CSE 5311

Spring 2003

Test 1

2.

4.

9.

Closed Book Questions

Multiple Choice - WRITE your answer to the LEFT of each problem. 4 points each.

- The reason for marking nodes in a Fibonacci heap is:
 - A. to allow computing the value of the potential function.
 - B. to assure that the structure is a Fibonacci heap rather than a binomial heap.
 - C. to improve the performance of CONSOLIDATE.
 - D. to indicate nodes that have lost a child since becoming a child themselves.
 - Which of the following statements is true?

A. A binary search tree may be assigned legal AVL tree balance factors if and only if it can be legally colored as a red-black tree.

- B. If a binary search tree may be assigned legal AVL tree balance factors, then it can be legally colored as a red-black tree.
- C. If a binary search tree can be legally colored as a red-black tree, then it may be assigned legal AVL tree balance factors.
- D. No binary search tree can be both assigned legal AVL tree balance factors and be legally colored as a red-black tree.
- Which property does not hold for binomial heaps? 3.
 - A. DECREASE-KEY takes O(1) time.
 - B. MINIMUM takes O(lg n) time.
 - C. Performing n INSERT operations into an empty heap will take O(n) time.
 - D. The number of trees is based on the binary representation of the number of stored items.
 - Which algorithm uses Union-Find trees?

A. Dijkstra's

- B. Kruskal's
- C. Prim's
- D. The MST technique based on Warshall's algorithm.
- Assuming that |E| > |V|, the time for Prim's algorithm using a Fibonacci heap is: 5.
 - A. O(|E|)
 - B. $O(|E| \log |E|)$
 - C. $O(|E| \log |V|)$
 - D. $O(|V| \log |V| + |E|)$
- Brent's rehash technique is used to improve the successful search performance of 6
 - A. (separate) chaining
 - B. double hashing
 - C. linear probing
 - D. optimal hashing
- 7. The perfect hashing method discussed in class may be viewed as improving which technique?
 - A. (separate) chaining
 - B. double hashing
 - C. linear probing
 - D. optimal hashing
- 8. Which of the following properties would not hold for the potential function for a data structure?
 - A. Changes in potential are used in computing the amortized cost of an operation.
 - B. The potential can become indefinitely large, even though the number of items stored is fixed.
 - C. The potential is never negative.
 - D. The potential is not stored by an implementation.
 - Which of the following is not true regarding self-adjusting lists?
 - A. The count method records the number of requests for each item and approximates the optimal fixed ordering.
 - B. The move-to-front method always brings the requested item to the beginning of the list.
 - C. The OPT method is efficient and is a good substitute for the move-to-front method.

D. The transpose method exchanges the requested item with the item before it unless the requested item is already at the beginning of the list.

- The master method is most closely related to this analysis technique. 10.
 - A. Divide-and-conquer
 - B. Hiring problem
 - C. Recursion-tree
 - D. Substitution
- 11. Suppose a gambling game involves a sequence of rolls from a standard six-sided die. A player wins \$1 when the value rolled is
- the same as the previous roll. If a sequence has 601 rolls, what is the expected amount paid out? (5 points) 12. Suppose a gambling game involves a sequence of rolls from a standard six-sided die. A player wins \$1 when the value rolled is
- larger than the previous roll. If a sequence has 601 rolls, what is the expected amount paid out? (5 points)

Name

Test 1

Open Book Questions

- 1. Use the master method to derive an asymptotic bound on $T(n) = T(n/2) + n^2$. (10 points)
- 2. a. Suppose that the root of a tree in a binomial heap has 4 children. What is the minimum number of nodes that may appear in the tree? (3 points)

b. Suppose that the rightmost path of a leftist heap has 4 nodes. What is the minimum number of nodes that may appear in the heap? (5 points)

c. Suppose that the root of a tree in a Fibonacci heap has 4 children. What is the minimum number of nodes that may appear in the tree? (7 points)

3. Does the following graph have a unique minimum spanning tree? Trace the contents of the disjoint set structure during the execution of the modified Kruskal's algorithm that checks for uniqueness. (Galler-Fischer representation, i.e. parent pointers, need **<u>not</u>** be shown.) (15 points)



