Christian Rodriguez

**Algorithms HW 6**

2. Prove that your algorithm takes O (kn ) or O (n lg n ) time. If your algorithm

takes longer than either of these, \_nd a faster algorithm. Hint: if you are

using an appropriate data structure, this question should be very simple.

2

Solutions:

1. Design an algorithm that solves the Josephus problem, using an appropriate data structure. I will be using nodes to create a circular linked list for the josephus algorithm. Assuming a circular linked list. Following pseudocode is referencing the following site: http://codereview.stackexchange.com/questions/4628/circular-linked-list

Input:

n: integer number of soldiers

k: integer ordinal number of the soldiers being killed

Output:

A subset of a single surviving soldier

1 Algorithm: JosephusAlgorithm

2 n = 41

3 k = 3

4 while n <= 1

5 for i = 1 to k with i++ increments

6 prev = tmp

7 tmp = tmp->next

8 end

9 prev->next = tmp->next

10 delete tmp

11 n--

12 end

1. Prove that your algorithm takes O(kn) or O(nlgn) time.

Using linked lists I can create a nested for loop in which the outer loop in line 4 iterates n times until there is only one rebel soldier left, and the inner loop in line 5 iterates every k times and once it reaches k it skips what would normally be the next linked list and goes to the one after it as shown in line 9, and proceeds to decrement the n number of rebel soldiers in line 11.