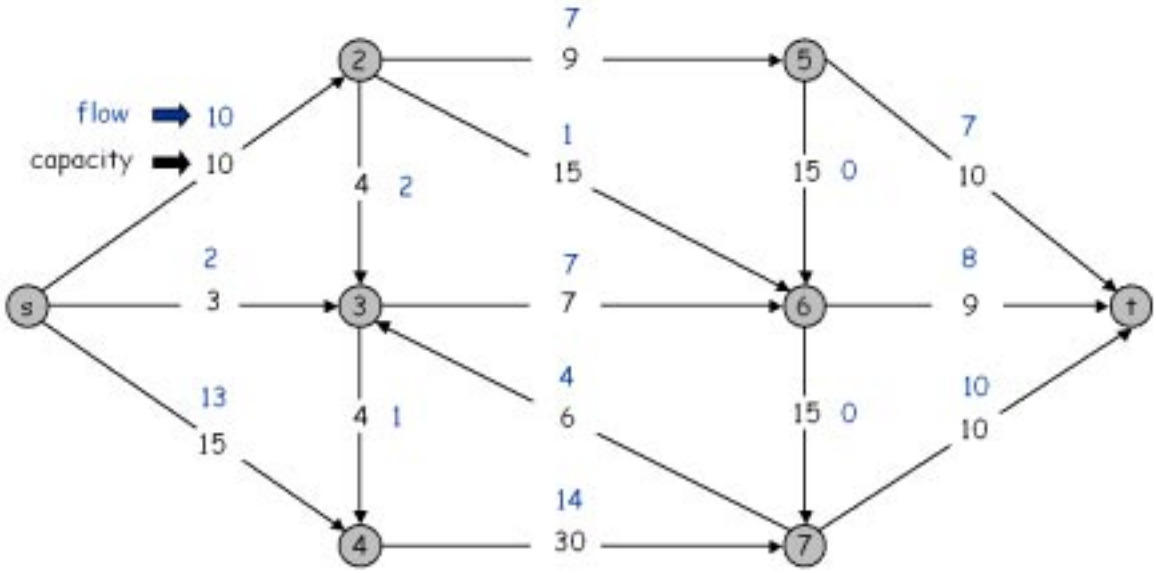


(a) List the *edges* in the MST in the order that Prim's algorithm chooses them. Start Prim's algorithm from vertex 1.

(b) List the *edges* in the MST in the order that Kruskal's algorithm selects them.

5. **Max flow, min cut. (10 points)**

Starting from the following flow (printed above or to the right of the capacities), perform one iteration of the Ford-Fulkerson algorithm. Choose the *fattest* augmenting path, i.e., the path with the highest residual capacity.



(a) Write down the fattest augmenting path.

(b) What is the value of the resulting flow?

(c) Is the resulting flow optimal? If so, give a min cut whose capacity is equal to the value of the flow. If not, give a fattest augmenting path.

6. **Data compression. (6 points)**

   Consider the following 21 character message that consists of 3 a's, 7 c's, 6 t's, and 5 g's.

   ```
   a a c c c c a c t t g g g t t t t c c g g
   ```

   Are the following 43 bits a possible Huffman encoding of the message above?

   ```
   00000011110001010100100100101010101110010 01
   ```

   Justify your answer as concisely and rigorously as possible.

7. **Algorithmic design. (8 points)**

   Given an undirected network with V vertices, $10V$ edges, and integer edge weights between 1 and V, describe how to find a MST asymptotically faster than $V \log V$. What is the worst-case running time of your algorithm.

8. **Theory vs. practice. (12 points)**

   For each of the following pairs, indicate which algorithm has the better worst-case running time and which one will likely perform better in practice on real-world inputs.

(a) Sorting:	I. Insertion sort	____ Worst-case
	II. Mergesort	____ Practice
(b) Sorting:	I. Quicksort	____ Worst-case
	II. Mergesort	____ Practice
(c) Min spanning tree:	I. Prim with binary heap	____ Worst-case
	II. Prim with Fibonacci heap	____ Practice
(d) Priority queue:	I. Binary search tree	____ Worst-case
	II. Binary heap	____ Practice
(e) Linear programming:	I. Ellipsoid algorithm	____ Worst-case
	II. Simplex algorithm	____ Practice
(f) Convex hull:	I. Package wrap	____ Worst-case
	II. Graham scan	____ Practice

9. **Choosing the right algorithms and data structures. (8 points)**

Circle the best answer to each of the following questions.

(a) What is the primary reason to use Floyd's algorithm to solve the all-pairs shortest path problem instead of Dijkstra's algorithm?

i. Faster for dense graphs
ii. Faster for sparse graphs
iii. Implementing Fibonacci heaps is difficult
iv. Can reconstruct the path itself in addition to the length of the shortest path

(b) What is the primary reason to use the Burrows-Wheeler data compression algorithm over LZW?

i. Uses less memory
ii. Better compression ratio
iii. Faster encoding
iv. Faster decoding

(c) Describe a situation when you would need to use breadth-first search instead of depth-first search.

i. Find a Euler tour
ii. Find a directed cycle
iii. Find a path between two vertices in an undirected graph
iv. Find a path between two vertices with the fewest number of edges

(d) Describe a situation when you would need to use the Bellman-Ford shortest path algorithm instead of Dijkstra's algorithm.

i. Compute routing directions in a GPS-based car navigation system
ii. Detect arbitrage opportunity in currency exchange
iii. Compute the shortest path between two vertices in an *undirected* graph
iv. Create a table that gives the lengths of the shortest routes that connect *all pairs* of cities

10. **Linear programming. (6 points)**

You are operating an iron foundry and need to produce 1,000 pounds of castings that contain at least 0.45% percent manganese and between 3.25% and 5.50% percent silicon. You have pure manganese and three types of pig iron available in essentially unlimited amounts, with the following properties and cost per thousand pounds:

	Manganese	Pig A	Pig B	Pig C
Silicon	0.00%	4.00%	1.00%	0.60%
Manganese	100.00%	0.45%	0.50%	0.40%
Cost	$8000	$26	$30	$20

Blending together various raw materials works as you'd expect: if you melt 1 pound of pig iron A with 1 pound of pig iron B, the resulting mixture is 2.5% silicon and 0.475% manganese.

Formulate (but do not solve) a linear program to determine the minimum cost way to produce 1,000 pounds of castings. Use the following variables:

M = thousands of pounds of manganese used
A = thousands of pounds of pig iron A used
B = thousands of pounds of pig iron B used
C = thousands of pounds of pig iron C used

It's not necessary to put it into standard form.

11. **Reductions. (8 points)**

Given an *undirected* network with positive edge weights, the *global min cut problem* is to find a subset of vertices S such that the sum of the weight of the edges with one endpoint in S and one endpoint not in S is as small as possible. Give a polynomial reduction from the global min cut problem to the max flow problem on *directed* networks.

Consider the following two statements.

A There exist a linear time algorithm for the max flow problem.

B There exists a linear time algorithm for the global min cut problem.

Given a polynomial reduction from the global min cut problem to the max flow problem, which one or more of the following cannot be ruled out?

i. A and B are both true.

ii. A is true but B is false.

iii. A is false but B is true.

iv. A and B are both false.