4. Consider the following information for constructing an optimal binary search tree.

$\begin{array}{llllllllllllllllllllllllllllllllllll$	n=6;	
$\begin{array}{llllllllllllllllllllllllllllllllllll$	q[0]=0.01;	Building c(0,2) using roots 1 thru 2
$ \begin{array}{llllllllllllllllllllllllllllllllllll$	key[1]=10;	Building c(1,3) using roots 2 thru 3
$\begin{array}{llllllllllllllllllllllllllllllllllll$	p[1]=0.2;	Building c(2,4) using roots 3 thru 4
$\begin{array}{llllllllllllllllllllllllllllllllllll$	q[1]=0.02;	Building c(3,5) using roots 4 thru 5
$\begin{array}{llllllllllllllllllllllllllllllllllll$	key[2]=20;	Building c(4,6) using roots 5 thru 6
$\begin{array}{llllllllllllllllllllllllllllllllllll$	p[2]=0.1;	Building c(0,3) using roots 1 thru 3
$\begin{array}{llllllllllllllllllllllllllllllllllll$	q[2]=0.03;	Building $c(1,4)$ using roots 3 thru 3
p[3]=0.2; Building c(3,6) using roots 4 thru 6 g[3]=0.04; Building c(0,4) using roots 3 thru 4 p[4]=0.2; Building c(1,5) using roots 3 thru 4 p[4]=0.6; Building c(0,5) using roots 3 thru 3 key[5]=50; Building c(1,6) using roots 3 thru 4 g[5]=0.06; Building c(1,6) using roots 3 thru 4 g[5]=0.04; Counts - root trick 29 without root trick 50 key[6]=60; Average probe length is ? p[6]=0.07; trees in parenthesized prefix c(0,0) cost 0.00000 c(1,1) cost 0.00000 w[0][1]=0.230000 c(4,4) cost 0.00000 w[0][1]=0.360000 c(5,5) cost 0.00000 w[0][1]=0.500000 c(5,6) cost 0.00000 w[1][1]=0.230000 c(1,2) cost 0.150000 w[1][1]=0.20000 c(4,4) cost 0.240000 w[1][1]=0.20000 c(4,4) cost 0.240000 w[1][1]=0.290000 c(2,3) cost 0.100000 w[1][1]=0.290000 c(3,5) cost 0.100000 w[1][1]=0.60000 c(4,5) cost 0.10000 w[1][1]=0.290000 c(3,5) cost 0.10000 w[1][1]=0.290000 c(4,2) cost 0.10000 w[1][1]=0.290000 c(3,5) cost 0.10000 w[1][1]=0.290000	key[3]=30;	Building c(2,5) using roots 3 thru 4
g[3]=0.04; Building c(1,5) using roots 2 thru 3 key[4]=0.2; Building c(2,6) using roots 3 thru 4 g[4]=0.0; Building c(2,6) using roots 3 thru 4 g[5]=0.06; Building c(1,6) using roots 3 thru 4 p[5]=0.04; Building c(0,6) using roots 7 thru 7 g[6]=0.07; Counts - root trick 29 without root trick 50 key[6]=60; Average probe length is ? p[6]=0.07; trees in parenthesized prefix g[6]=0.03; c(0,0) cost 0.00000 w[0][1]=0.230000 c(2,2) cost 0.000000 w[0][2]=0.360000 c(3,3) cost 0.00000 w[0][1]=0.230000 c(4,4) cost 0.000000 w[0][1]=0.20000 c(4,4) cost 0.000000 w[0][1]=0.20000 c(4,4) cost 0.00000 w[0][1]=0.20000 c(5,5) cost 0.00000 w[1][1]=0.20000 c(2,3) cost 0.22000 w[1][1]=0.20000 c(2,3) cost 0.240000 w[1][1]=0.190000 c(4,5) cost 0.10000 50 w[1][1]=0.20000 c(4,5) cost 0.10000 50 w[1][1]=0.20000 c(2,4) cost 0.30000 10(,20) w[1][1]=0.20000 c(5,5) cost 0.140000 60 w[1][1]=0.20000 c(4,6) cost 0.30000 30(20,) w[1][1]=0.200000 <t< td=""><td>p[3]=0.2;</td><td>Building c(3,6) using roots 4 thru 6</td></t<>	p[3]=0.2;	Building c(3,6) using roots 4 thru 6
