Josephus Problem Recursion

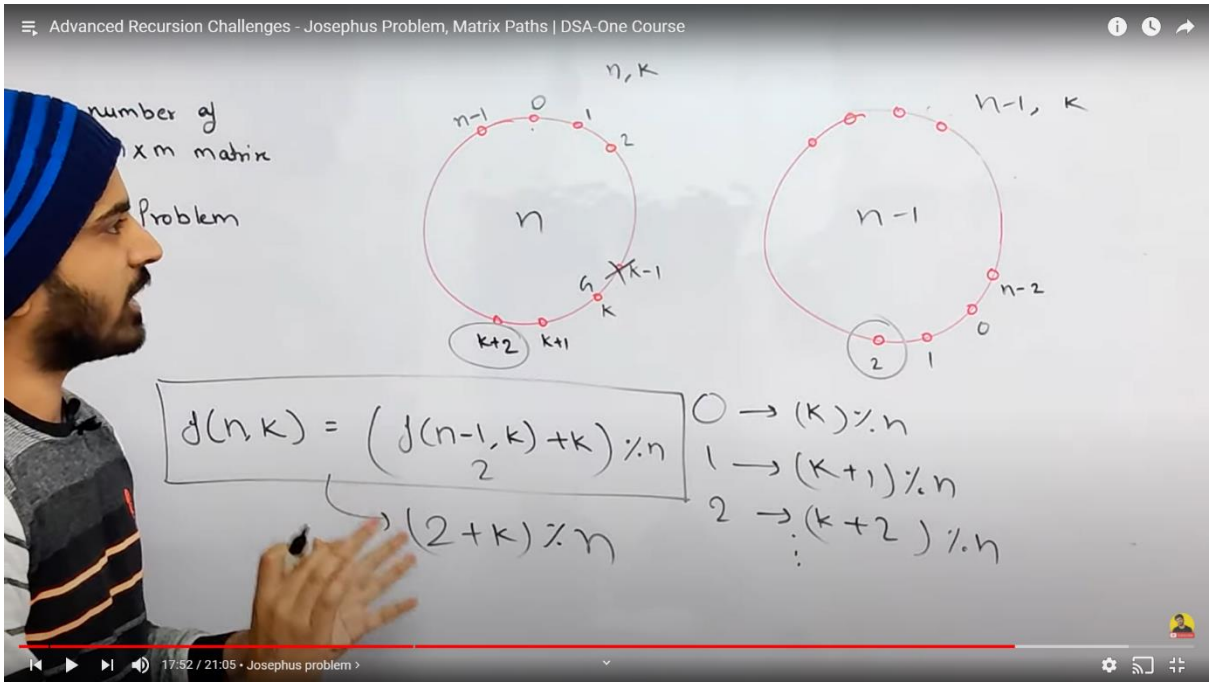
There are n people standing in a circle waiting to be executed. The counting out begins at some point in the circle and proceeds around the circle in a fixed direction. In each step, a certain number of people are skipped and the next person is executed. The elimination proceeds around the circle (which is becoming smaller and smaller as the executed people are removed), until only the last person remains, who is given freedom. Given the total number of persons n and a number k which indicates that k-1 persons are skipped and kth person is killed in circle. The task is to choose the place in the initial circle so that you are the last one remaining and so survive.

