http://algs4.cs.princeton.edu

3.4 LINEAR PROBING DEMO

3.4 Linear Probing Demo

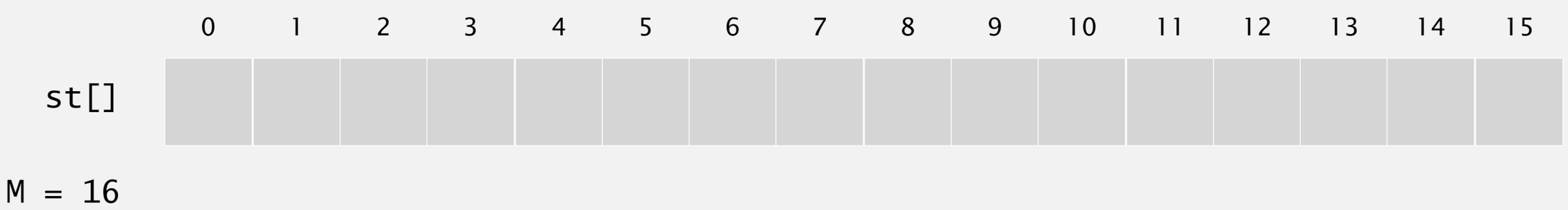
- *insert*
- *search*

Linear-probing hash table demo: insert

Hash. Map key to integer i between 0 and M-1.

Insert. Put at table index i if free; if not try i+1, i+2, etc.

linear-probing hash table



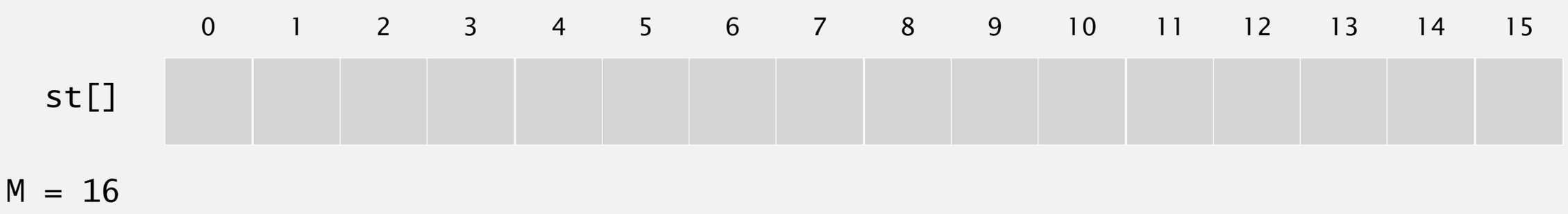
Linear-probing hash table demo: insert

Hash. Map key to integer i between 0 and M-1.

Insert. Put at table index i if free; if not try i+1, i+2, etc.

insert S

hash(S) = 6



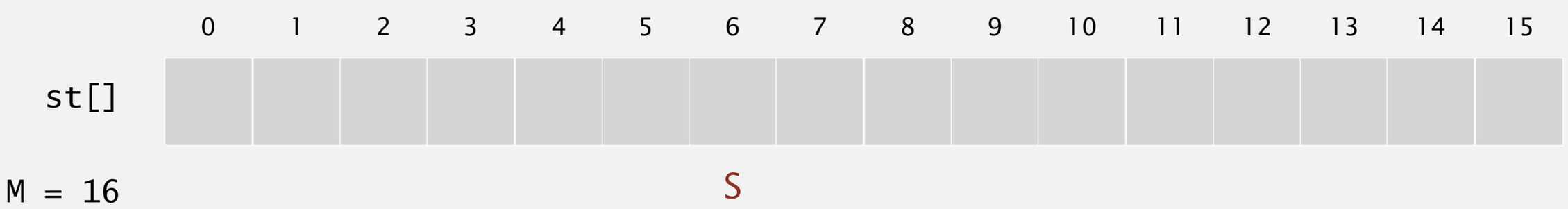
Linear-probing hash table demo: insert

Hash. Map key to integer i between 0 and M-1.

Insert. Put at table index i if free; if not try i+1, i+2, etc.

insert S

hash(S) = 6



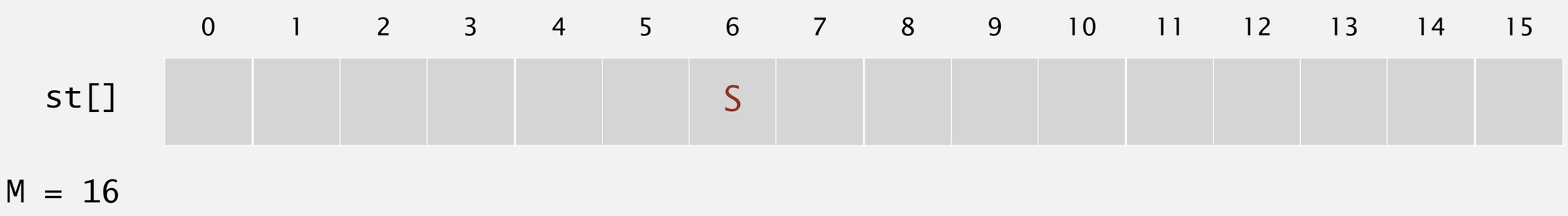
Linear-probing hash table demo: insert

Hash. Map key to integer i between 0 and M-1.