$\begin{array}{llllllllllllllllllllllllllllllllllll$	q[3]=0.04;	Building c(0,4) using roots 2 thru 3
$ \begin{array}{llllllllllllllllllllllllllllllllllll$	key[4]=40;	Building c(1,5) using roots 3 thru 4
$\begin{array}{llllllllllllllllllllllllllllllllllll$	p[4]=0.2;	Building c(2,6) using roots 4 thru 4
key[5]=50; Building c(1,6) using roots 3 thru 4 q[5]=0.06; Building c(0,6) using roots 3 thru 4 q[5]=0.04; Counts - root trick 29 without root trick 50 key[6]=60; Average probe length is ? g[6]=0.03; c(0,0) cost 0.00000 w[0][1]=0.230000 c(1,1) cost 0.000000 w[0][1]=0.360000 c(2,2) cost 0.000000 w[0][1]=0.360000 c(3,3) cost 0.000000 w[0][1]=0.4600000 c(4,4) cost 0.000000 w[0][1]=0.30000 c(5,5) cost 0.000000 w[0][1]=0.20000 c(6,6) cost 0.230000 10 w[1][1]=0.30000 c(3,2) cost 0.270000 30 w[1][1]=0.390000 c(3,2) cost 0.270000 30 w[1][1]=0.390000 c(4,5) cost 0.100000 50 w[1][1]=0.390000 c(4,5) cost 0.100000 50 w[1][1]=0.390000 c(4,5) cost 0.100000 50 w[1][1]=0.390000 c(3,4) cost 0.540000 30(20,1) w[1][1]=0.390000 c(4,5) cost 0.100000 50 w[1][1]=0.290000 c(4,6) cost 0.30000 00 w[1][1]=0.290000 c(4,5) cost 0.40000 40 w[1][1]=0.390000 c(4,6) cost 0.30000 00 w[1][1]=0.290000 c(4,6) cost 0.	q[4]=0.0;	Building c(0,5) using roots 3 thru 3
p[5]=0.06; Building c(0,6) using roots ? thru ? q[5]=0.04; Counts - root trick 29 without root trick 50 key[6]=0.07; trees in parenthesized prefix q[6]=0.03; c(0,0) cost 0.00000 w[0][1]=0.230000 c(1,1) cost 0.00000 w[0][2]=0.36000 c(3,3) cost 0.00000 w[0][1]=0.230000 c(3,3) cost 0.00000 w[0][2]=0.36000 c(3,3) cost 0.00000 w[0][1]=0.20000 c(6,6) cost 0.00000 w[0][1]=0.80000 c(6,6) cost 0.00000 w[0][1]=0.20000 c(6,6) cost 0.23000 10 w[1][1]=0.15000 c(2,3) cost 0.27000 30 w[1][1]=0.39000 c(3,4) cost 0.240000 40 w[1][1]=0.39000 c(5,6) cost 0.14000 50 w[1][1]=0.39000 c(1,2) cost 0.51000 10(,20) w[1][1]=0.39000 c(2,4) cost 0.510000 10(,20) w[1][1]=0.30000 c(3,5) cost 0.440000 40(3,60) w[1][2]=0.33000 c(3,5) cost 0.440000 40(50,1) w[2][2]=0.03000 c(2,4) cost 0.71000 30(,40) w[1][2]=0.15000 c(2,5) cost 0.440000 40(3,50) w[1][2]=0.47000 c(3,5) cost 0.440000 40(50,1) w[2][2]=0.33000 c(3,6) cost 0.74000 40(30,60(50,1)) w[2][3]=0.40000	key[5]=50;	Building c(1,6) using roots 3 thru 4
