Data Structures
Fall 2000
Final Exam.

1. (15 pts) **(a)** (5 pts) Define leftist (min) heaps. **(b)** (10 pts) Draw the leftist heap resulting from merging the two leftist heaps in Figure 1. Show your derivation in detail.

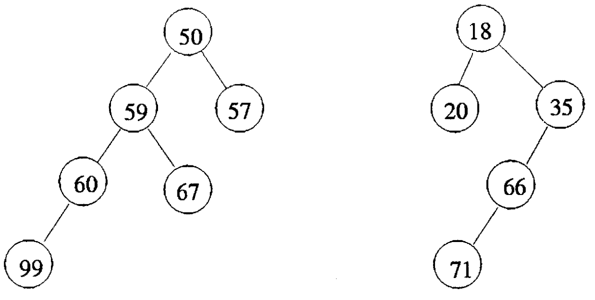
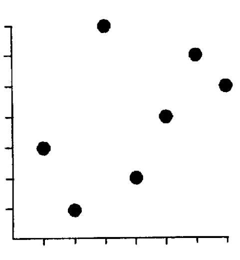


Figure 1: A leftist heap.

2. (12 pts) The behavior of a sorting algorithm can be illustrated pictorially by displaying the contents of the array to be sorted using the so-called characteristics diagrams. In displaying the contents of an array, say A[1..n], a '•' is placed at position (i,j) for A[i]=j. See Figure 2 for a simple example. Clearly in a sorted array each '•' appears above the one to its left. In what follows you will see four characteristic diagrams (a) – (d), representing some intermediate configurations of some sorting algorithms. Indicate their corresponding sorting algorithms from the list of *insertion sort, selection sort, quicksort, merge sort, bubble sort, shell sort, heap sort, radix sort, bucket sort.*



A[1..7]= 3 1 7 2 4 6 5

Figure 2: An example.

3. (20 pts)

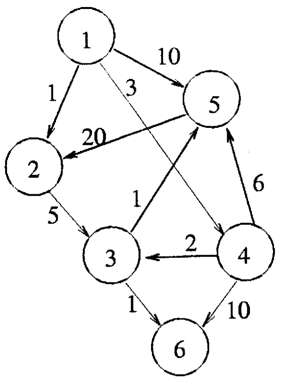
   (a) (10 pts) Insert 20, 5, 16, 77, 3, 1, 23, 56, 99, 12 into an initially empty (min) binomial heap. Show your derivation in sufficient detail.

   (b) (10 pts) Suppose we want to use the potential method to calculate the time needed to perform n insertions.
   (1) Can you suggest a potential function for binomial heaps that enables us to carry out the above analysis?
   (2) Suppose we define the actuial cost of an insertion to be one time unit plus an extra unit for each linking step. (E.g., inserting a node into $B_0 - -B_1 - -B_2$ results in B_3, and the cost of such an insertion is 1+ (1+1+1)=4.) For each of the insertions in (a), specify the actual cost as well as the potential difference before and after the insertion. To be more precise, complete the following table for <u>all</u> the insertions.

insertion / actual cost / potential difference	20	5	. . .	12
actual cost			• •	
potential difference			•	

Figure 3: binomial heap.

4. (18 pts) Recall that at each step of Dijkstra's algorithm, a set S is maintained to record the set of vertices whose shortest distance from the source is already known. Furthermore, an array d is used to record the length of the shortest path to each vertex through vertices in S.
Given a graph G (shown below), complete the following table. (Note that vertex 1 is the source.)



iteration	S	d[2]	d[3]	d[4]	d[5]	d[6]
initial	{1}	1		3	10	∞
1		1		3		
2		1		3		
3		1	5	3		
4		1	5	3		
5		1	5	3		

Figure 4: Dijkstra's algorithm

5. (11 pts) Draw the decision tree for applying *shell sort* to an array of 3 keys x, y, z using 1, 2 as the increment sequence.

6. (12 pts) What is the worst-case running time of performing the following operations on a heap (of n nodes) given below. (That is, complete the following table.)

worst case running time of a single operation

operation / types of heaps	find min	union	delete min
binary heap			
binomial heap			
leftist heap			
skew heap			

Figure 5: Heaps

7. (12 pts) For each of the following algorithms discussed in class, identify a key abstract data type (ADT) that is crucial in making the algorithm efficient. Explain (in no more than three lines) the role played by each of such ADTs in the respective algorithm.
(1) Kruskal's algorithm (2) Prim's algorithm (3) Dijkstra's algorithm