Insert. Put at table index i if free; if not try i+1, i+2, etc.

insert S

hash(S) = 6

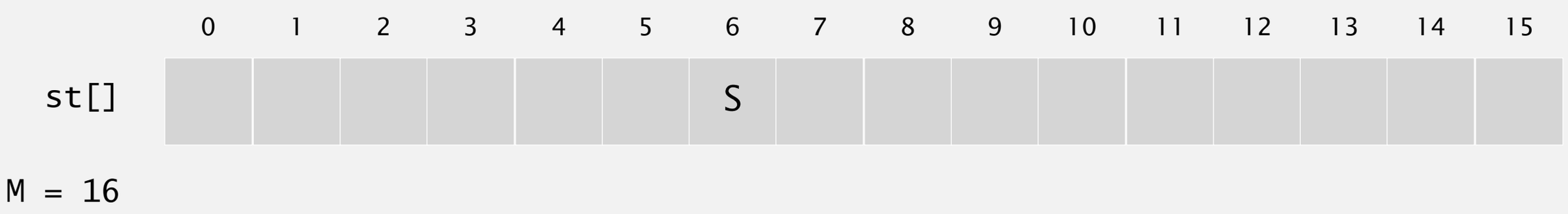


Linear-probing hash table demo: insert

Hash. Map key to integer i between 0 and M-1.

Insert. Put at table index i if free; if not try i+1, i+2, etc.

linear-probing hash table



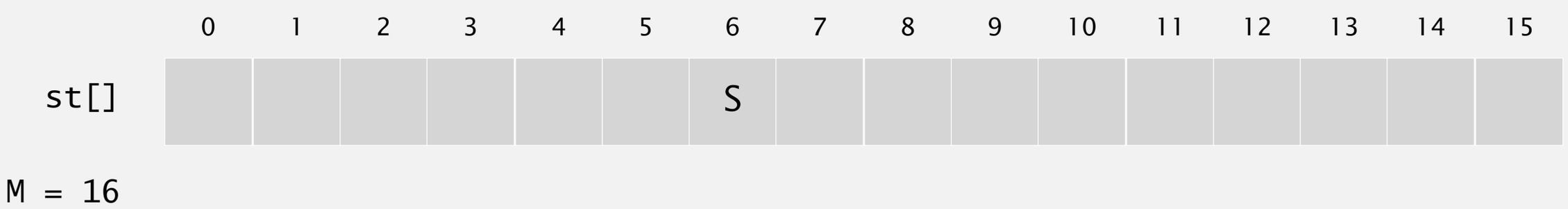
Linear-probing hash table demo: insert

Hash. Map key to integer i between 0 and M-1.

Insert. Put at table index i if free; if not try i+1, i+2, etc.

insert E

hash(E) = 10



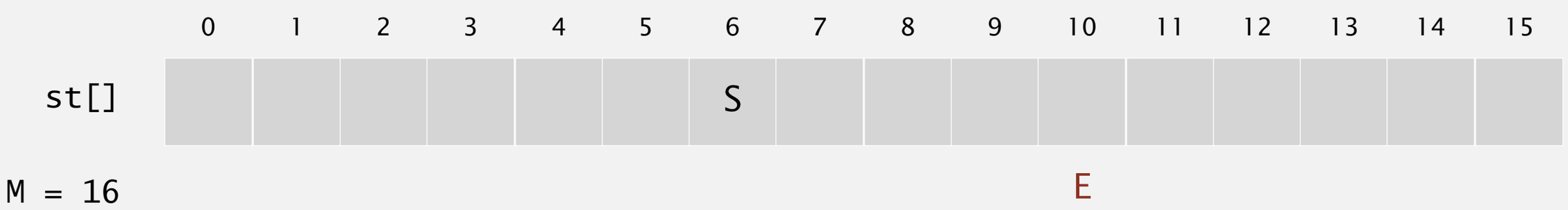
Linear-probing hash table demo: insert

Hash. Map key to integer i between 0 and M-1.

Insert. Put at table index i if free; if not try i+1, i+2, etc.

insert E

hash(E) = 10



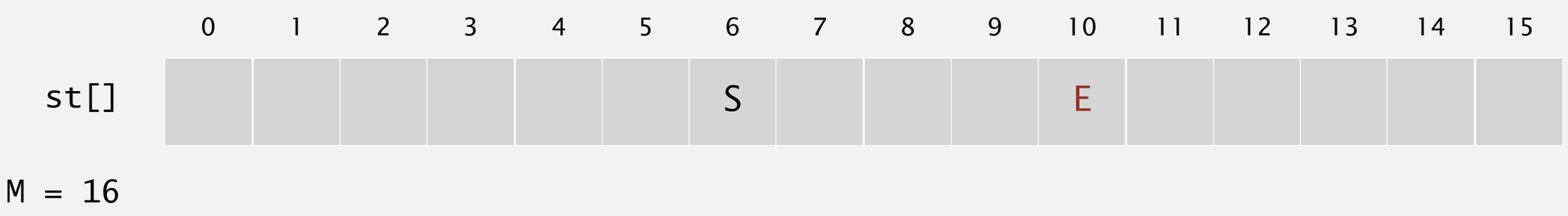
Linear-probing hash table demo: insert

Hash. Map key to integer i between 0 and M-1.

Insert. Put at table index i if free; if not try i+1, i+2, etc.

insert E

hash(E) = 10

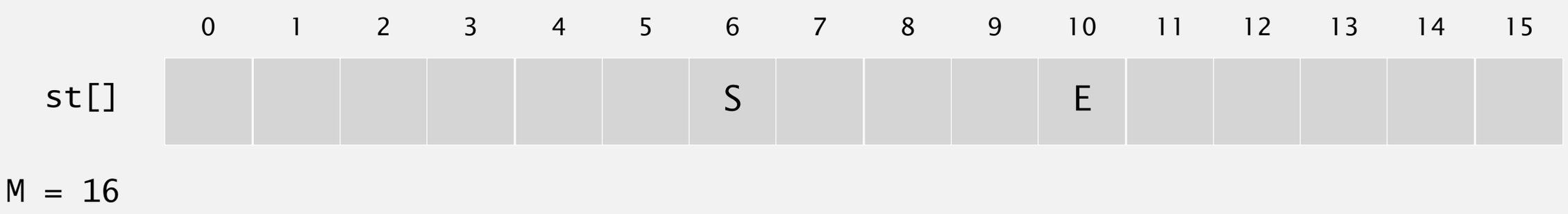


Linear-probing hash table demo: insert

Hash. Map key to integer i between 0 and M-1.

