THE UNIVERSITY OF VERMONT
COLLEGE OF ENGINEERING &
MATHEMATICAL SCIENCES

Hash Tables: Linear Probing

CS 124 / Department of Computer Science

Linear probing

Earlier, we saw our first collision resolution policy, separate chaining.

Linear probing is another approach to resolving hash collisions.

Unlike separate chaining, we only allow a single object at a given index.

Linear probing

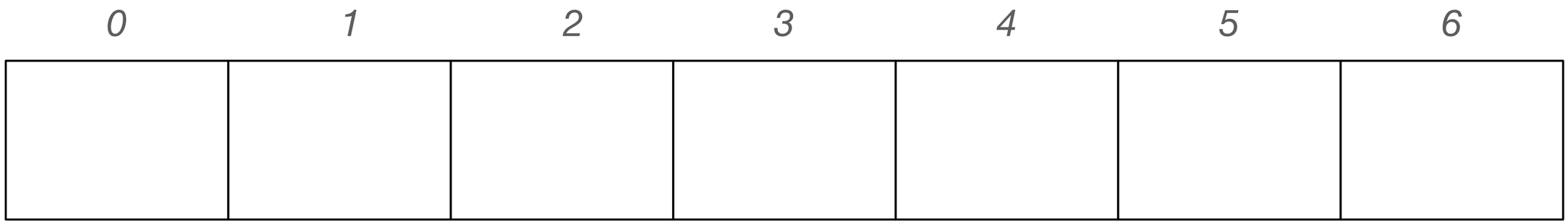
Earlier, we saw our first collision resolution policy, separate chaining.

Linear probing is another approach to resolving hash collisions.

Unlike separate chaining, we only allow a single object at a given index.

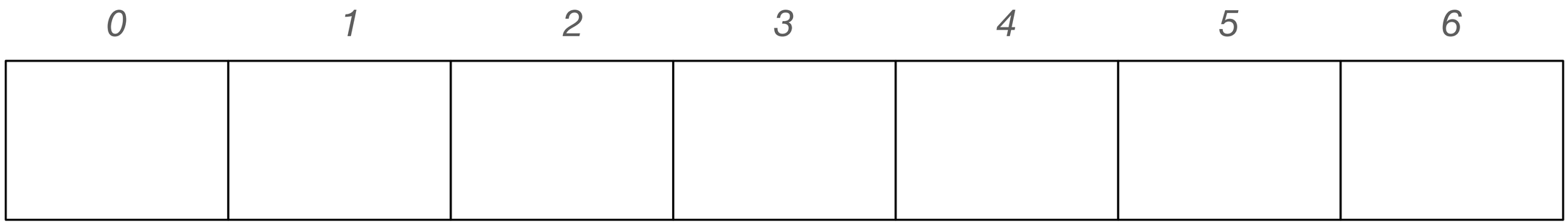
The idea behind linear probing is simple: if a collision occurs, we probe our hash table taking one step at a time until we find an empty spot for the object we wish to insert.

