Name: ________________________________

Email address: ________________________________

Quiz Section: __________

SECTION B

CSE 326 Winter 2007: Final Exam

(closed book, closed notes, calculators o.k.)

Instructions Read the directions for each question carefully before answering. We will give partial credit based on the work you **write down**, so show your work! Use only the data structures and algorithms we have discussed in class or which were mentioned in the book.

Note: For questions where you are drawing pictures, please circle your final answer for any credit. There is one extra page at the end of the exam that you may use for extra space on any problem. If you detach this page it must still be turned in with your exam when you leave.

Advice You have 1hour and 50 minutes, **do the easy questions first**, and work quickly!

Total: Time: 1hr and 50 minutes.

Question	Max Points	Score
1	24	
2	20	
3	10	
4	4	
5	13	
6	11	
7	8	
8	10	
9	8	
Total	108	

1) **[24 points total, 2 points each] True/False**. Circle True or False below. You do *not* need to justify your answers.

a.	The amortized running time of M insertions into an initially empty AVL tree is O(log N) per operation.	**True**	**False**
b.	If unions are performed using union-by-size on N elements, then the depth of any node is never more than log N .	**True**	**False**
c.	In a d-Heap, if the heap stores N items, then an insert is more costly than a deleteMin.	**True**	**False**
d.	Dijkstra's algorithm for shortest path and Prim's minimum spanning tree algorithm have the same big-Oh worst case running time.	**True**	**False**
e.	Minimum spanning trees are unique: that is for a given graph with given weights, there is only one subgraph which is the minimum spanning tree.	**True**	**False**
f.	The worst case runtime of heapsort becomes O(N) if Floyd's buildheap is used.	**True**	**False**
g.	The worst case running time of Quicksort is O(N log N).	**True**	**False**
h.	In an AVL tree, after inserting a key, we have to do at most one (single or double) rotation to maintain the balance property.	**True**	**False**
i.	An N-element array of Boolean values (true or false) can be sorted in O(N) time.	**True**	**False**
j.	The best case runtime for a merge operation on two binomial *queues* is O(log N).	**True**	**False**
k.	Traversing the contents of a sorted array in order has good temporal locality (on the array).	**True**	**False**
l.	Traversing the contents of a sorted array in *reverse* order has good spatial locality (on the array).	**True**	**False**

2) **[20 points total] Short Answer Problems:** *Be sure to answer all parts of the question!!*

   a) **[2 points]** When merging two heaps, do leftist heaps have better, worse, or the same worst case running time (asymptotically) as compared to binary heaps? Justify this answer.

   b) **[2 points]** What is the maximum number of trees in a binomial queue whose tallest tree is of height h?

c) **[4 points]** Give the big-O best and worst case running time for find in a hash table with N elements where separate chaining is used and each bucket is an AVL tree. For full credit, describe *what* the best and worst cases are.

Best case:

Worst case:

d) **[4 points]** Suppose that in performing quicksort, the pivot that you pick is always the smallest element (that is in each recursive call to quicksort the pivot picked is always the smallest element in the supplied list.) Give the best and worst case running times (up to constant factors) of this quicksort algorithm when sorting N distinct values. Justify your answer.

e) **[4 points]** If a graph has |V| vertices and |E| edges, what is the space requirement (big-O notation is fine) for representing this graph as an adjacency matrix? As an adjacency list? If we perform a topological sort on a sparse graph, which representation should we use and why?

f) **[4 points]** Describe what a greedy algorithm is and give an example of one greedy algorithm we have studied in class.

3) [10 points] Splay Trees

Imagine that the following operations are performed on an initially empty splay tree:

Insert(10), Insert(1), Insert (3), Insert (5), Insert (7), Insert (13), Find (3).

Show the state of the splay tree after performing each of the above operations. Be sure to label each of your trees with what operations you have just completed.

4) [4 points] B-trees

Given the following parameters:

Disk access time = 2milli-sec per byte

1 Page on disk = 2048 bytes

Key = 16 bytes

Pointer = 8 bytes

Data = 128 bytes per record (includes key)

a) **[2 points]** What are the best values in a B-tree for:

M =

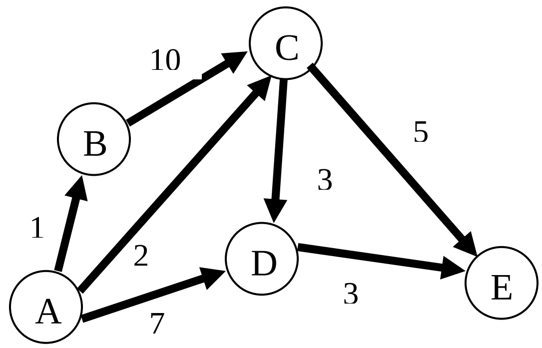
and

L =

b) **[2 points]** Briefly explain why these values are the best values for the B-tree.