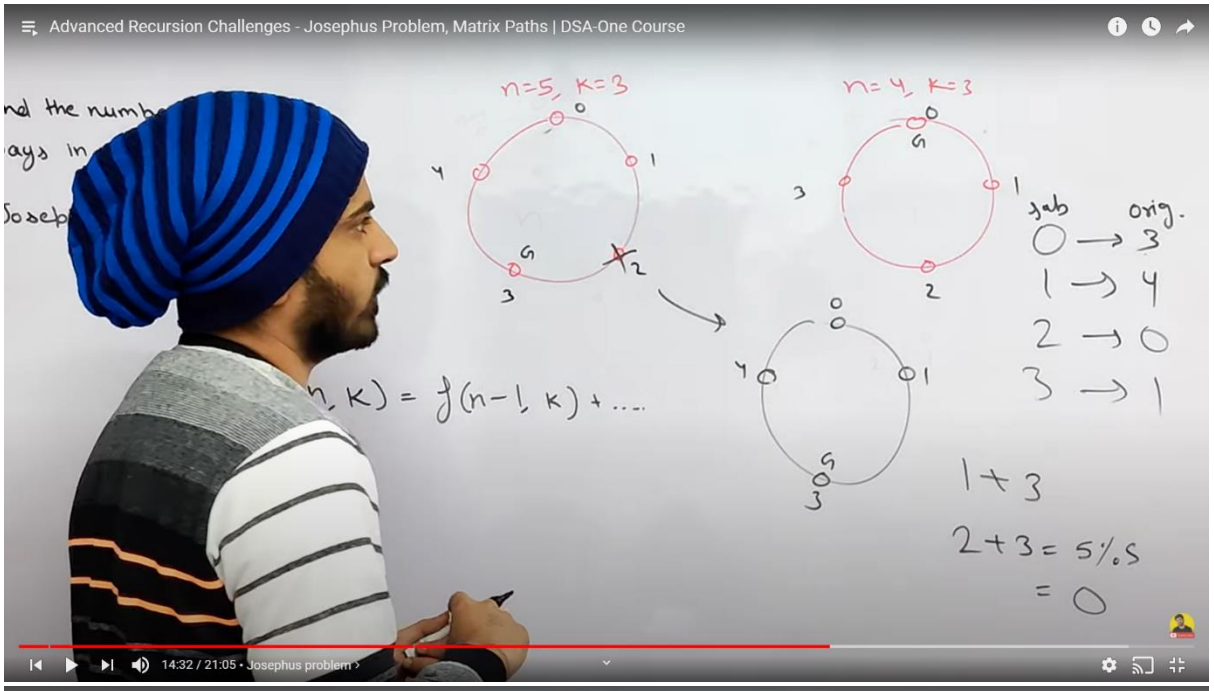
For example, if n = 5 and k = 3, then the safe position is 3. Firstly, the person at position 2 is killed, then person at position 4 is killed, then person at position 1 is killed. Finally, the person at position 5 is killed. So the person at position 3 survives. If n = 7 and k = 4, then the safe position is 4. The persons at positions 3, 6, 2, 7, 5, 1 are killed in order, and person at position 4 survives.

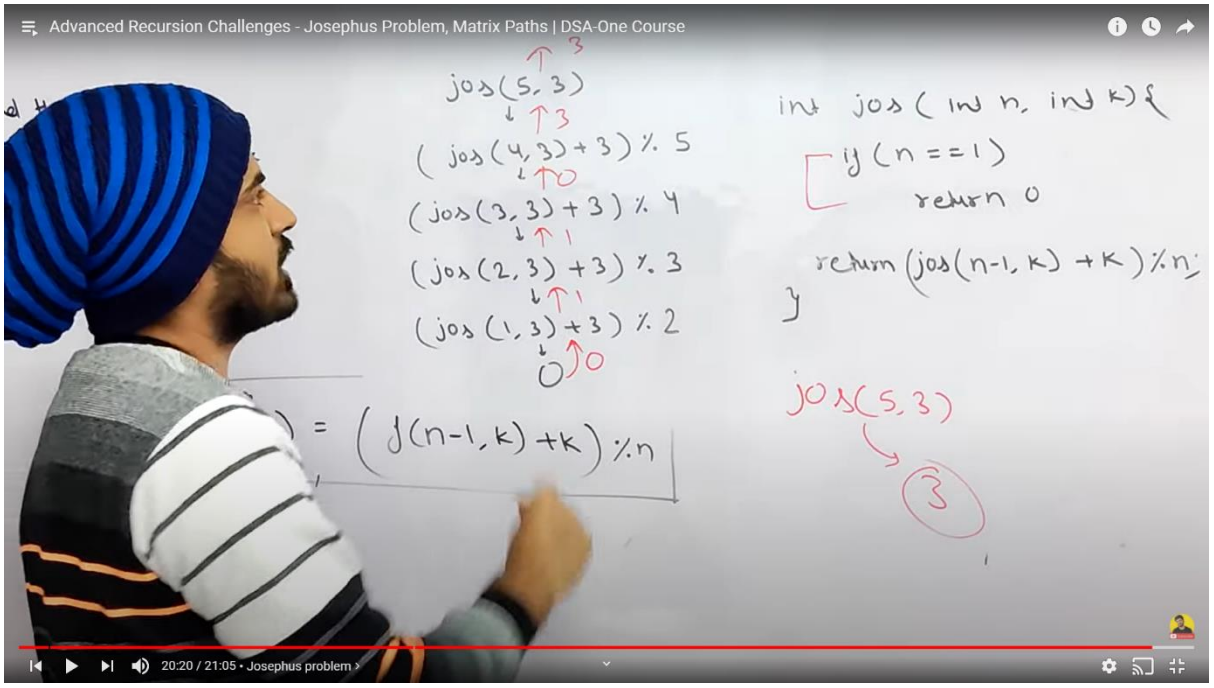
The problem has following recursive structure.

```
josephus(n, k) = ((josephus(n - 1, k) + k) % n );
josephus(1, k) = 0 → index 0 → person 1.
```