Linear probing



Linear probing

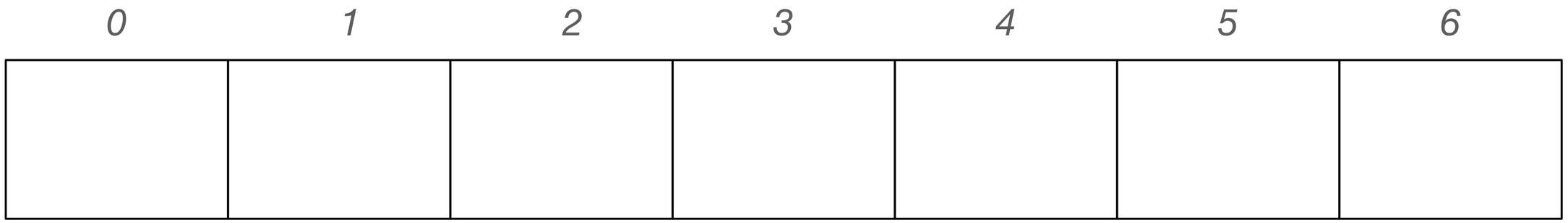
Hash function:
$f(x) = x \bmod 7$



Linear probing

Hash function:
$f(x) = x \bmod 7$

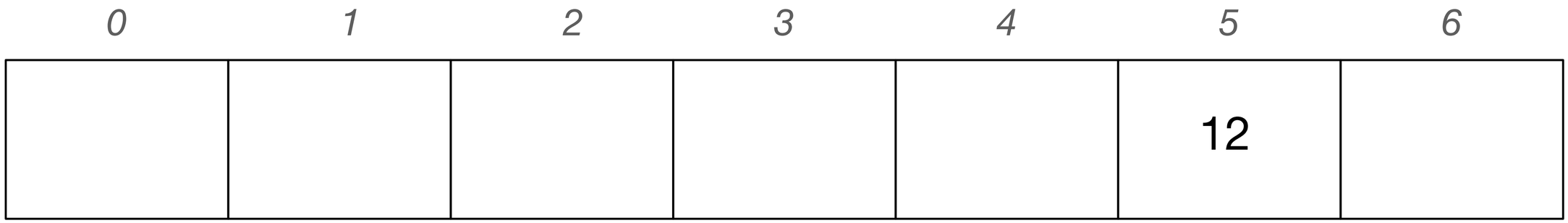
Insert 12 : 12 mod 7 = 5



Linear probing

Hash function:
$f(x) = x \bmod 7$

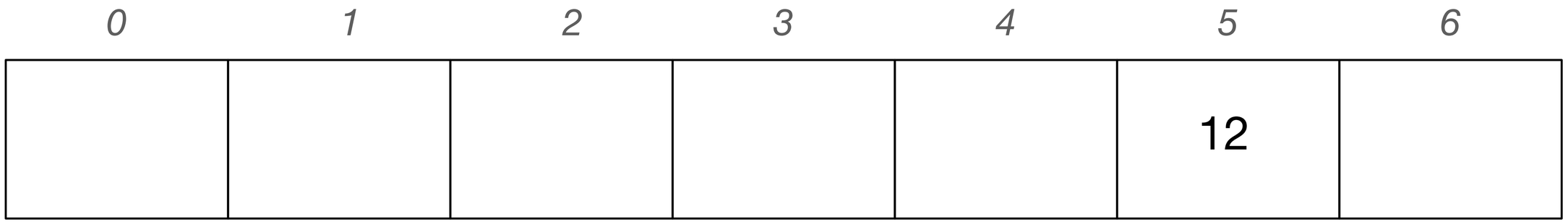
Insert 12 : 12 mod 7 = 5



Linear probing

Hash function:
$f(x) = x \bmod 7$

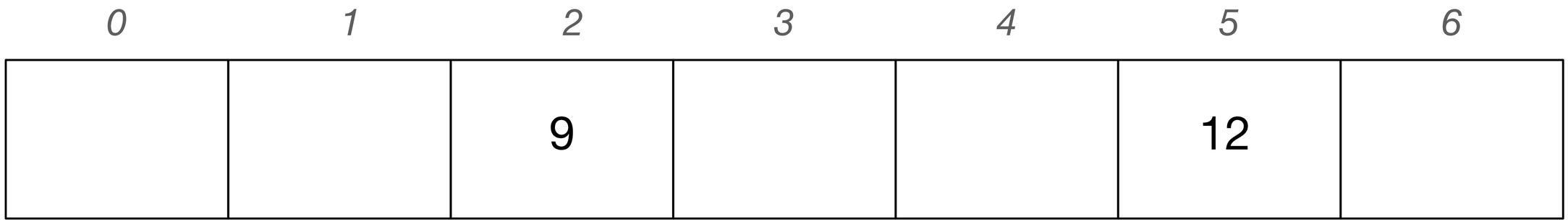
Insert 9 : 9 mod 7 = 2



Linear probing

Hash function:
$f(x) = x \bmod 7$

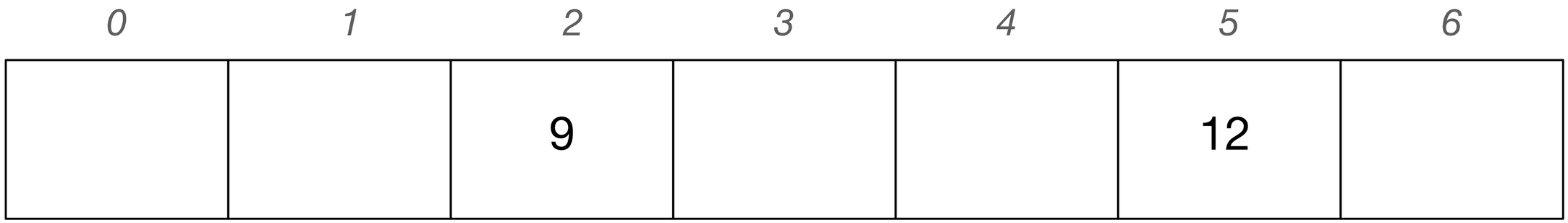
Insert 9 : 9 mod 7 = 2



Linear probing

Hash function:
$f(x) = x \bmod 7$

Insert 44 : 44 mod 7 = 2

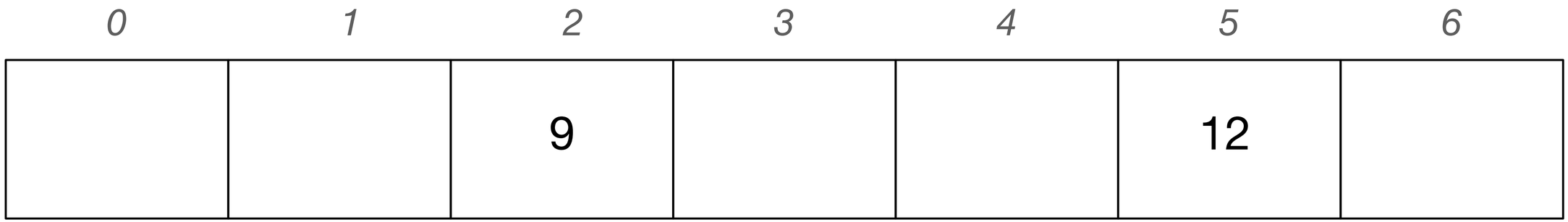


Linear probing

Hash function:
$f(x) = x \bmod 7$

Insert 44 : 44 mod 7 = 2

collision!

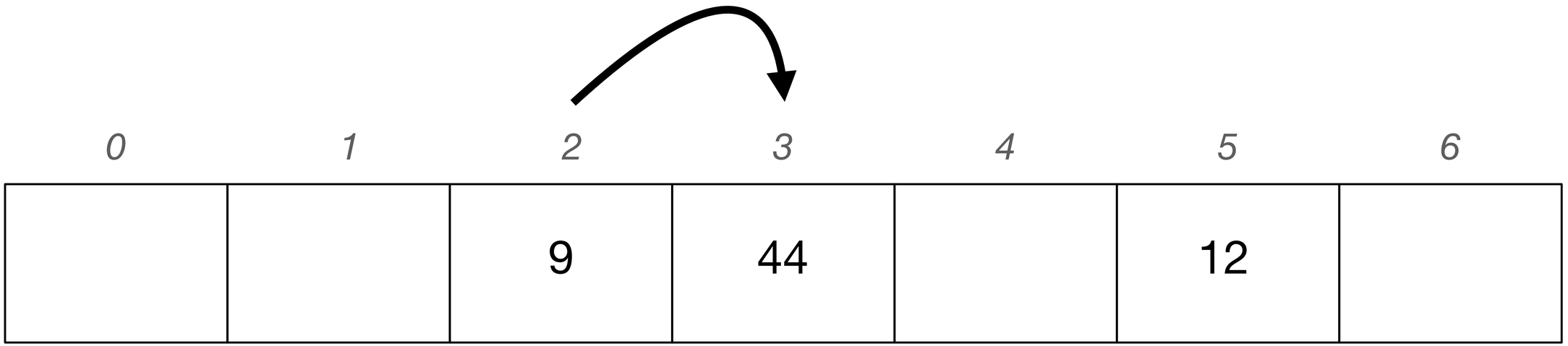


Linear probing

Hash function:
$f(x) = x \bmod 7$

Insert 44 : 44 mod 7 = 2

collision!