Insert. Put at table index i if free; if not try i+1, i+2, etc.

linear-probing hash table



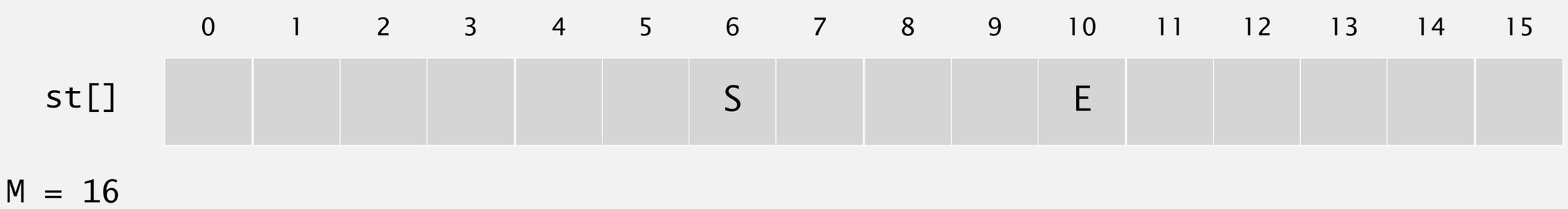
Linear-probing hash table demo: insert

Hash. Map key to integer i between 0 and M-1.

Insert. Put at table index i if free; if not try i+1, i+2, etc.

insert A

hash(A) = 4



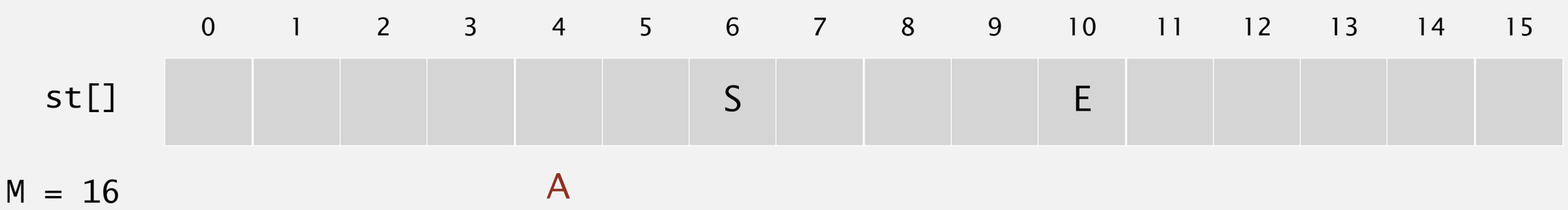
Linear-probing hash table demo: insert

Hash. Map key to integer i between 0 and M-1.

Insert. Put at table index i if free; if not try i+1, i+2, etc.

insert A

hash(A) = 4



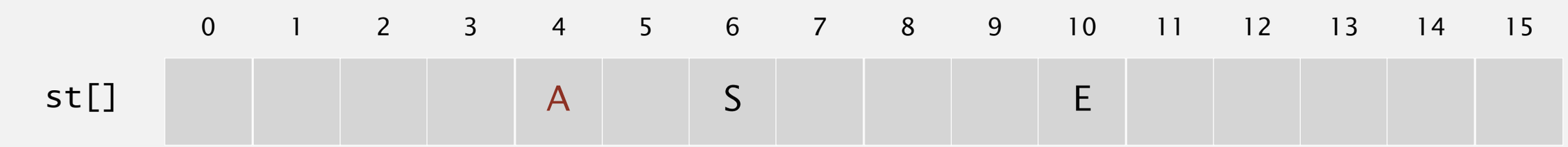
Linear-probing hash table demo: insert

Hash. Map key to integer i between 0 and M-1.

Insert. Put at table index i if free; if not try i+1, i+2, etc.

insert A

hash(A) = 4

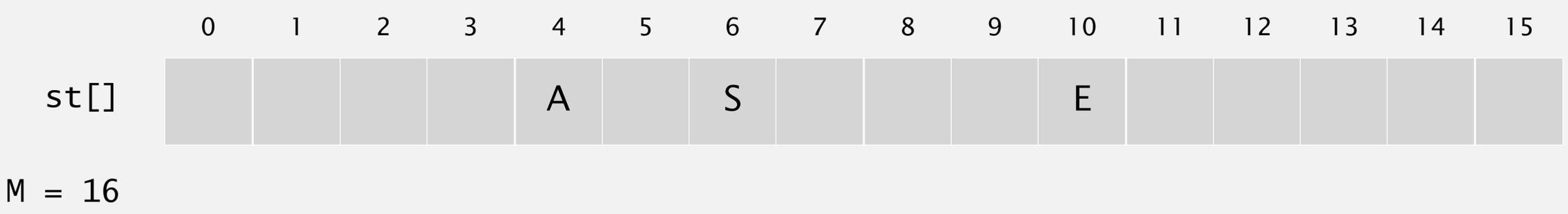


Linear-probing hash table demo: insert

Hash. Map key to integer i between 0 and M-1.