g[5]=0.04; Counts - root trick 29 without root trick 50 key[6]=60; Average probe length is ? g[6]=0.07; trees in parenthesized prefix g[6]=0.03; c(0,0) cost 0.00000 w[0][1]=0.230000 c(1,1) cost 0.000000 w[0][1]=0.360000 c(3,3) cost 0.000000 w[0][1]=0.600000 c(4,4) cost 0.000000 w[0][4]=0.80000 c(5,5) cost 0.000000 w[0][5]=0.90000 c(6,6) cost 0.000000 w[0][1]=0.220000 c(1,2) cost 0.150000 20 w[1][1]=0.20000 c(3,4) cost 0.240000 40 w[1][1]=0.590000 c(3,4) cost 0.240000 40 w[1][1]=0.590000 c(4,5) cost 0.14000 50 w[1][1]=0.290000 c(1,2) cost 0.510000 10 w[1][1]=0.390000 c(3,4) cost 0.540000 30(20,) w[1][1]=0.290000 c(4,5) cost 0.140000 60 w[1][1]=0.290000 c(2,4) cost 0.510000 10(,20) w[1][1]=0.290000 c(3,2) cost 0.540000 30(20,) w[1][1]=0.290000 c(2,4) cost 0.710000 30(,40) w[1][1]=0.270000 c(2,2) cost 0.740000 40(,50,) w[1][1]=0.270000 c(2,2) cost 0.740000 40(,50,) w[1][1]=0.270000 c(2,2) cost 0.740000 30(20,) w[1][1]=0.470000	p[5]=0.06;	Building c(0,6) using roots ? thru ?
key[5]=60; Average probe length is ? p[6]=0.07; trees in parenthesized prefix q[6]=0.03; c(0,0) cost 0.00000 w[0][1]=0.10000 c(1,1) cost 0.00000 w[0][1]=0.230000 c(2,2) cost 0.000000 w[0][3]=0.60000 c(3,3) cost 0.000000 w[0][1]=0.90000 c(4,4) cost 0.000000 w[0][6]=1.00000 c(6,6) cost 0.000000 w[0][6]=1.00000 c(6,6) cost 0.000000 w[1][2]=0.150000 c(2,3) cost 0.230001 10 w[1][1]=0.290000 c(3,4) cost 0.230000 10 w[1][1]=0.120000 c(3,4) cost 0.240000 40 w[1][1]=0.150000 c(3,4) cost 0.240000 40 w[1][1]=0.150000 c(3,4) cost 0.240000 40 w[1][1]=0.590000 c(4,5) cost 0.100000 50 w[1][1]=0.590000 c(5,6) cost 0.140000 60 w[1][1]=0.270000 c(2,4) cost 0.71000 30(20,) w[2][2]=0.30000 c(2,5) cost 0.44000 40(,50) w[2][1]=0.270000 c(2,5) cost 0.44000 40(,50) w[2][1]=0.270000 c(2,5) cost 0.74000 30(20,40) w[2][1]=0.270000 c(2,5) cost 0.74000 40(,30,50) w[2][1]=0.270000 c(2,5) cost 0.74000 40(,30,50	q[5]=0.04;	Counts - root trick 29 without root trick 50
$\begin{array}{llllllllllllllllllllllllllllllllllll$	key[6]=60;	Average probe length is ?
$ \begin{array}{llllllllllllllllllllllllllllllllllll$	p[6]=0.07;	trees in parenthesized prefix
	q[6]=0.03;	c(0,0) cost 0.000000
	w[0][0]=0.010000	c(1,1) cost 0.000000
w[0][2]=0.360000 c(3,3) cost 0.00000 w[0][4]=0.80000 c(4,4) cost 0.00000 w[0][5]=0.90000 c(5,5) cost 0.000000 w[0][6]=1.00000 c(0,1) cost 0.23000 10 w[1][1]=0.02000 c(1,2) cost 0.150000 20 w[1][1]=0.02000 c(2,3) cost 0.270000 30 w[1][2]=0.15000 c(2,3) cost 0.270000 30 w[1][4]=0.59000 c(4,5) cost 0.100000 50 w[1][5]=0.69000 c(1,2) cost 0.510000 10(,20) w[1][6]=0.79000 c(1,3) cost 0.71000 30(20,) w[2][2]=0.03000 c(1,3) cost 0.71000 30(20,) w[2][4]=0.47000 c(3,5) cost 0.440000 40(,50) w[2][5]=0.57000 c(4,6) cost 0.30000 w[2][5]=0.57000 c(1,4) cost 0.98000 30(20,40) w[3][3]=0.04000 c(1,4) cost 0.98000 30(20,40) w[3][4]=0.24000 c(2,5) cost 0.74000 40(30,50) w[3][6]=0.44000 c(1,5) cost 1.55000 30(10(,20),40) w[4][4]=0.00000 c(1,5) cost 1.28000 30(20,40(,50)) w[4][5]=0.10000 c(2,6) cost 1.630000 40(30(50,)) w[4][6]=0.20000 c(2,6) cost 1.630000 40(30(20,40(,50))) w[4][6]=0.10000 c(2,6) cost 1.630000 40(30(20,40(,50))) w[5][6]=0.14000 c(0,6) cost ???????????????????	w[0][1]=0.230000	c(2,2) cost 0.000000
