1. (10%) Write a function (pseudo codes) that returns the smallest value in a given **linked list**.

Note: A **node** in a **linked list** has a pointer, named 'next', pointing to the next node.

2. (20%) Write a function (pseudo codes) that frees the nodes in odd positions (the first, third, fifth, and so forth) in a given **double linked list**.

Note: A node in a **double linked list** has two pointers, named 'next' and 'prev', respectively, where the 'next' pointer is used to find the next node and the 'prev' pointer is used to find the previous node.

3. (15%) (a) Provide the definition of an **adjacency list** for representing a graph. (b) Provide the definition of an **adjacency matrix** for representing a graph. (c) List the advantages and disadvantages for both representations.

4. (20%) (a) Please design a procedure (pseudo codes) to perform breadth-first search using a **queue** (**FIFO**). This function takes one node of an **un-directed graph** as the input, where the graph is represented using an adjacency list. Assume that your queue has the two basic operations: **enqueue** (insert a new item onto queue) and **dequeue** (remove and return an item from the queue). (b) Analyze the time complexity of your codes, assuming that your queue adopts the linked list implementation.

5. (20%) (a) What is a **complete binary tree**? (b) What is a **max heap?** (c) Design a procedure (pseudo codes) to construct a max heap in linear time.

6. (15%) (a) Please explain what a **priority queue** is. (b) How can a heap improve the performance (time complexity) of a priority queue when compared with using an **array** or **list** implementation? You can answer this question by listing the complexity of finding the largest element in a **max priority queue** in different implementations.

試題隨卷繳回