Insert. Put at table index i if free; if not try i+1, i+2, etc.

linear-probing hash table



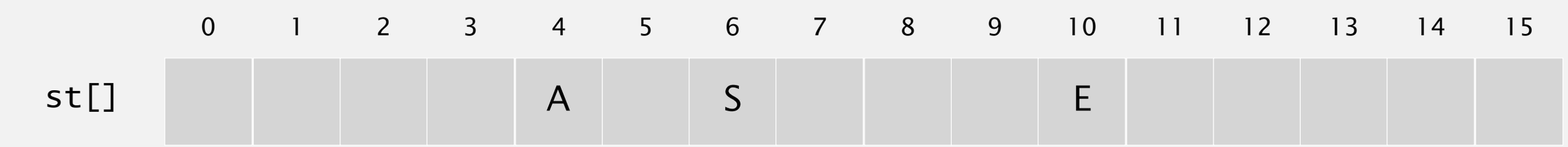
Linear-probing hash table demo: insert

Hash. Map key to integer i between 0 and M-1.

Insert. Put at table index i if free; if not try i+1, i+2, etc.

insert R

hash(R) = 14



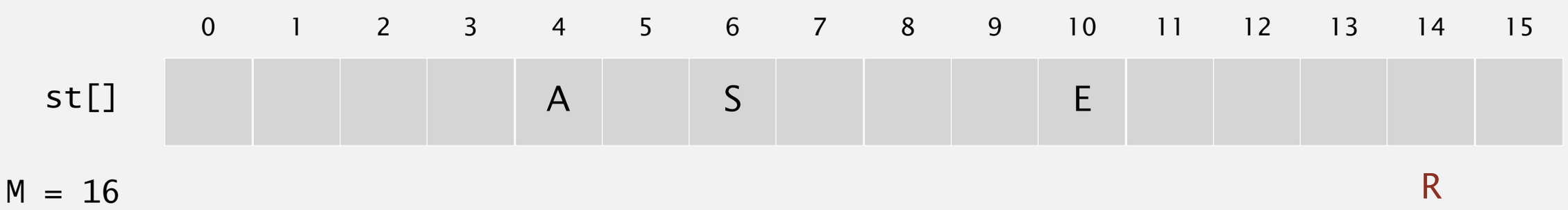
Linear-probing hash table demo: insert

Hash. Map key to integer i between 0 and M-1.

Insert. Put at table index i if free; if not try i+1, i+2, etc.

insert R

hash(R) = 14



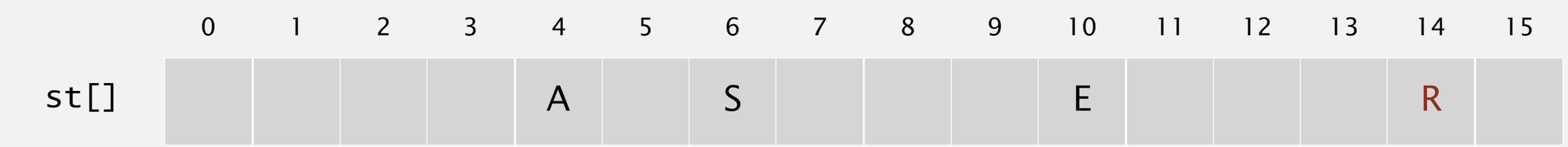
Linear-probing hash table demo: insert

Hash. Map key to integer i between 0 and M-1.

Insert. Put at table index i if free; if not try i+1, i+2, etc.

insert R

hash(R) = 14

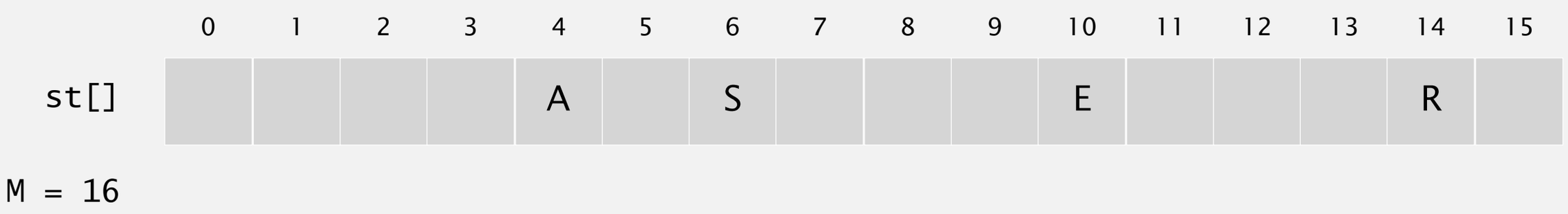


Linear-probing hash table demo: insert

Hash. Map key to integer i between 0 and M-1.

Insert. Put at table index i if free; if not try i+1, i+2, etc.

linear-probing hash table



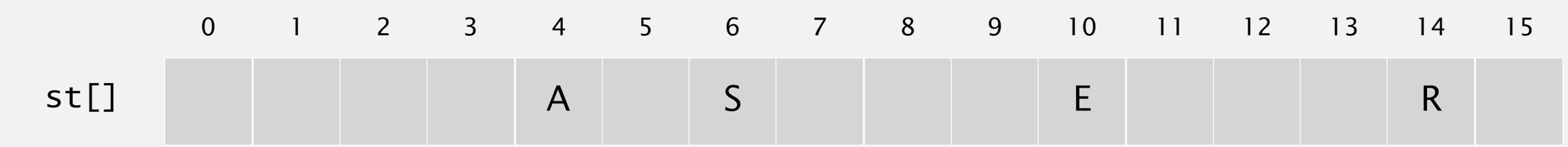
Linear-probing hash table demo: insert

Hash. Map key to integer i between 0 and M-1.

