1. There are 2 identical computers and $N$ jobs to be run. Each job runs on a single computer and each computer can only run one job at a time. The run times of the jobs are known a priori to be $T_1, T_2, \cdots, T_N$ and all the jobs are available and ready to run at the beginning of the processing period. The problem is to find a schedule for the jobs that will complete all of them in the shortest total time?

(a) Give a greedy algorithm for this problem and state its time complexity.

(b) Either explain why your algorithm IS optimal, OR give a small example of a set of jobs that demonstrates that your greedy algorithm isn’t optimal.
2. A certain insurance company has 1,000,000 clients and wishes to construct a Data Dictionary to support *insert*, *delete*, and *find* functions. The records are identified by the social security numbers of the clients. Each record is of size 300 bytes. Assume that a pointer is 4 bytes long.

(a) Calculate the memory space taken up by a Separate Chaining hash table of length 500,001.

(b) Calculate the memory space taken up by an Open Addressing hash table of length 2,000,001.

(c) Calculate the mean number of links traversed during a search in case (a) above.
Question 2 continued

(d) Calculate the mean number of probes during a successful search for scheme (b) above with linear probing. State any assumptions you make.

(e) Calculate the mean number of probes during a successful search for scheme (b) using quadratic probing.

(f) What simple change to scheme (a) would improve its performance? Explain.
3. A data dictionary is required that supports insert, deleteMin and deleteMax operations.

(a) If a minHeap is used, what will be the upper bound on the time for an insert operation?

(b) If a minHeap is used, what will be the upper bound on the time for a deleteMin operation?

(c) If a minHeap is used, what will be the upper bound on the time for a deleteMax operation?
Question 3 continued

(d) A scheme is suggested in which pointers to the data records are stored in both a minHeap and a maxHeap. Each data record contains the array indices of the positions of that record in the two heaps. Give the pseudocode for a deleteMin operation for this scheme and give its time complexity.
Question 3 continued

(e) What other data structure could efficiently support these three operations? Give the time complexities of the three required operations using the data structure that you suggest.
4. Two arrays, $X$ and $Y$, each contain $N$ numbers. The following scheme is suggested to form a sorted list of the entire $2N$ values:

1. Sort the array $X$ using algorithm $A$
2. Sort the array $Y$ using algorithm $B$
3. Merge the two sorted arrays using algorithm $C$

(a) Show that this scheme is suboptimal, regardless of the choices for algorithms $A$, $B$, and $C$.

(b) If arrays $X$ and $Y$ are already sorted, but not combined into a single array, give an $O(\log N)$ algorithm for finding the median of the $2N$ values.
5. In the ad-hoc radios network used in the project, a graph, \( G = (V, E) \) is used to represent the communications network. Each radio is represented by a vertex in \( V \) and each edge in \( E \) represents a radio link. A minimal spanning tree, \( T = (V, E_T) \) is used for broadcast communications. Assume that a data structure containing a description of the entire graph \( G \) is available at a node in which the following computations are to take place.

(a) If a radio link \( e \in E_T \) fails for some reason, the spanning tree no longer connects all the nodes. Describe an efficient scheme for producing a new minimal spanning tree \( T' \) that connects all the nodes, if possible. Your scheme should return a Boolean to indicate whether the scheme succeeded.

(b) Prove that, if \( T \) was minimal, your scheme does produce a minimal spanning tree.
Question 5 continued

(c) Suppose that a radio fails. How would you efficiently compute a new minimal spanning tree if one existed?

(d) Suppose that the articulation points of $G$ have been computed by the algorithm given in the text.

i. How might the articulation points be used by the managers of the network?

ii. If a link fails, how could the articulation points be efficiently recalculated, assuming the data structures remained from the previous computation of the articulation points.