CSC 212 Tutorial #5 Solution
Queue & PQueue

Problem 1

```
public static <T> void split(Queue<T> q,Queue<T> oq,Queue<T> eq) {
    recSplit(q, oq, eq, q.length(), 1);
}
private static <T> void recSplit(Queue<T> q, Queue<T> oq, Queue<T>
    eq, int count, int pos) {
    if (pos > count)
        return;
    T element = q.serve();
    q.enqueue(element);
    if (pos % 2 == 1) {
        if (!oq.full())
            oq.enqueue(element);
    }
    else {
        if (!eq.full())
            eq.enqueue(element);
    }
    recSplit(q, oq, eq, count, pos + 1);
}
```

Problem 2

```
public static <T> void merge(PriorityQueue<T> mq, PriorityQueue<T>
    pq1, PriorityQueue<T> pq2) {
    PQElement<T> element = null;
    int count = pq1.length();
    for (int i = 0; i < count; i++) {
        element = pq1.serve();
        if (!mq.full())
            mq.enqueue(element.data, element.priority);
    }
    count = pq2.length();
    for (int i = 0; i < count; i++) {
        element = pq2.serve();
        if (!mq.full())
            mq.enqueue(element.data, element.priority);
    }
}
```

Problem 3

```
public static <T> void remove(PriorityQueue<T> pq, int p) {
    PQElement<T> element = null;
    PriorityQueue<T> tempPQ = new LinkedPQ<T>();
    int count = pq.length();
    for (int i = 0; i < count; i++) {
        element = pq.serve();
        if (element.priority >= p)
            tempPQ.enqueue(element.data, element.priority);
    }
    count = tempPQ.length();
    for (int i = 0; i < count; i++) {
        element = tempPQ.serve();
        pq.enqueue(element.data, element.priority);
    }
}
```