w[0][3]=0.600000 c(4,4) cost 0.00000 w[0][4]=0.80000 c(5,5) cost 0.00000 w[0][5]=0.90000 c(6,6) cost 0.000000 w[0][6]=1.000000 c(0,1) cost 0.230000 10 w[1][1]=0.150000 c(1,2) cost 0.150000 20 w[1][3]=0.39000 c(3,4) cost 0.24000 40 w[1][4]=0.590000 c(4,5) cost 0.100000 50 w[1][5]=0.690000 c(5,6) cost 0.510000 10(,20) w[1][6]=0.79000 c(1,3) cost 0.540000 30(2,0) w[2][2]=0.030000 c(1,3) cost 0.710000 30(,40) w[2][4]=0.470000 c(3,5) cost 0.140000 40(,50) w[2][5]=0.570000 c(4,4) cost 0.980000 30(2,0) w[3][3]=0.040000 c(1,4) cost 0.980000 30(2,0,0) w[3][4]=0.240000 c(2,5) cost 0.140000 40(,50,0) w[3][6]=0.440000 c(2,5) cost 0.740000 40(30,50) w[3][6]=0.440000 c(2,5) cost 1.280000 30(2,0,40) w[4][6]=0.200000 c(2,5) cost 1.240000 40(30,60(50,1)) w[4][6]=0.200000 c(2,5) cost 1.240000 40(30,60(50,1)) w[4][6]=0.140000 c(0,6) cost ????????????????????????????????????	w[0][2]=0.360000	c(3,3) cost 0.000000
w[0][4]=0.800000 c(5,5) cost 0.00000 w[0][5]=0.90000 c(6,6) cost 0.00000 w[1][1]=0.020000 c(0,1) cost 0.230000 10 w[1][1]=0.020000 c(1,2) cost 0.150000 20 w[1][2]=0.150000 c(2,3) cost 0.270000 30 w[1][3]=0.390000 c(3,4) cost 0.240000 40 w[1][5]=0.690000 c(4,5) cost 0.140000 60 w[1][6]=0.790000 c(1,2) cost 0.510000 10(,20) w[1][6]=0.790000 c(1,3) cost 0.540000 30(20,) w[2][3]=0.270000 c(1,3) cost 0.710000 30(,40) w[2][4]=0.470000 c(3,5) cost 0.140000 40(,50) w[2][6]=0.670000 c(4,6) cost 0.30000 60(50,) w[2][6]=0.570000 c(1,4) cost 0.980000 30(20,40) w[3][3]=0.040000 c(2,5) cost 0.740000 40(,60(50,)) w[3][6]=0.440000 c(3,6) cost 1.550000 30(10(,20),40) w[4][6]=0.00000 c(1,5) cost 1.280000 30(20,40,50) w[4][6]=0.140000 c(0,5) cost 1.630000 40(30(50,)) w[4][6]=0.140000 c(0,6) cost 7?????? ?????????????????????????????	w[0][3]=0.600000	c(4,4) cost 0.000000
w[0][5]=0.900000 c(6,6) cost 0.00000 w[0][6]=1.000000 c(0,1) cost 0.230000 10 w[1][1]=0.20000 c(1,2) cost 0.150000 20 w[1][2]=0.150000 c(2,3) cost 0.270000 30 w[1][3]=0.390000 c(3,4) cost 0.240000 40 w[1][4]=0.590000 c(4,5) cost 0.100000 50 w[1][5]=0.690000 c(5,6) cost 0.140000 60 w[1][6]=0.790000 c(1,3) cost 0.540000 30(20,4) w[2][2]=0.030000 c(2,4) cost 0.710000 30(40) w[2][4]=0.470000 c(3,5) cost 0.440000 40(,50) w[2][5]=0.570000 c(4,6) cost 0.30000 60(50,1) w[2][6]=0.570000 c(1,4) cost 0.980000 30(20,40) w[3][4]=0.240000 c(2,5) cost 1.100000 20(10,30) w[3][5]=0.340000 c(2,6) cost 1.550000 30(10(,20),40) w[4][4]=0.000000 c(2,6) cost 1.280000 30(20,40,50)) w[4][4]=0.00000 c(2,6) cost 1.280000 30(20,40,50)) w[4][6]=0.200000 c(0,5) cost 1.850000 30(10(,20),40(,50)) w[4][6]=0.200000 c(0,6) cost 1.630000 40(30(20,1),60(50,1)) w[4][6]=0.140000 c(0,6) cost 1.630000 40(30(20,1),60(50,1))	w[0][4]=0.800000	c(5,5) cost 0.000000