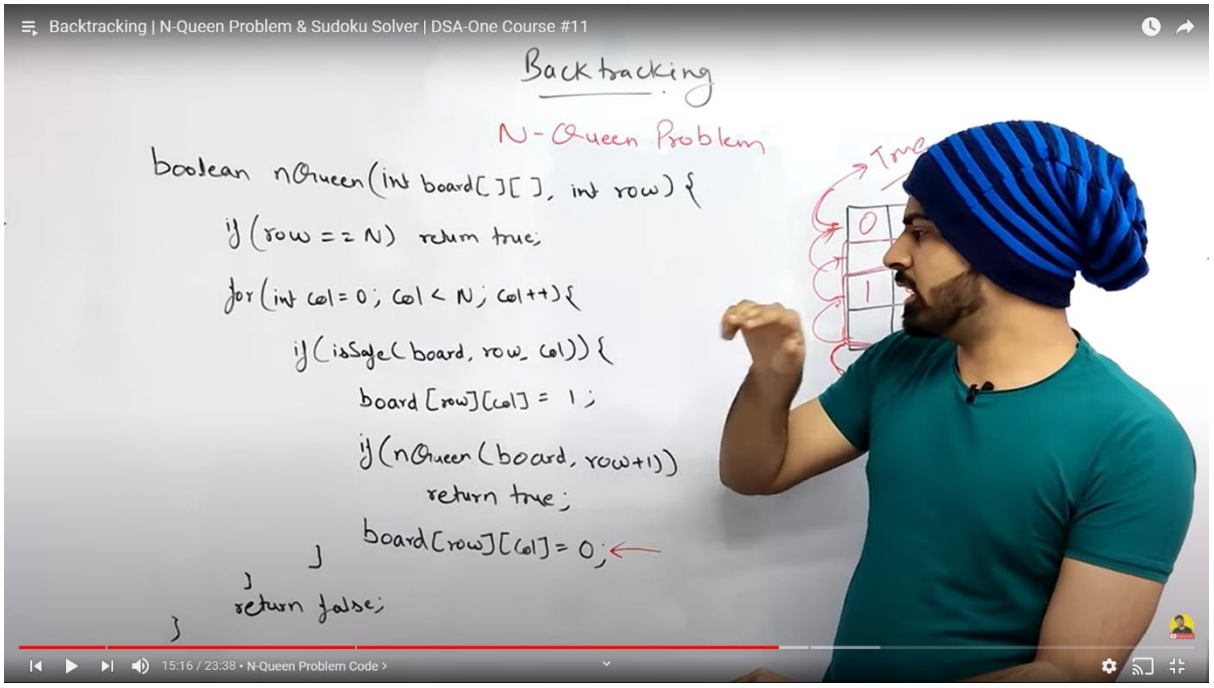
After the first person (kth from beginning) is killed, n-1 persons are left. So we call josephus(n – 1, k) to get the position with n-1 persons. But the position returned by josephus(n – 1, k) will consider the position starting from k%n . So, we must make adjustments to the position returned by josephus(n – 1, k).




Advanced Recursion Challenges - Josephus Problem, Matrix Paths | DSA-One Course
n=5, K=3
n=4, K=3
sub orig.
0 → 3
1 → 4
2 → 0
3 → 1
1+3
2+3 = 5%5
= 0
14:32 / 21:05 • Josephus problem


Advanced Recursion Challenges - Josephus Problem, Matrix Paths | DSA-One Course
jos(5,3)
(jos(4,3)+3)%5
(jos(3,3)+3)%4
(jos(2,3)+3)%3
(jos(1,3)+3)%2
int jos(int n, int k){
if(n==1)
return 0
return (jos(n-1,k)+k)%n;
}
jos(5,3)
3
20:20 / 21:05 • Josephus problem


05/02/21
2:24 AM
MAIN CAMERA


Backtracking | N-Queen Problem & Sudoku Solver | DSA-One Course #11
Backtracking
N-Queen Problem
boolean nQueen(int board[][], int row){
if(row == N) return true;
for(int col=0; col<N; col++){
if(isSafe(board, row, col)){
board[row][col] = 1;
if(nQueen(board, row+1))
return true;
board[row][col] = 0;
}
}
return false;
}
15:16 / 23:38 • N-Queen Problem Code ›