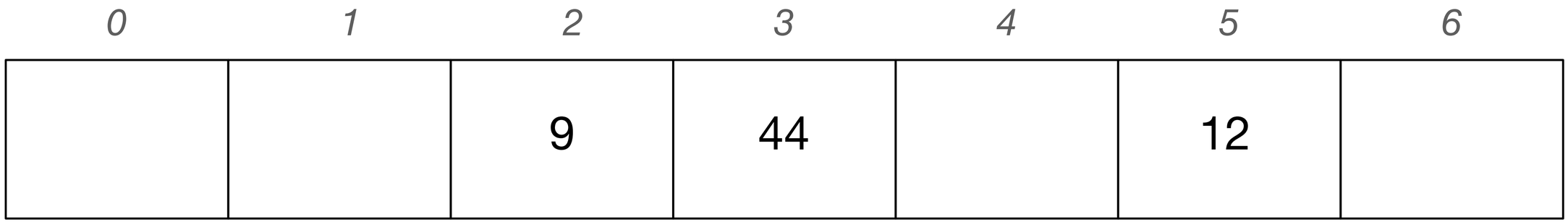


Linear probing

Hash function:
$f(x) = x \bmod 7$

Insert 30 : 30 mod 7 = 2

collision!

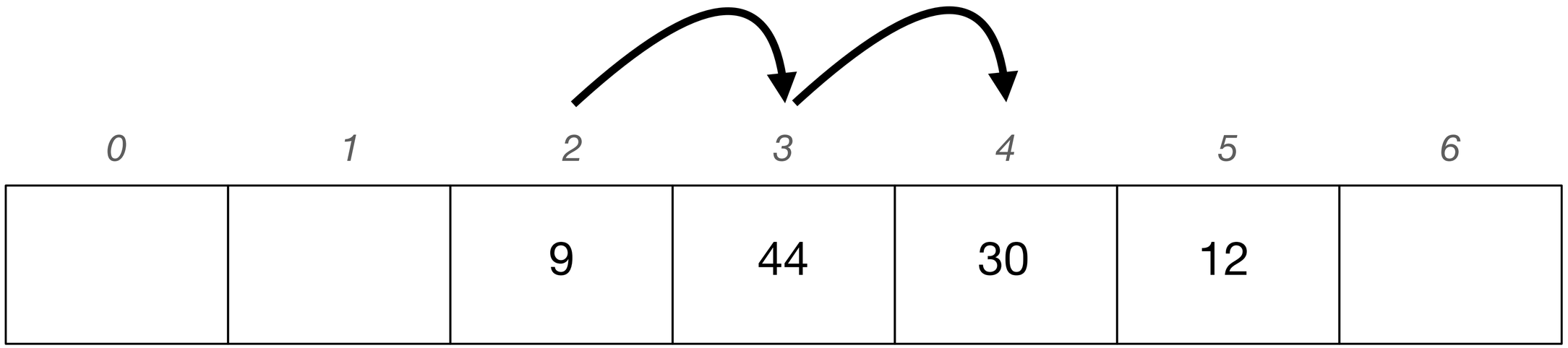


Linear probing

Hash function:
$f(x) = x \bmod 7$

Insert 30 : 30 mod 7 = 2

collision!

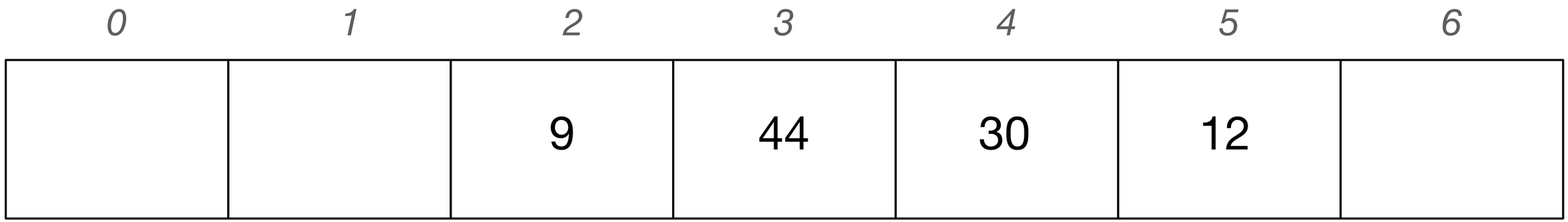


Linear probing

Hash function:
$f(x) = x \bmod 7$

Insert 2 : 2 mod 7 = 2

collision!

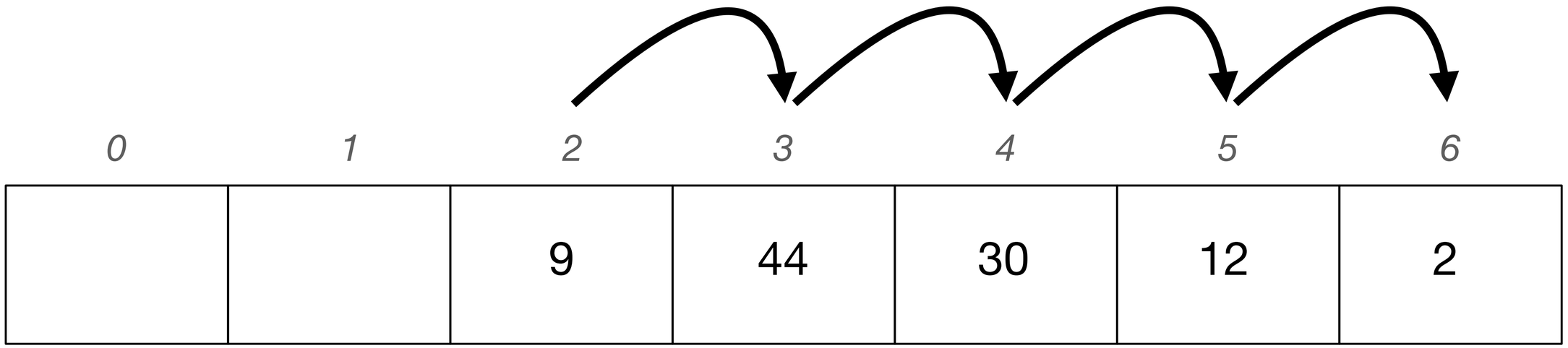


Linear probing

Hash function:
$f(x) = x \bmod 7$

Insert 2 : 2 mod 7 = 2

collision!

