CSC 212 Tutorial 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 remove(PriorityQueue<T> pq, int p) {
    PriorityQueue<T> tempPQ = new LinkedPQ<T>();
    int count = pq.length();
    for (int i = 0; i < count; i++) {
        PQElement<T> element = pq.serve();
        if (element.priority >= p)
            tempPQ.enqueue(element.data, element.priority);
    }
    count = tempPQ.length();
    for (int i = 0; i < count; i++) {
        PQElement<T> element = tempPQ.serve();
        pq.enqueue(element.data, element.priority);
    }
}
```

Problem 3

```
public static <T> boolean search(Stack<T> st, T e) {
    if (st.empty())
        return false;
    T top = st.pop();
    boolean found;
    if (top.equals(e))
        found = true;
    else
        found = search(st, e);
    st.push(top);
    return found;
}
```