	w[0][5]=0.900000	c(6,6) cost 0.000000
$ \begin{array}{llllllllllllllllllllllllllllllllllll$	w[0][6]=1.000000	c(0,1) cost 0.230000 10
$ \begin{array}{llllllllllllllllllllllllllllllllllll$	w[1][1]=0.020000	c(1,2) cost 0.150000 20
$ \begin{array}{llllllllllllllllllllllllllllllllllll$	w[1][2]=0.150000	c(2,3) cost 0.270000 30
$ \begin{array}{llllllllllllllllllllllllllllllllllll$	w[1][3]=0.390000	c(3,4) cost 0.240000 40
w[1][5]=0.690000c(5,6) cost 0.140000 60w[1][6]=0.790000c(0,2) cost 0.510000 10(,20)w[2][2]=0.030000c(1,3) cost 0.540000 30(20,)w[2][3]=0.270000c(2,4) cost 0.710000 30(,40)w[2][4]=0.470000c(3,5) cost 0.440000 40(,50)w[2][5]=0.570000c(4,6) cost 0.30000 60(50,)w[3][6]=0.670000c(1,4) cost 1.10000 20(10,30)w[3][3]=0.040000c(1,4) cost 0.980000 30(20,40)w[3][5]=0.340000c(2,5) cost 0.740000 40(,60(50,))w[3][6]=0.440000c(0,4) cost 1.550000 30(10(,20),40)w[4][4]=0.000000c(1,5) cost 1.280000 30(20,40(,50))w[4][5]=0.100000c(1,6) cost 1.280000 30(10(,20),40(,50))w[4][6]=0.20000c(0,5) cost 1.850000 30(10(,20),40(,50))w[6][6]=0.140000c(0,6) cost 2????????????????????????????????????	w[1][4]=0.590000	c(4,5) cost 0.100000 50
w[1][6]=0.790000c(0,2) cost 0.510000 10(,20)w[2][2]=0.030000c(1,3) cost 0.540000 30(20,)w[2][3]=0.270000c(2,4) cost 0.710000 30(,40)w[2][4]=0.470000c(3,5) cost 0.440000 40(,50)w[2][5]=0.570000c(4,6) cost 0.300000 60(50,)w[3][3]=0.040000c(1,4) cost 0.980000 30(20,40)w[3][4]=0.240000c(2,5) cost 0.940000 40(30,50)w[3][5]=0.340000c(3,6) cost 0.740000 40(,60(50,))w[3][6]=0.440000c(1,5) cost 1.550000 30(10(,20),40)w[4][4]=0.000000c(1,5) cost 1.280000 30(20,40(,50))w[4][5]=0.100000c(2,6) cost 1.240000 40(30,60(50,))w[4][6]=0.20000c(0,5) cost 1.630000 40(30(20,),60(50,))w[6][6]=0.140000c(0,6) cost ??????? ????????????????????????????	w[1][5]=0.690000	c(5,6) cost 0.140000 60
<pre>w[2][2]=0.030000 c(1,3) cost 0.540000 30(20,) w[2][3]=0.270000 c(2,4) cost 0.710000 30(,40) w[2][5]=0.570000 c(3,5) cost 0.440000 40(,50) w[2][6]=0.670000 c(4,6) cost 0.300000 60(50,) w[3][3]=0.040000 c(1,4) cost 0.980000 30(20,40) w[3][4]=0.240000 c(2,5) cost 0.940000 40(30,50) w[3][5]=0.340000 c(3,6) cost 0.740000 40(,60(50,)) w[3][6]=0.440000 c(3,6) cost 1.550000 30(10(,20),40) w[4][4]=0.000000 c(1,5) cost 1.280000 30(20,40(,50)) w[4][5]=0.100000 c(1,5) cost 1.240000 40(30,60(50,)) w[4][6]=0.200000 c(1,6) cost 1.850000 30(10(,20),40(,50)) w[5][5]=0.040000 c(1,6) cost 1.630000 40(30(20,),60(50,)) w[5][6]=0.140000 c(0,6) cost 2????????????????????????????????????</pre>	w[1][6]=0.790000	c(0,2) cost 0.510000 10(,20)
