

# Josephus problem

From Wikipedia, the free encyclopedia

In computer science and mathematics, the **Josephus Problem** (or **Josephus permutation**) is a theoretical problem related to a certain counting-out game.

There are 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.

The task is to choose the place in the initial circle so that you are the last one remaining and so survive.

## Contents

- 1 History
- 2 Solution
  - 2.1  $k=2$
  - 2.2 The general case
- 3 Variants and generalizations
  - 3.1 Extended Josephus problem
- 4 Notes
- 5 References
- 6 External links

## History

The problem is named after Flavius Josephus, a Jewish historian living in the 1st century. According to Josephus' account of the siege of Yodfat, he and his 40 soldiers were trapped in a cave, the exit of which was blocked by Romans. They chose suicide over capture and decided that they would form a circle and start killing themselves using a step of three. Josephus states that by luck or possibly by the hand of God, he and another man remained the last and gave up to the Romans.

The reference comes from Book 3, Chapter 8, par 7 of Josephus' The Jewish War (writing of himself in the third person):

However, in this extreme distress, he was not destitute of his usual sagacity; but trusting himself to the providence of God, he put his life into hazard [in the manner following]: "And now," said he, "since it is resolved among you that you will die, come on, let us commit our mutual deaths to determination by lot. He whom the lot falls to first, let him be killed by him that hath the second lot, and thus fortune shall make its progress through us all; nor shall any of us perish by his own right hand, for it would be unfair if, when the rest are gone, somebody should repent and save himself." This proposal appeared to them to be very just; and when he had prevailed with them to determine this matter by lots, he drew one of the lots for himself also. He who had the first lot laid his neck bare to him that had the next, as supposing that the general would die among them immediately; for they thought death, if Josephus might but die with them, was sweeter than life; yet was he with another left to the last, whether we must say it happened so by chance, or whether by the providence of God. And as he was very desirous neither to be condemned by the lot, nor, if he had been left to the last, to imbrue his right hand in the blood of his countrymen, he persuaded him to trust his fidelity to him, and to live as well as himself.<sup>[1]</sup>

## Solution

In the following,  $n$  denotes the number of people in the initial circle, and  $k$  denotes the count for each step, that is,  $k - 1$  people are skipped and the  $k$ -th is executed. The people in the circle are numbered from 1 to  $n$ .

### $k=2$

We explicitly solve the problem when every 2nd person will be killed, i.e.  $k = 2$ . (For the more general case  $k \neq 2$ , we outline a solution below.) We express the solution recursively. Let  $f(n)$  denote the position of the survivor when there are initially  $n$  people (and  $k = 2$ ). The first time around the circle, all of the even-numbered people die. The second time around the circle, the new 2nd person dies, then the new 4th person, etc.; it's as though there were no first time around the circle.

If the initial number of people was even, then the person in position  $x$  during the second time around the circle was originally in position  $2x - 1$  (for every choice of  $x$ ). Let  $n = 2j$ . The person at  $f(j)$  who will now survive was originally in position  $2f(j) - 1$ . This gives us the recurrence

$$f(2j) = 2f(j) - 1 .$$

If the initial number of people was odd, then we think of person 1 as dying at the end of the first time around the circle. Again, during the second time around the circle, the new 2nd person dies, then the new 4th person, etc. In this case, the person in position  $x$  was originally in position  $2x + 1$ . This gives us the recurrence

$$f(2j + 1) = 2f(j) + 1 .$$

When we tabulate the values of  $n$  and  $f(n)$  we see a pattern:

$n$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
$f(n)$	1	1	3	1	3	5	7	1	3	5	7	9	11	13	15	1

This suggests that  $f(n)$  is an increasing odd sequence that restarts with  $f(n) = 1$  whenever the index  $n$  is a power of 2. Therefore, if we choose  $m$  and  $l$  so that  $n = 2^m + l$  and  $0 \leq l < 2^m$ , then  $f(n) = 2 \cdot l + 1$ . It is clear that values in the table satisfy this equation. Or we can think that after  $l$  people are dead there are only  $2^m$  people and we go to the  $2l + 1$ th person. He must be the survivor. So  $f(n) = 2l + 1$ . Below, we give a proof by induction.

**Theorem:** If  $n = 2^m + l$  and  $0 \leq l < 2^m$ , then  $f(n) = 2l + 1$ .

**Proof:** We use strong induction on  $n$ . The base case  $n = 1$  is true. We consider separately the cases when  $n$  is even and when  $n$  is odd.

If  $n$  is even, then choose  $l_1$  and  $m_1$  such that  $n/2 = 2^{m_1} + l_1$  and  $0 \leq l_1 < 2^{m_1}$ . Note that  $l_1 = l/2$ . We have  $f(n) = 2f(n/2) - 1 = 2((2l_1) + 1) - 1 = 2l + 1$ , where the second equality follows from the induction hypothesis.

If  $n$  is odd, then choose  $l_1$  and  $m_1$  such that  $(n - 1)/2 = 2^{m_1} + l_1$  and  $0 \leq l_1 < 2^{m_1}$ . Note that  $l_1 = (l - 1)/2$ . We have  $f(n) = 2f((n - 1)/2) + 1 = 2((2l_1) + 1) + 1 = 2l + 1$ , where the second equality follows from the induction hypothesis. This completes the proof.