Insert. Put at table index i if free; if not try i+1, i+2, etc.

insert C

hash(C) = 5



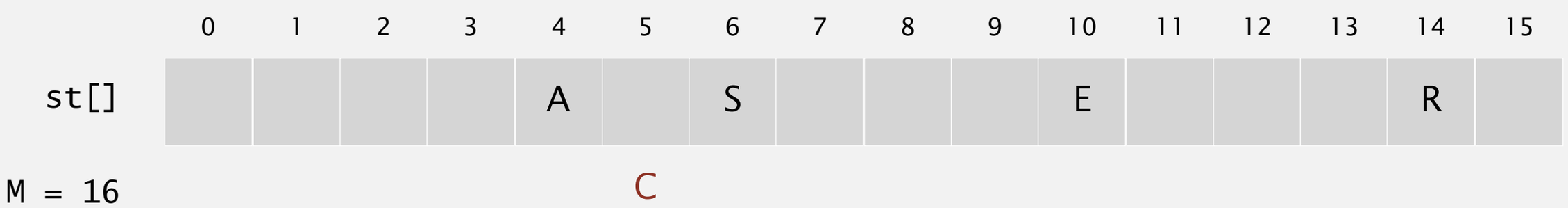
Linear-probing hash table demo: insert

Hash. Map key to integer i between 0 and M-1.

Insert. Put at table index i if free; if not try i+1, i+2, etc.

insert C

hash(C) = 5



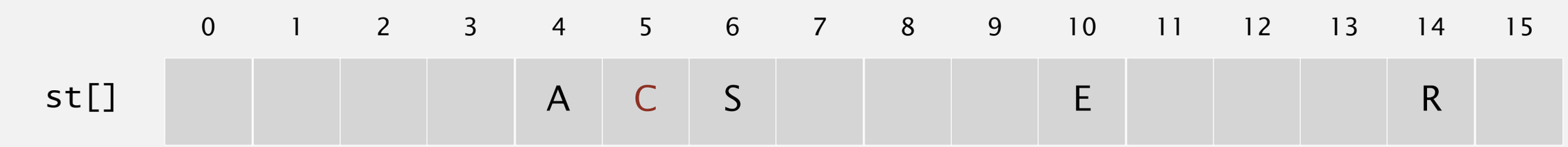
Linear-probing hash table demo: insert

Hash. Map key to integer i between 0 and M-1.

Insert. Put at table index i if free; if not try i+1, i+2, etc.

insert C

hash(C) = 5

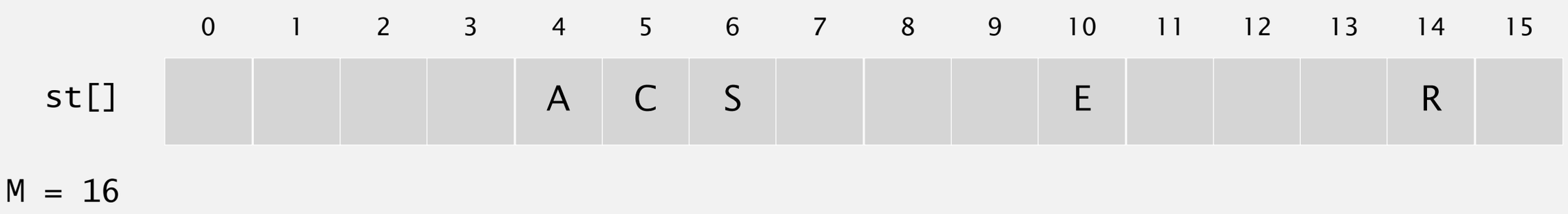


Linear-probing hash table demo: insert

Hash. Map key to integer i between 0 and M-1.

Insert. Put at table index i if free; if not try i+1, i+2, etc.

linear-probing hash table



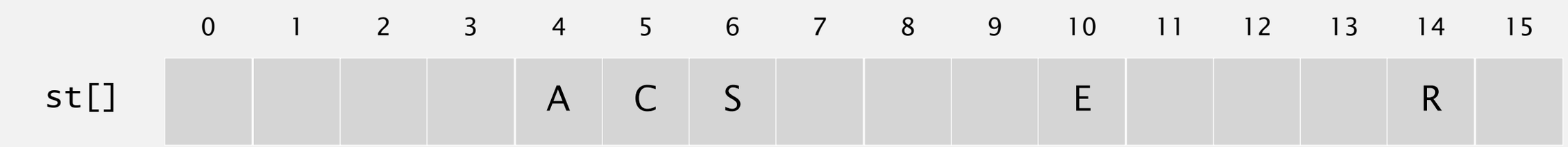
Linear-probing hash table demo: insert

Hash. Map key to integer i between 0 and M-1.

Insert. Put at table index i if free; if not try i+1, i+2, etc.

insert H

hash(H) = 4



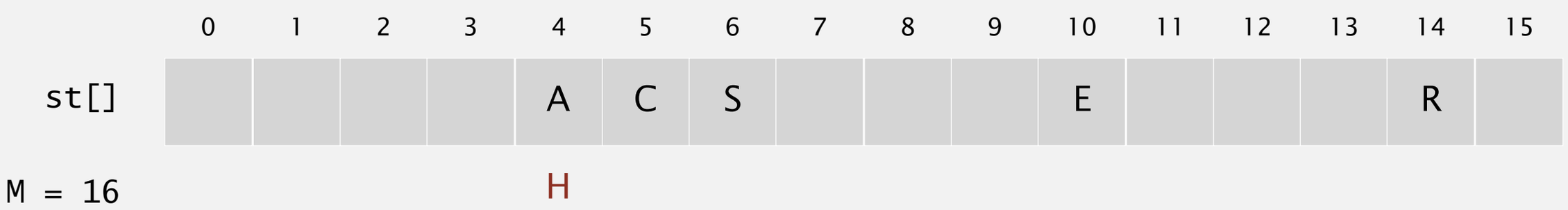
Linear-probing hash table demo: insert

Hash. Map key to integer i between 0 and M-1.

