Programming Component

1. (10 marks) An online retail store has a leger that stores all transactions happened in this month. The leger is implemented using a singly linked list in reverse order as shown in the following figure.

Head is the reference pointing to the most recent transaction record. \( T_x \) is the record of the \( i \)-th transaction happened in this month, i.e. \( T_x_1 \) is the very first transaction happened in this month.

The class of a transaction record is defined as:

```python
class Record:
    def __init__(self, merchandise, time, prev):
        self.merchandise = merchandise
        self.time = time
        self.prev_trans = prev
```

The attribute `prev_trans` is a reference to the previous transaction. The attribute `merchandise` indicates the merchandise of this transaction. Time field is represented using the Unix timestamp which is the number of seconds that have elapsed since January 1, 1970 (midnight UTC/GMT).

The owner of this online retail website suddenly found the \( m \)-th transaction \( T_x_m \) is a duplicated transaction, which should be deleted. Please write a Python function
delete_transaction that given the head reference of the leger head_trans and a valid index \( m (1 \leq m < n) \), deletes the m-th transaction happened in this month. The function returns the merchandise field value of the deleted transaction record. The owner requires you to do this operation in one pass of the leger.

**Hint:**
1) You do not have the information of the length of the list. And you can not traverse the linked list to get the length of the list before deleting operation because of the “one pass” requirement.
2) “One pass” means after you scan the list once, you can not come back to the head and start another scan.
3) There is no information in the transaction record indicating the position of this record in the list.
4) An example of building a leger and delete the 2\(^{nd}\) record as follows:
   ```python
   tx1 = Record("apple", 1580, None)
   tx2 = Record("banana", 2390, tx1)
   tx3 = Record("carrot", 3452, tx2)
   tx4 = Record("doll", 3789, tx3)
   
   delete_transaction(tx4, 2) -> "banana"
   ```

2. (10 marks) For some reason, the leger was disordered by mistake. Please implement the quick sort algorithm on the leger in Question 1 based on the values of time field. Note: you can not convert the linked list to a Python list or use Python built-in sort()/sorted() methods. The function name is leger_quickSort. It takes a head reference of the leger named as leger_head and returns the new head of the leger.

**Hint:**
Please note that after sorting, the leger_head should point to the most recent transaction. As a result, the timestamp of the leger_head is the largest one.

3. (10 marks) You are the admin of one website. This website has a log system that stores all events in an array and orders events by time. Because of some failure of the memory chip, the array was split into two arrays and concatenated in the wrong order. The whole process is shown as follows:
\( E_i \) is the i-th event. Each event record is defined as:

```python
class Event:
    def __init__(self, time, event):
        self.time = time
        self.event = event
```

Note that \( E_k \).time > \( E_j \).time if and only if \( k > j \). time field is represented the same as Question 1.

Your manager asks you to find the very first event in this disordered array and the time cost of your program must be \( O(\log n) \).

Please write a Python function `find_first`. It takes an array named `disordered` and returns the time of the first event. You will be using the array implementation in the provided `myArray.py`.

**Written Component**

4. (5 marks) Consider the same online retail store problem of Question 1. The owner of this online retail store found another duplicated transaction by scanning the list by himself. This time, he gave you the reference to the duplicated transaction and let you delete it. But unfortunately, he forgot to give you the head reference. Please describe how to delete the transaction from the leger. You don’t have to write a program. You can use natural language to describe your solution in a short paragraph, or you can provide a segment of pseudo-code, or Python style code.

**Hint:**
Without a head reference, you can not traverse the linked list.

5. (10 marks) Describe a worst-case of quick sort algorithm. What is the runtime in the worst-case (using the Big-O notation)? Justify your answer.

**Submission:** Please use the provided files for the programming component: `a2q1.py`, `a2q2.py`, `a2q3.py`. Please submit pdf files for Question 4 and Question 5.