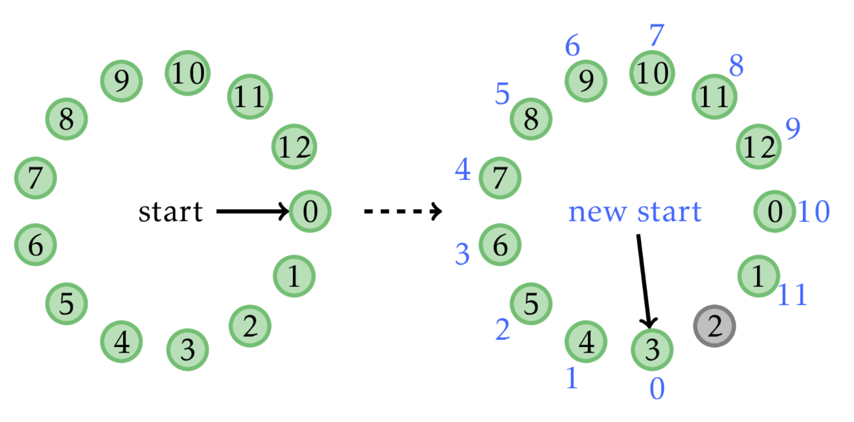
Ace Your Next Coding Interview by Learning Algorithms (/course/102772)
Course progress: 106/353

1 Welcome
2 Algorithms and Complexity
3 Algorithm Design Techniq...

As the Figure above illustrates, the number of alive rebels is reduced by a factor of approximately $k/(k-1)$ after each traversal of the circle. Indeed, since each k-th rebels is killed during each traversal of a circle with m rebels, m/k rebels are killed during this traversal, reducing the number of rebels by a factor $m/(m-m/k)$. Therefore, it will take approximately $\log_{k/(k-1)} n$ traversals of the entire circle to find $\mathrm{Josephus}(n, k)$, implying that the running time of $\texttt{Josephus}(n, k)$ is $O(n \log n)$.

Fast algorithm for solving the Josephus Problem. After the first out of n rebels is killed, we face a new Josephus Problem with $n-1$ rebels, but with a new starting position and new numbering of rebels: rebel k is now rebel 0, rebel $k+1$ is now rebel $1, \ldots$, rebel 0 is now rebel $n-k$, and the (last) rebel $k-2$ is now rebel $n-2$.



Here is the formula for transforming old numbers into new numbers for $n = 13$ and $k = 3$:

$$(\text{old number} - 3) \bmod 13 = \text{new number}\ .$$

In general,

$$(\text{old number} - k) \bmod n = \text{new number}\ .$$

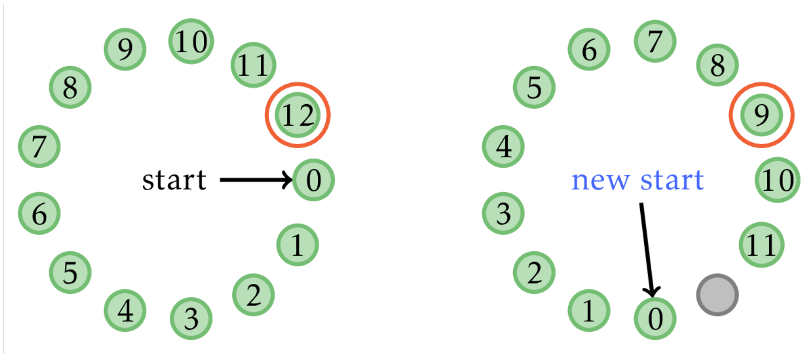
For example, $\mathrm{Josephus}(13, 3) - 3 \bmod 13 = 12 - 3$ and $\mathrm{Josephus}(12, 3) = 9$ as illustrated in Figure below.

Ace Your Next Coding Interview by Learning Algorithms (/course/102772)
Course progress: 106/353

1 Welcome

2 Algorithms and Complexity

3 Algorithm Design Techniq...



Therefore,

$$(\text{Josephus}(n,k) - k) \bmod n = \text{Josephus}(n-1,k)\ .$$

Exercise Break: Using this recurrence, implement a faster algorithm for computing $\text{Josephus}(n,k)$ in $O(n)$ time.

STOP and Think: Can you design an even faster algorithm for the Josephus Problem in the case $k=2$ referred to as the *binary* Josephus problem?