Insert. Put at table index i if free; if not try i+1, i+2, etc.

insert H

hash(H) = 4



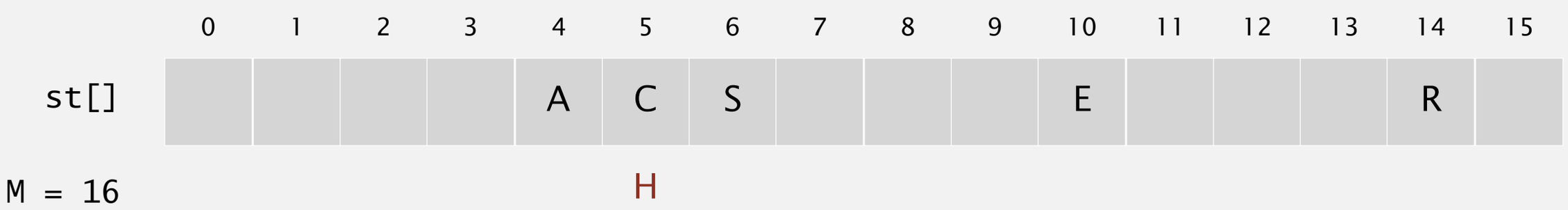
Linear-probing hash table demo: insert

Hash. Map key to integer i between 0 and M-1.

Insert. Put at table index i if free; if not try i+1, i+2, etc.

insert H

hash(H) = 4



Linear-probing hash table demo: insert

Hash. Map key to integer i between 0 and M-1.

Insert. Put at table index i if free; if not try i+1, i+2, etc.

insert H

hash(H) = 4

Linear-probing hash table demo: insert

Hash. Map key to integer i between 0 and M-1.

Insert. Put at table index i if free; if not try i+1, i+2, etc.

insert H

hash(H) = 4

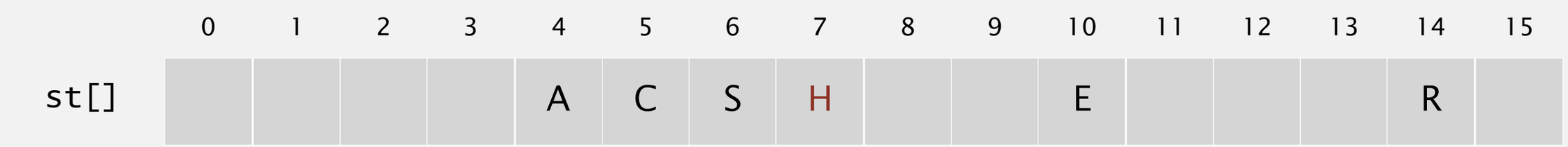
Linear-probing hash table demo: insert

Hash. Map key to integer i between 0 and M-1.

Insert. Put at table index i if free; if not try i+1, i+2, etc.

insert H

hash(H) = 4

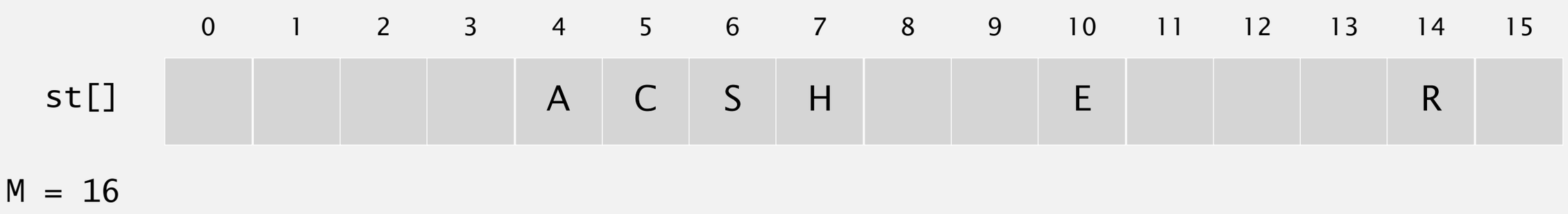


Linear-probing hash table demo: insert

Hash. Map key to integer i between 0 and M-1.

Insert. Put at table index i if free; if not try i+1, i+2, etc.

linear-probing hash table



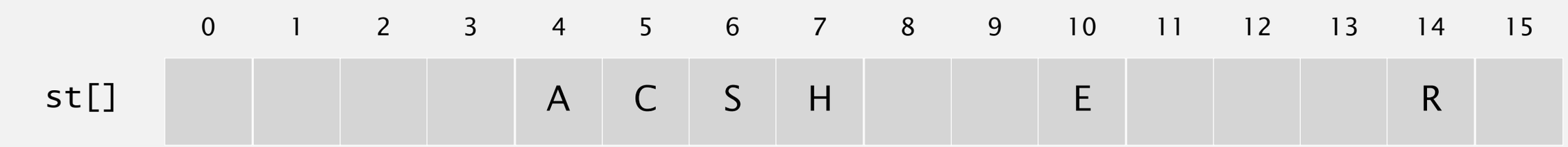
Linear-probing hash table demo: insert

Hash. Map key to integer i between 0 and M-1.

Insert. Put at table index i if free; if not try i+1, i+2, etc.

insert X

hash(X) = 15



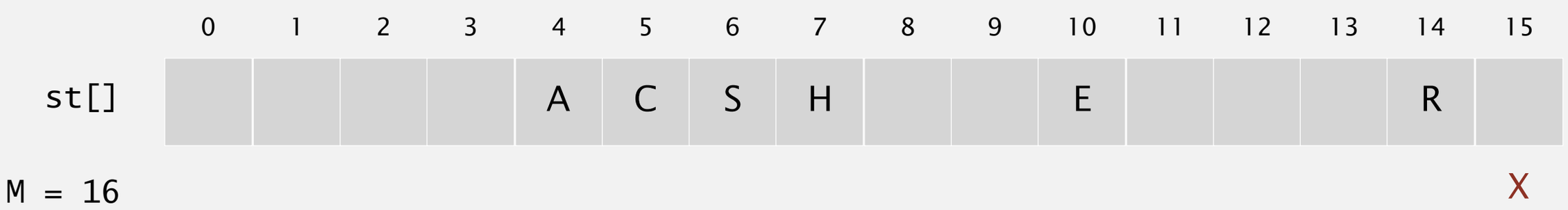
Linear-probing hash table demo: insert

Hash. Map key to integer i between 0 and M-1.