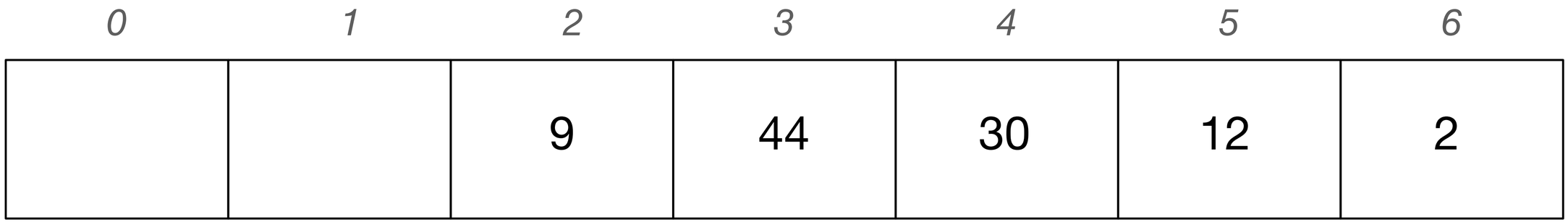


Linear probing

Hash function:
$f(x) = x \bmod 7$

Insert 16 : 16 mod 7 = 2

collision!

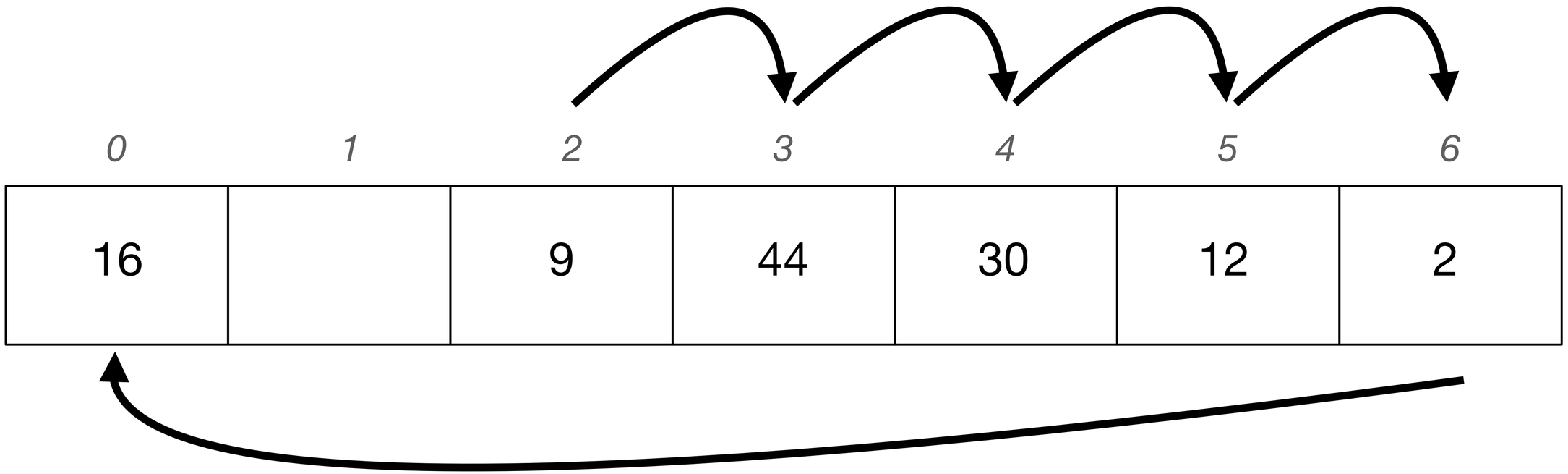


Linear probing

Hash function:
$f(x) = x \bmod 7$

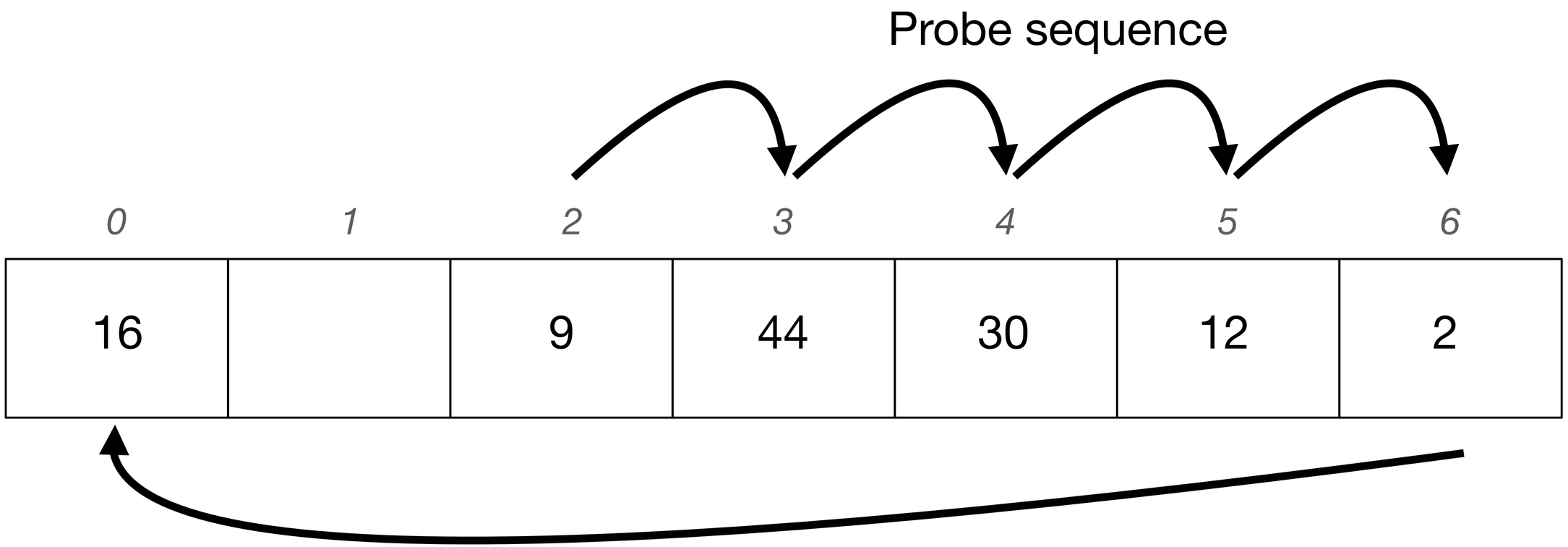
Insert 16 : 16 mod 7 = 2

collision!