5) [13 points total] Graphs and Dijkstra's Algorithm

Use the following graph for this problem:



a) **[2 points]** Draw both the adjacency matrix and adjacency list representations of this graph. Be sure to specify which is which.

b) **[2 points]** List all valid topological orderings of the nodes in the graph above.

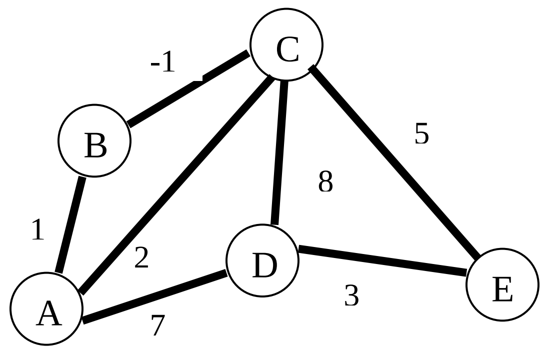
c) **[5 points]** Step through Dijkstra's Algorithm to calculate the single source shortest path from A to every other vertex. You only need to show your final table, but you should show your steps in the table below for partial credit. Show your steps by crossing through values that are replaced by a new value. *Note that the next question asks you to recall what order vertices were declared known*.

Vertex	Known	Distance	Path
A			
B			
C			
D			
E			

d) **[2 points]** In what order would Dijkstra's algorithm mark each node as *known***?**

e) **[2 points]** Describe an edge which could be added to the above graph (i.e. give its start and end points and its weight) which would make Dijkstra's algorithm fail.

6) [11 points] Prim's and Kruskal's Algorithms

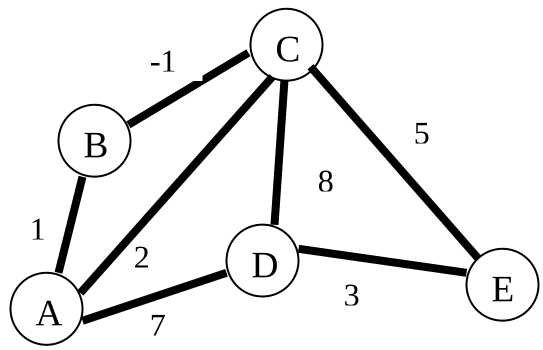


a) **[5 points]** Step through Prim's Algorithm to calculate a minimum spanning tree for the above graph. Start with selecting node A. You only need to show your final table, but you should show your steps in the table below for partial credit. Show your steps by crossing through values that are replaced by a new value. *Note that the next question asks you to recall what order vertices were declared known*.

Vertex	Known	Cost	Path
A			
B			
C			
D			
E			

b) **[2 points]** In what order are the vertices declared known in your running of Prim's algorithm?

Reproduction of the graph on the previous page:



c) **[2 points]** Draw the minimum spanning tree produced by running Kruskal's algorithm on the graph above.

d) **[2 points]** What is the total cost of the minimum spanning tree shown above in part c)?

7) [8 points] Hashing

a) **[5 points]** Draw the contents of the hash table in the boxes below given the following conditions:

The size of the hash table is 11.
Open addressing and quadratic probing is used to resolve collisions.
The hash function used is H(k) = k mod 11

What values will be in the hash table after the following sequence of insertions? Draw the values in the boxes below, and show your work for partial credit.

28, 1, 13, 10, 12, 45, 20

0	
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	

b) **[3 points]** The above hash table you have created has the problem that there is no guarantee that the next insertion into the table will result in an available empty location. If we rehash the table we created in part (a) (still using open addressing and quadratic probing), what is the smallest table size for the new rehashed table such that we are guaranteed that the next insertion will be successful? (give an exact number)

8) [10 points] Disjoint Sets

Give the state of the disjoint sets data structure after the following sequence of operations, starting from singleton sets {1}, . . . , {8}. Use union-by-size and path compression. In case of ties, always make the lower numbered root point to the higher numbered one.

(a) **[4 points]** Draw the state of the disjoint sets data structure after performing UNION(1,2), UNION(3,4), UNION(5,6), UNION(7,8), UNION(1,4), UNION(6,7), UNION(4,5)

(b) **[2 points]** Draw the state of the disjoint sets data structure after performing FIND(1) on the disjoints set data structure produced at the end of part (a)

(c) **[4 points]** Assuming the up-tree representation as described in class, in an implementation where UNION must always be passed a root, what is the worst case runtime of a *single* UNION operation when:

Union-by-size is used (without path compression):

Union-by-size is used (with path compression):

9) [8 points] Graph Problems

a) **[4 points]** Give a connected directed graph with 5 nodes that has the largest possible number of topological sorts. How many different topological sorts does it have?

b) **[4 points]** Does adding a negative edge weight *always* cause Dijsktra's algorithm to fail? Justify your answer.

Extra page for scratch work. Please turn in with exam.