We can solve for  $l$  to get an explicit expression for  $f(n)$ :

$$f(n) = 2(n - 2^{\lfloor \log_2(n) \rfloor}) + 1$$

The most elegant form of the answer involves the binary representation of size  $n$ :  $f(n)$  can be obtained by a one-bit left cyclic shift of  $n$  itself. If we represent  $n$  in binary as  $n = 1b_1b_2b_3 \dots b_m$ , then the solution is given by  $f(n) = b_1b_2b_3 \dots b_m1$ . The proof of this follows from the representation of  $n$  as  $2^m + l$  or from the above expression for  $f(n)$ .

**Implementation:** A simple python function can be:

```
def josephus_2( n ):
    from math import log
    return 2*(n - 2**(int(log(n, 2))))+1
```

## The general case

The easiest way to solve this problem in the general case is to use dynamic programming by performing the first step and then using the solution of the remaining problem. When the index starts from one, then the person at  $s$  shifts from the first person is in position  $((s - 1) \bmod n) + 1$ , where  $n$  is the total number of persons. Let  $f(n, k)$  denote the position of the survivor. After the  $k$ -th person is killed, we're left with a circle of  $n - 1$ , and we start the next count with the person whose number in the original problem was  $(k \bmod n) + 1$ . The position of the survivor in the remaining circle would be  $f(n - 1, k)$  if we start counting at 1; shifting this to account for the fact that we're starting at  $(k \bmod n) + 1$  yields the recurrence

$$f(n, k) = ((f(n - 1, k) + k - 1) \bmod n) + 1, \text{ with } f(1, k) = 1,$$

which takes the simpler form

$$g(n, k) = (g(n - 1, k) + k) \bmod n, \text{ with } g(1, k) = 0$$

if we number the positions from 0 to  $n - 1$  instead.

This approach has running time  $O(n)$ , but for small  $k$  and large  $n$  there is another approach. The second approach also uses dynamic programming but has running time  $O(k \log n)$ . It is based on considering killing  $k$ -th,  $2k$ -th, ...,  $(\lfloor n/k \rfloor k)$ -th people as one step, then changing the numbering.

**Implementation:** A simple python function can be (but might be slow/fail for large  $n$  and small  $k$ ):

```
def josephus(n, k):
    if n == 1:
        return 1
    else:
        return ((josephus(n-1, k) + k - 1) % n) + 1
```

Alternatively a simple loop can also be run like:

```
def josephus(n, k):
    r = 0
    i = 1
    while i <= n:
        r = (r + k) % i
        i += 1
    return r + 1
```

A dynamic programming solution in Haskell:

```

josephus n k = josephus' [1..n] k where
  josephus' xs k
    | length xs == 1 = head xs
    | otherwise      = josephus' (killnext k xs) k where
      killnext k xs = take (length xs - 1) (drop k (cycle xs))

```

A somewhat more elaborate version in Java:

```

int josephus(int n, int k) {
    return josephus(n, k, 1);
}
int josephus(int n, int k, int startingPoint) {
    if(n == 1)
        return 1;
    int newSp = (startingPoint + k - 2) % n + 1;

    int survivor = josephus(n - 1, k, newSp);
    if (survivor < newSp) {
        return survivor;
    } else
        return survivor + 1;
}

```

## Variants and generalizations

According to Concrete Mathematics, section 1.3, Josephus had an accomplice; the problem was then to find the places of the two last remaining survivors (whose conspiracy would ensure their survival).

## Extended Josephus problem

Problem definition: There are  $n$  persons, numbered 1 to  $n$ , around a circle. We eliminate second of every two remaining persons until one person remains. Given the  $n$ , determine the number of  $x$ th person who is eliminated.<sup>[2]</sup>

## Notes

1. ^ Flavius Josephus, Wars Of The Jews III. 8. 7. (<http://www.gutenberg.org/dirs/2/8/5/2850/2850-h/book3.htm#2HCH0008>) (Translation by William Whiston).
2. ^ Armin Shams-Baragh Formulating The Extended Josephus Problem (<http://www.cs.man.ac.uk/~shamsbaa/Josephus.pdf>).

## References

- Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein (2001). "Chapter 14: Augmenting Data Structures". Introduction to Algorithms (Second Edition ed.). MIT Press and McGraw-Hill. p. 318. ISBN 0-262-03293-7.

## External links

- Josephus Flavius game (<http://www.cut-the-knot.org/recurrence/flavius.shtml>) (Java Applet) at cut-the-knot allowing selection of every  $n^{\text{th}}$  out of 50 (maximum).
- Josephus Problem at the MathWorld encyclopedia (<http://mathworld.wolfram.com/JosephusProblem.html>)
- Josephus Problem at Shippensburg University (<http://www.maa.org/publications/periodicals/loci/resources/the-josephus-problem>)

Retrieved from "http://en.wikipedia.org/w/index.php?title=Josephus\_problem&oldid=591707811"

Categories: Combinatorics | Permutations | Mathematical problems | Computational problems

---

- This page was last modified on 21 January 2014 at 12:57.
- Text is available under the Creative Commons Attribution-ShareAlike License; additional terms may apply. By using this site, you agree to the Terms of Use and Privacy Policy. Wikipedia® is a registered trademark of the Wikimedia Foundation, Inc., a non-profit organization.