Linear probing

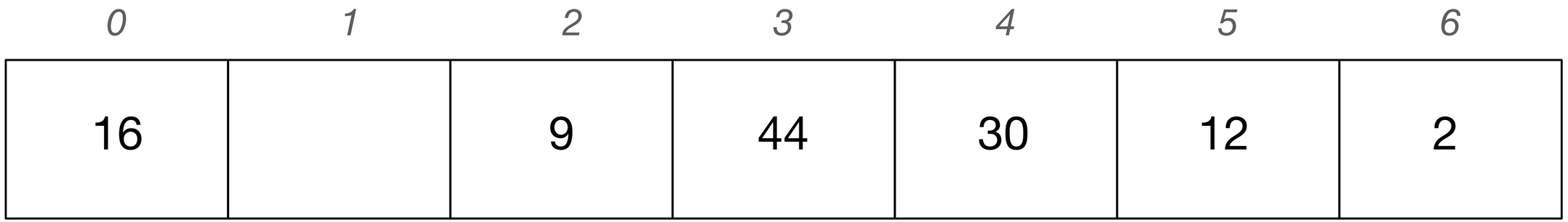
Hash function:
$f(x) = x \bmod 7$



Linear probing

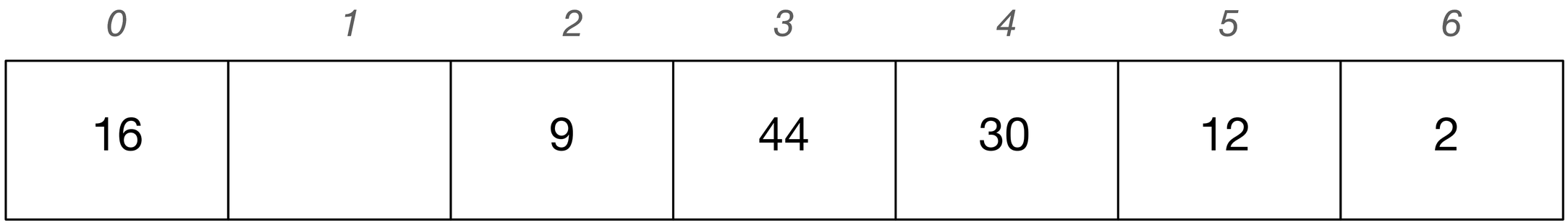
Hash function:
$f(x) = x \bmod 7$

key	probe sequence
9	2
44	2, 3
30	2, 3, 4
2	2, 3, 4, 5, 6
16	2, 3, 4, 5, 6, 0



Linear probing

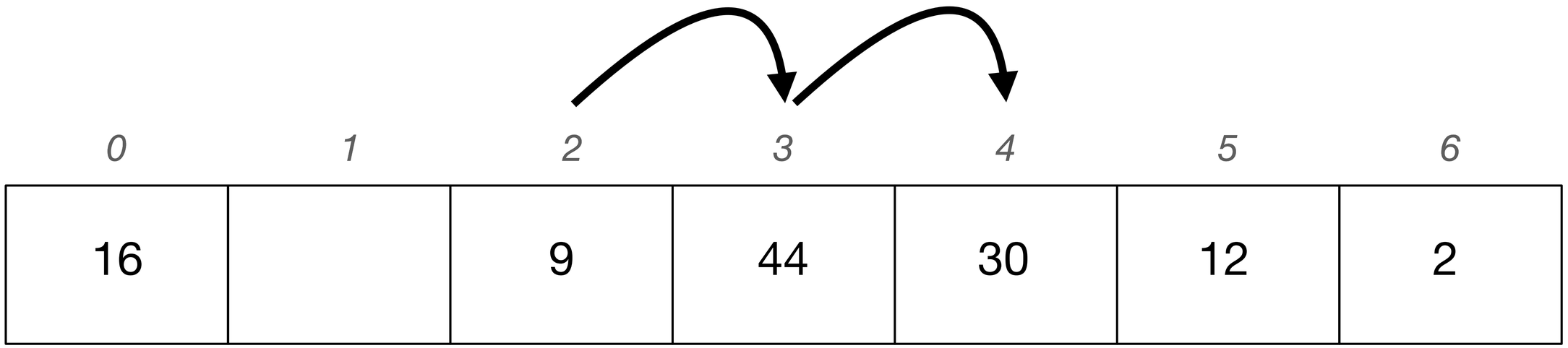
Hash function:
$f(x) = x \bmod 7$



Linear probing

Hash function:
$f(x) = x \bmod 7$

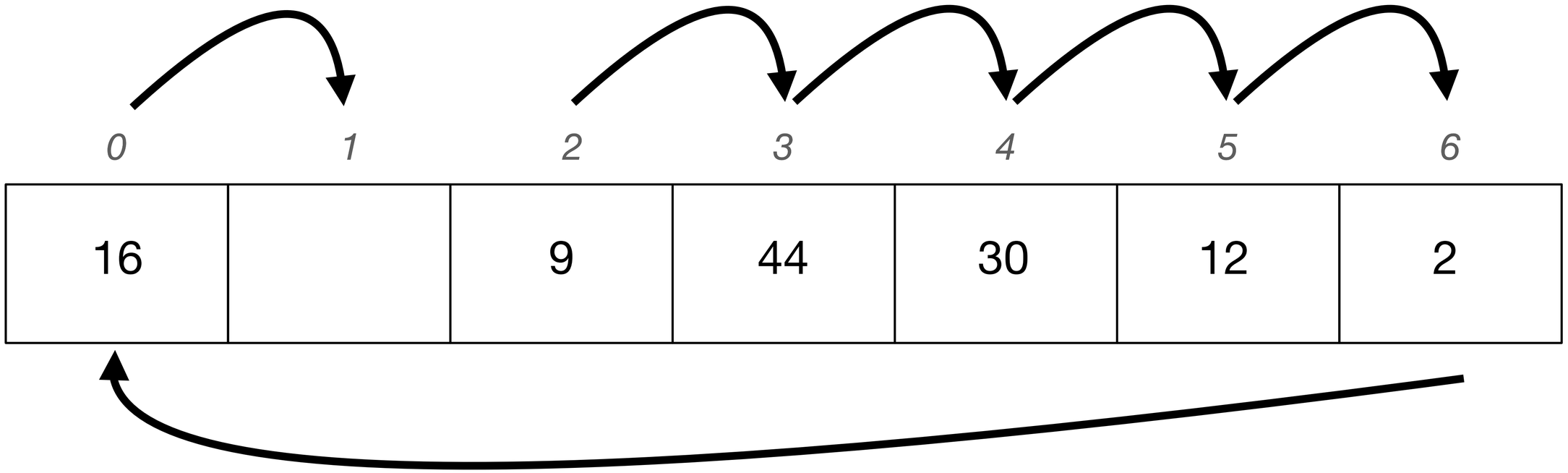
Find 30 : 30 mod 7 = 2



Linear probing

Hash function:
$f(x) = x \bmod 7$

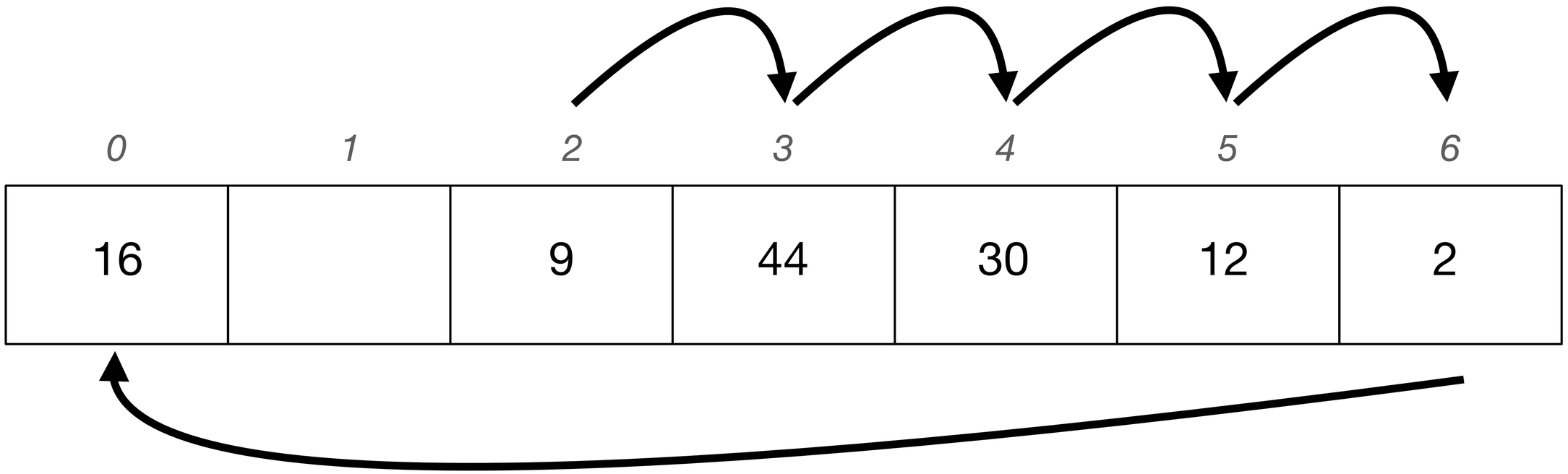
Find 72 : 72 mod 7 = 2