$ \begin{array}{llllllllllllllllllllllllllllllllllll$	w[2][2]=0.030000	c(1,3) cost 0.540000 30(20,)
$ \begin{array}{llllllllllllllllllllllllllllllllllll$	w[2][3]=0.270000	c(2,4) cost 0.710000 30(,40)
$ \begin{array}{llllllllllllllllllllllllllllllllllll$	w[2][4]=0.470000	c(3,5) cost 0.440000 40(,50)
$ \begin{array}{llllllllllllllllllllllllllllllllllll$	w[2][5]=0.570000	c(4,6) cost 0.300000 60(50,)
w[3][3]=0.040000c(1,4) cost 0.980000 30(20,40)w[3][4]=0.240000c(2,5) cost 0.940000 40(30,50)w[3][5]=0.340000c(3,6) cost 0.740000 40(,60(50,))w[3][6]=0.440000c(0,4) cost 1.550000 30(10(,20),40)w[4][4]=0.000000c(1,5) cost 1.280000 30(20,40(,50))w[4][5]=0.100000c(2,6) cost 1.240000 40(30,60(50,))w[4][6]=0.200000c(2,6) cost 1.850000 30(10(,20),40(,50))w[5][5]=0.040000c(1,6) cost 1.630000 40(30(20,),60(50,))w[5][6]=0.140000c(0,6) cost 2????????????????????????????????????	w[2][6]=0.670000	c(0,3) cost 1.100000 20(10,30)
$ \begin{array}{llllllllllllllllllllllllllllllllllll$	w[3][3]=0.040000	c(1,4) cost 0.980000 30(20,40)
$ \begin{array}{llllllllllllllllllllllllllllllllllll$	w[3][4]=0.240000	c(2,5) cost 0.940000 40(30,50)
<pre>w[3][6]=0.440000 w[4][4]=0.000000 w[4][5]=0.100000 w[4][6]=0.200000 w[5][5]=0.040000 w[5][6]=0.140000 w[5][6]=0.140000 w[6][6]=0.030000</pre>	w[3][5]=0.340000	c(3,6) cost 0.740000 40(,60(50,))
<pre>w[4][4]=0.000000 w[4][5]=0.100000 w[4][6]=0.200000 w[5][5]=0.040000 w[5][6]=0.140000 w[6][6]=0.030000</pre> c(1,5) cost 1.280000 30(20,40(,50)) c(2,6) cost 1.240000 40(30,60(50,)) c(0,5) cost 1.850000 30(10(,20),40(,50)) c(1,6) cost 1.630000 40(30(20,),60(50,)) c(0,6) cost ????????????????????????????????????	w[3][6]=0.440000	c(0,4) cost 1.550000 30(10(,20),40)
<pre>w[4][5]=0.100000 c(2,6) cost 1.240000 40(30,60(50,)) w[4][6]=0.200000 c(0,5) cost 1.850000 30(10(,20),40(,50)) w[5][5]=0.040000 c(1,6) cost 1.630000 40(30(20,),60(50,)) w[5][6]=0.140000 c(0,6) cost ????????????????????????????????????</pre>	w[4][4]=0.000000	c(1,5) cost 1.280000 30(20,40(,50))
<pre>w[4][6]=0.200000 c(0,5) cost 1.850000 30(10(,20),40(,50)) w[5][5]=0.040000 c(1,6) cost 1.630000 40(30(20,),60(50,)) w[5][6]=0.140000 c(0,6) cost ????????????????????????????????????</pre>	w[4][5]=0.100000	c(2,6) cost 1.240000 40(30,60(50,))
w[5][5]=0.040000c(1,6) cost 1.630000 40(30(20,),60(50,))w[5][6]=0.140000c(0,6) cost ????????????????????????????????????	w[4][6]=0.200000	c(0,5) cost 1.850000 30(10(,20),40(,50))
w[5][6]=0.140000 c(0,6) cost ????????????????????????????????????	w[5][5]=0.040000	c(1,6) cost 1.630000 40(30(20,),60(50,))
w[6][6]=0.030000	w[5][6]=0.140000	c(0,6) cost ????????????????????????????????????
	w[6][6]=0.030000	