Insert. Put at table index i if free; if not try i+1, i+2, etc.

insert X

hash(X) = 15



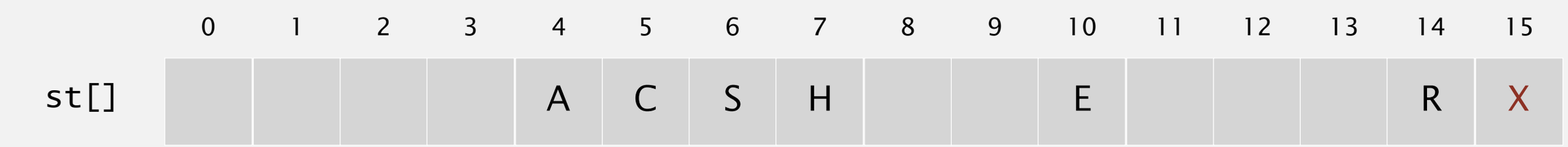
Linear-probing hash table demo: insert

Hash. Map key to integer i between 0 and M-1.

Insert. Put at table index i if free; if not try i+1, i+2, etc.

insert X

hash(X) = 15

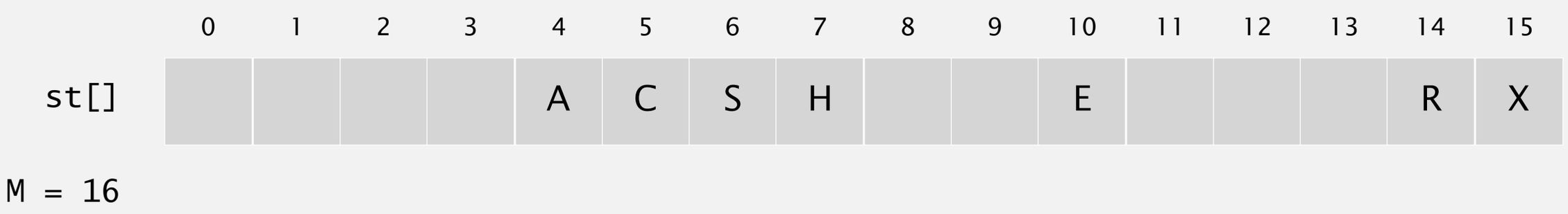


Linear-probing hash table demo: insert

Hash. Map key to integer i between 0 and M-1.

Insert. Put at table index i if free; if not try i+1, i+2, etc.

linear-probing hash table



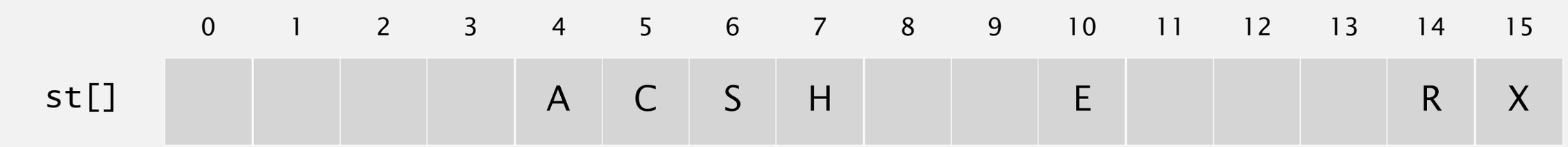
Linear-probing hash table demo: insert

Hash. Map key to integer i between 0 and M-1.

Insert. Put at table index i if free; if not try i+1, i+2, etc.

insert M

hash(M) = 1



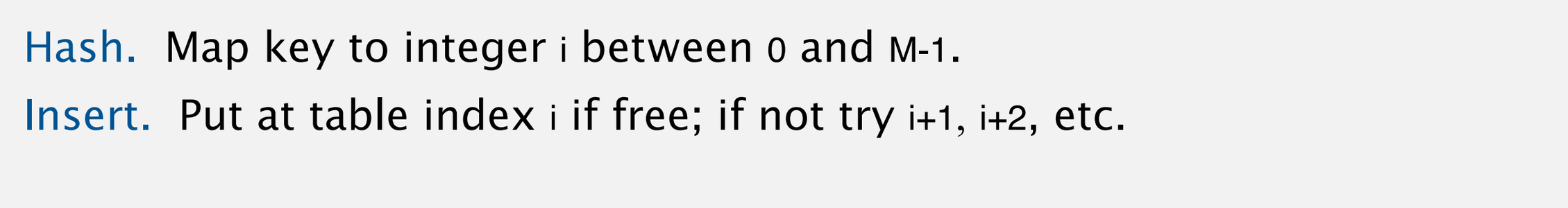
Linear-probing hash table demo: insert

Hash. Map key to integer i between 0 and M-1.