Linear probing

Hash function:
$f(x) = x \bmod 7$

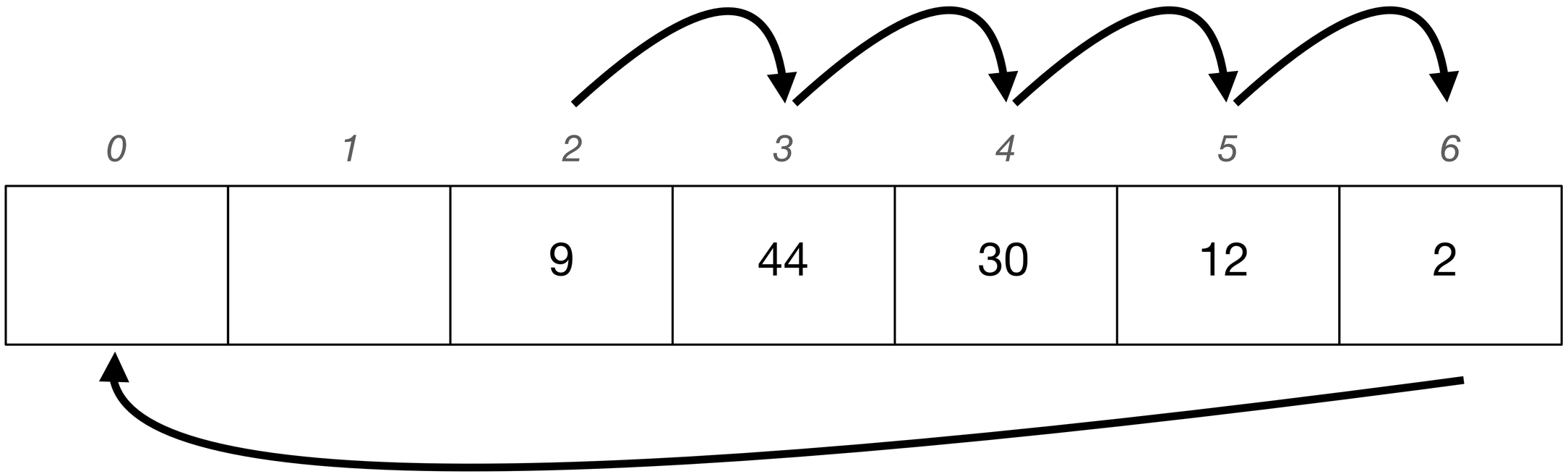
Remove 16 : 16 mod 7 = 2



Linear probing

Hash function:
$f(x) = x \bmod 7$

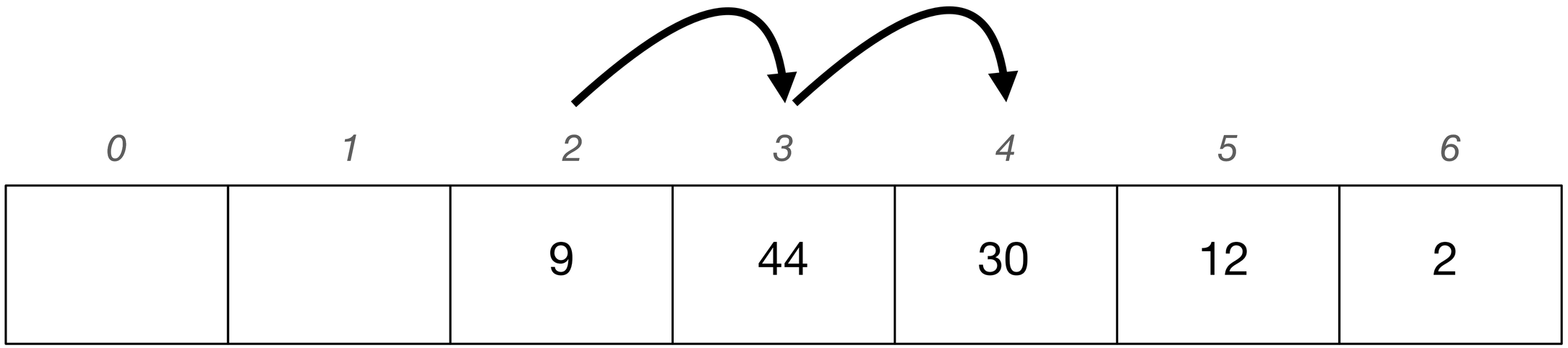
Remove 16 : 16 mod 7 = 2



Linear probing

Hash function:
$f(x) = x \bmod 7$

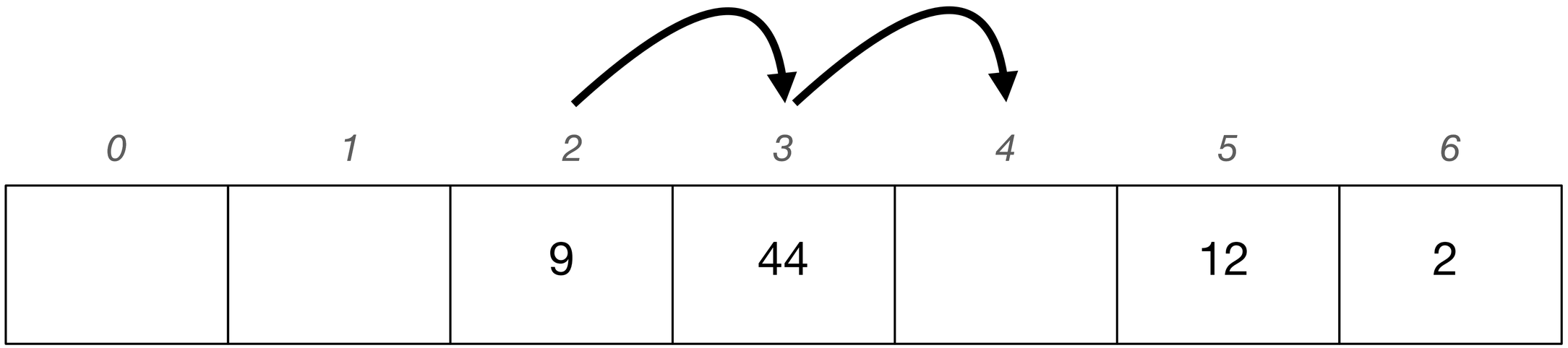
Remove 30 : 30 mod 7 = 2



Linear probing

Hash function:
$f(x) = x \bmod 7$

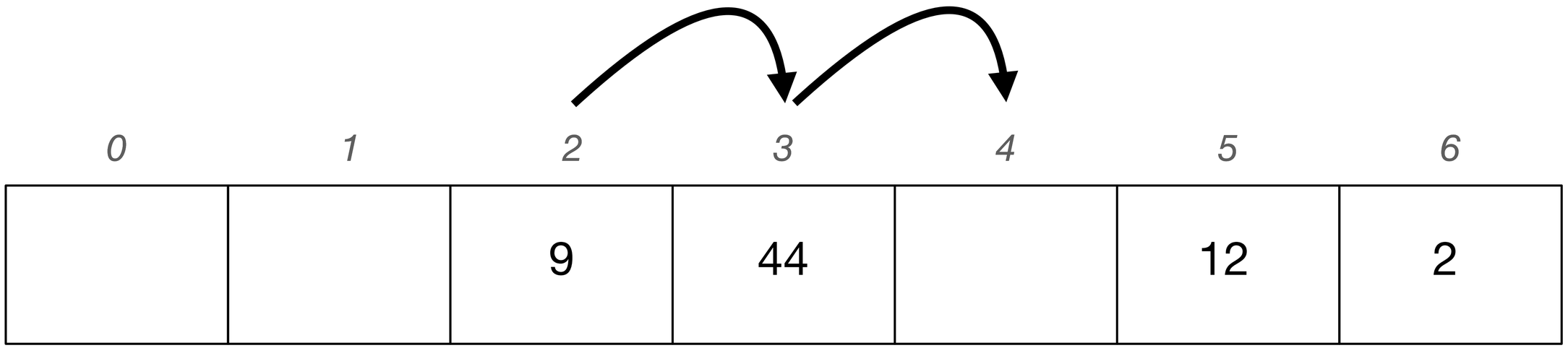
Remove 30 : 30 mod 7 = 2



Linear probing

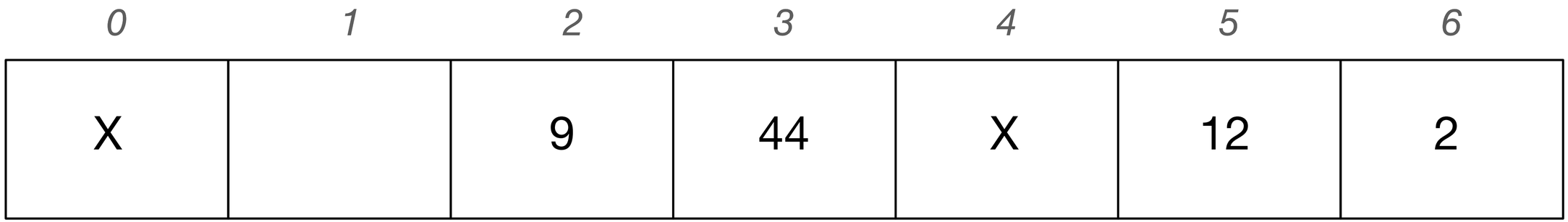
Hash function:
$f(x) = x \bmod 7$

Find 2 : 2 mod 7 = 2