Construct the final optimal binary search tree and give its cost. SHOW YOUR WORK. (10 points)

CSE 5311

Spring 2003

Test 2

Closed Book Questions

Multiple Choice - WRITE your answer to the LEFT of each problem. 4 points each.

- 1. The length of the TSP tour found by the triangle inequality technique achieves what minimization ratio?
 - A. 0.5
 - B. 1 + ∈
 - C. 1.5
 - D. 2
- 2. When performing bin packing using the First-Fit Decreasing technique, the total number of items placed in the bins past the optimal bins (1 .. OPT) is bounded by:
 - A. 1 + ∈
 - B. 2
 - C. OPT 1
 - D. OPT

3.

- Which of the following is NOT required when showing that problem B is NP-complete by a reduction from problem A. A. The reduction takes polynomial time.
 - B. The reduction has an inverse that takes each instance of problem B to an instance of problem A.
 - C. Problem A is NP-complete

D. The reduction must be consistent for the decision results for each instance of problem A and the corresponding instance of problem B.

- 4. When <u>combining</u> the left and right parts for the 2-d closest pair algorithm along the vertical dividing line, each left-side point "near" the line has its distance computed to no more than this number of right-side points.
 - A. 6
 - B. 8
 - C. lg n
- D. n/2 5. Radix s
 - Radix sort is useful for which of the following:
 - A. Constructing a longest common subsequence using O(m+n) space.
 - B. Constructing a suffix array.
 - C. Constructing a suffix tree.
 - D. Constructing the polynomial for Karp-Rabin string search
- 6. Which of the following is not required before relabeling ("lifting") a vertex to a new height?
 - A. Any eligible edges for the present height have been saturated.
 - B. Both breadth-first searches have been done.
 - C. The vertex is not the source or sink.
 - D. The vertex is overflowing.
- 7. Which of the following is not true for the stable marriages technique that yields the man-optimal matching?
 - A. Men propose starting with the beginning of their preference lists.
 - B. The order of proposals is irrelevant and the same matching will always be obtained.
 - C. When a man proposes to a woman and the woman accepts, they must be paired in the final matching.
 - D. Women improve their situation by accepting proposals toward the beginning of their preference lists.
 - Among the problems listed, this problem achieves the best approximation ratio:
 - A. Edge coloring
 - B. Set cover

8.

- C. TSP with the triangle inequality
- D. Vertex cover
- 9. Articulation points are found by:
 - A. 2-d closest pairs
 - B. Convex hull
 - C. Depth-first search
 - D. Strongly connected components
- 10. On an augmenting path, a critical edge is:
 - A. an edge from the source
 - B. an edge into the sink
 - C. an edge that used to be saturated
 - D. an edge with the minimum residual capacity
- 11. Fill in the Z table. 10 points

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
a	С	a	b	a	С	a	С	a	b	a	С	a	b	a	С

Name

CSE	531	1								Name										
Sprin	ng 20	003																		
Test	2																			
Oper	n Bo	ok Que	estion	S																
1.	Give the suffix array for this string. 15 points																			
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15				
	а	С	а	b	а	С	а	С	а	b	а	С	а	b	а	С				
2.	Sup	pose a	in app	licatio	on req	uires f	inding	small	vertex	x cover	s for	many	undire	ected g	graphs	. Even	though you	are initial	ly	
	con	cerned	l abou	it this	issue,	you p	roceed	with	great c	confide	nce a	fter le	arning	that e	every g	graph v	vill be bipar	tite. Why	? 10 po	ints.
3.	Tra	ce the	execu	ition o	f the l	oiconn	ected of	compo	onents	algorit	hm o	n the f	follow	ing gr	aph by	giving	g the discove	ery time ar	nd "bacl	<"
	v.01	no for	anah t	ortow	alon	- with	listing	thory	ortions	in anal	h hia		ad an		nt 15	noint	9	•		

value for each vertex, along with listing the vertices in each biconnected component. 15 points Vertex: A B C D E F G H I Discovery: Back:



4. List the remaining operations to complete this instance of network flows by push-relabel. 10 points

