1. (10%) If you are going to write a filter program that reads in a sequence of integers and prints out the integers when they appear for the first time (for example, if the input is 1 2 2 1 5 1 1 7 7 7 7 1 1 1 1 1 1 1 1, your program should print out 1 2 5 7), what data structure will you use in storing the information required? Please analyze the space complexity of your program.
2. (15%) *Adjacency matrix* and *adjacency list* are two of the widely adopted choices for representing a graph. Please analyze the time and space tradeoffs between these two implementations. Which one will you consider when the graphs to be stored are *sparse*? Justify your answer.
3. (10%) If you are going to write a program **Parentheses** that reads in a text stream from standard input and determines whether its parentheses are properly balanced, what data structure will you use in order to simplify the implementation? For example, your program should print true for [0]{}{[00]0} and false for [(]). Justify your answer.
4. (15%) Write a function that takes the first Node in a linked list as argument and reverses the list, returning the first Node in the result.
5. (10%) Please provide the definition of a binary heap. Is an array that is sorted in decreasing order a max-oriented heap?
6. (20%) Please provide a hash function for a hash table that has M buckets and uses strings as the keys. In what situations your hash function might have a bias? What technologies are usually employed to revolve the situation of collisions in hash implementation?
7. (20%) Please provide the definition of a *priority queue*. By using what data structure (ordered array, unordered array, or heap) in the implementation of priority queues, the running time of an *insert* operation can be optimized? By using what data structure, the running time of the operation *remove the maximum* can be optimized? By using what data structure, both *remove the maximum* and *insert* can be achieved in logarithmic time?

試題隨卷繳回