Homework 6

Due 10/01/15

September 24, 2015

Flavius Josephus was a Jewish historian living in the first century. According to Josephus' account of the siege of Yodfat, he and his 40 soldiers were trapped in a cave by Roman soldiers. They chose suicide over capture, deciding to form a circle and start killing every third person. Josephus states that by luck or possibly by the hand of God, he and another man remained the last and surrendered to the Romans (*from Wikipedia*).

The Josephus Problem, named for this apocryphal story, asks the question of where to stand in a circle of n people in order to be the last person alive when we kill every k^{th} person.

When $n = 41$ and $k = 3$, the solution to the Josephus Problem is 31. At the beginning, all 41 rebels are alive:

$r_1, r_2, r_3, r_4, r_5, r_6, r_7, r_8, r_9, r_{10}, r_{11}, r_{12}, r_{13}, r_{14}, r_{15}, r_{16}, r_{17}, r_{18}, r_{19}, r_{20}, r_{21},$
$r_{22}, r_{23}, r_{24}, r_{25}, r_{26}, r_{27}, r_{28}, r_{29}, r_{30}, r_{31}, r_{32}, r_{33}, r_{34}, r_{35}, r_{36}, r_{37}, r_{38}, r_{39}, r_{40}, r_{41}$

After 13 killings, one "round" of the circle is completed, and 28 rebels remain (the most recently dead rebel is marked by $\star$):

$r_1, r_2, r_4, r_5, r_7, r_8, r_{10}, r_{11}, r_{13}, r_{14}, r_{16}, r_{17}, r_{19}, r_{20}, r_{22}, r_{23}, r_{25}, r_{26}, r_{28}, r_{29},$
$r_{31}, r_{32}, r_{34}, r_{35}, r_{37}, r_{38}, \star, r_{40}, r_{41}$

In the second round, 10 die and 18 remain:

$r_2, r_4, r_7, r_8, r_{11}, r_{13}, r_{16}, r_{17}, r_{20}, r_{22}, r_{25}, r_{26}, r_{29}, r_{31}, r_{34}, r_{35}, r_{38}, r_{40}, \star$

The remaining rounds appear in the following table (no one dies in round 8):

Round	Rebels
3	$r_2, r_4, r_8, r_{11}, r_{16}, r_{17}, r_{22}, r_{25}, r_{29}, r_{31}, r_{35}, r_{38}, \star$
4	$r_2, r_4, r_{11}, r_{16}, r_{22}, r_{25}, r_{31}, r_{35}, \star$
5	$r_2, r_4, r_{16}, r_{22}, \star, r_{31}, r_{35}$
6	$r_4, r_{16}, \star, r_{31}, r_{35}$
7	$r_{16}, r_{31}, \star$
9	$\star, r_{31}$

1. Design an algorithm that solves the Josephus Problem, using an appropriate data structure.

2. Prove that your algorithm takes $O(kn)$ or $O(n \lg n)$ time. If your algorithm takes longer than either of these, find a faster algorithm. *Hint:* if you are using an appropriate data structure, this question should be very simple.