Linear probing

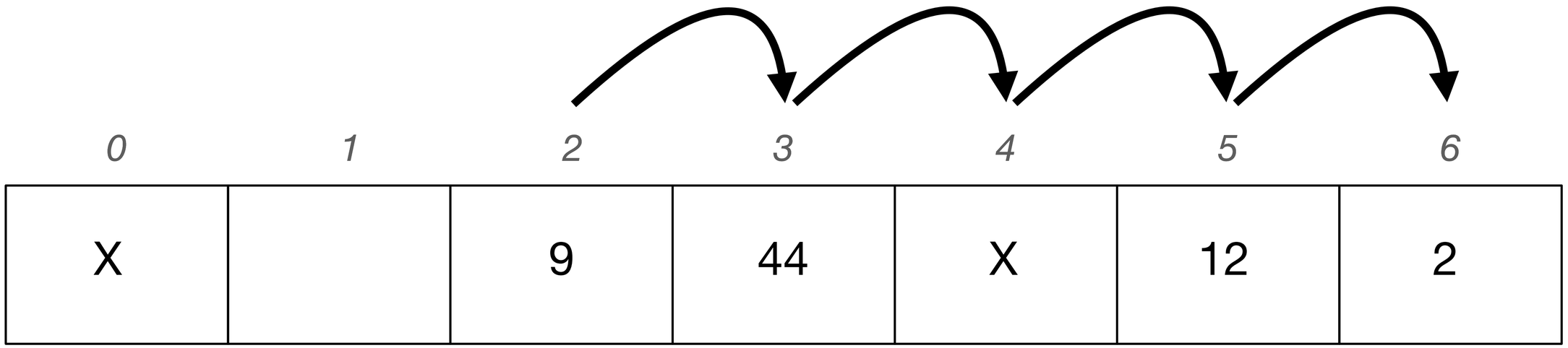
Hash function:
$f(x) = x \bmod 7$



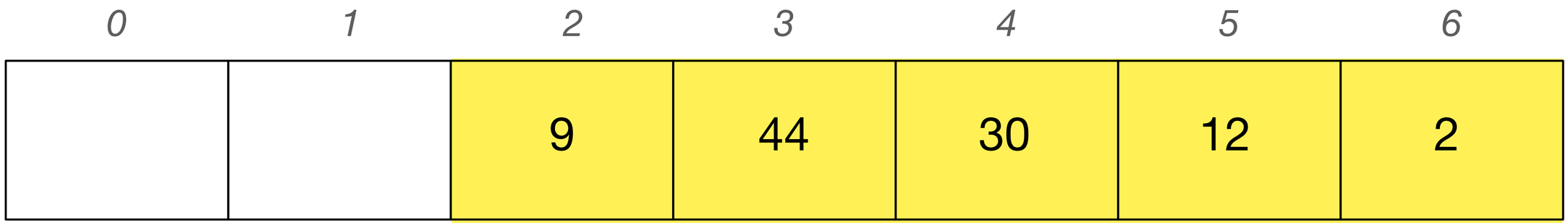
Linear probing

Hash function:
$f(x) = x \bmod 7$

Find 2 : 2 mod 7 = 2



Linear probing: primary clustering



Linear probing vs separate chaining

Linear probing	Separate chaining
On collisions we probe	On collisions we extend the chain
Fixed upper limit on number of objects we can insert (size of hash table)	Because we can extend chains, we are only limited by memory / system constrants as to how many objects we can isert
Prone to primary clustering	Clustering does not occur

Linear probing: summary

- We only allow a single object at a given index.
- Upon hash collisions, we probe our hash table, one step at a time, until we find an empty position in which we may insert our object.
- Unlike separate chaining, where we can extend chains, linear probing has a fixed limit on the number of objects we can insert into our hash table.

Questions

- What can we do when we run out of space in our hash table?
- If we set our stride to some value greater than one, why is it a good idea to have a hash table size that's a prime number?