Insert. Put at table index i if free; if not try i+1, i+2, etc.

insert M

hash(M) = 1



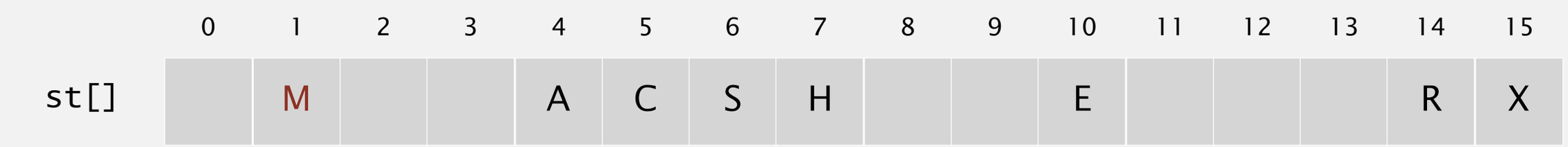
Linear-probing hash table demo: insert

Hash. Map key to integer i between 0 and M-1.

Insert. Put at table index i if free; if not try i+1, i+2, etc.

insert M

hash(M) = 1

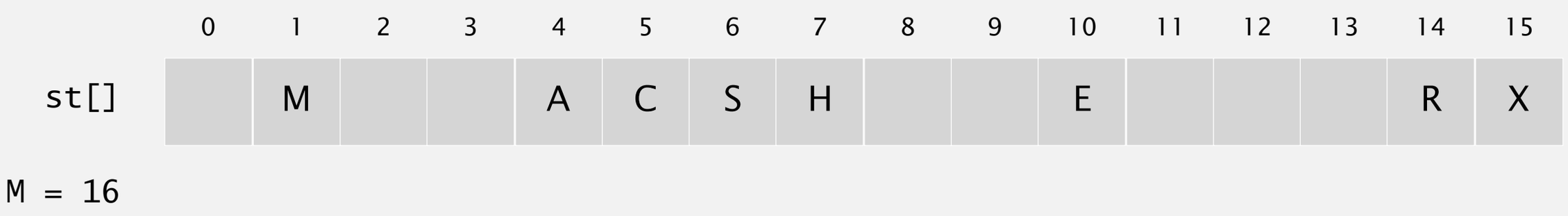


Linear-probing hash table demo: insert

Hash. Map key to integer i between 0 and M-1.

Insert. Put at table index i if free; if not try i+1, i+2, etc.

linear-probing hash table



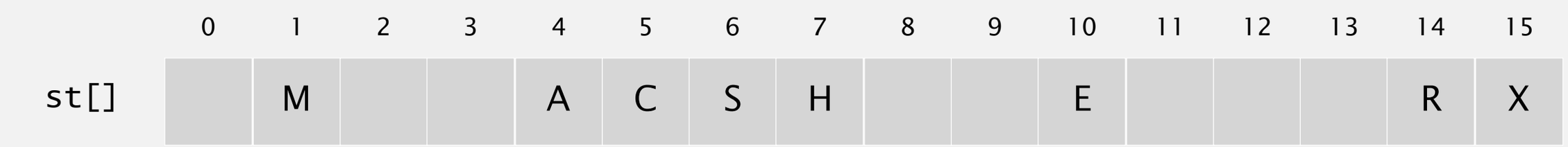
Linear-probing hash table demo: insert

Hash. Map key to integer i between 0 and M-1.

Insert. Put at table index i if free; if not try i+1, i+2, etc.

insert P

hash(P) = 14



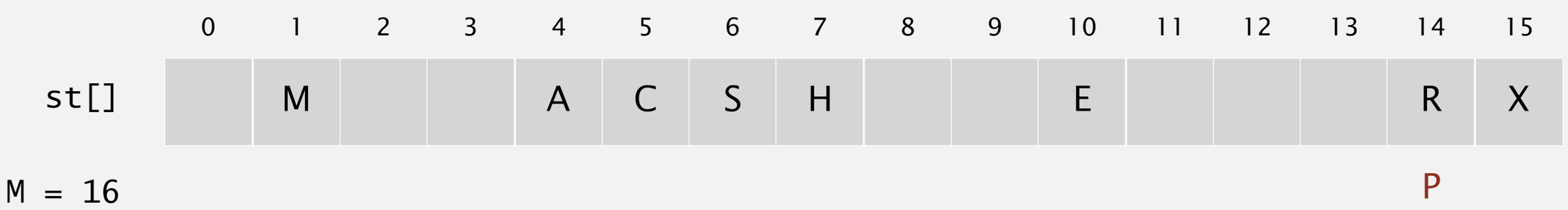
Linear-probing hash table demo: insert

Hash. Map key to integer i between 0 and M-1.

Insert. Put at table index i if free; if not try i+1, i+2, etc.

insert P

hash(P) = 14



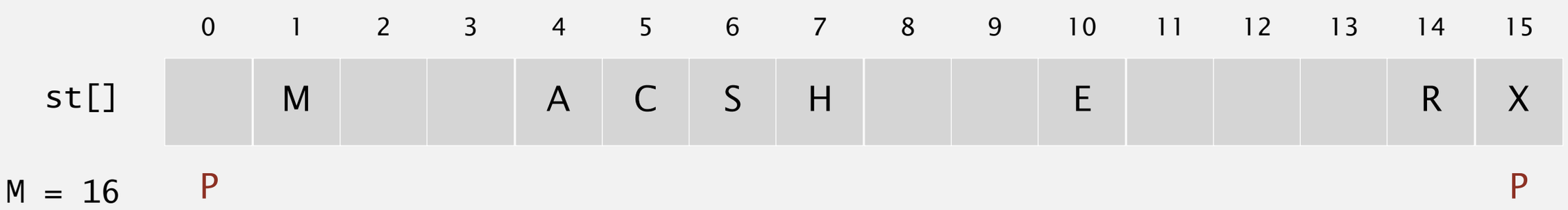
Linear-probing hash table demo: insert

Hash. Map key to integer i between 0 and M-1.

Insert. Put at table index i if free; if not try i+1, i+2, etc.

insert P

hash(P) = 14



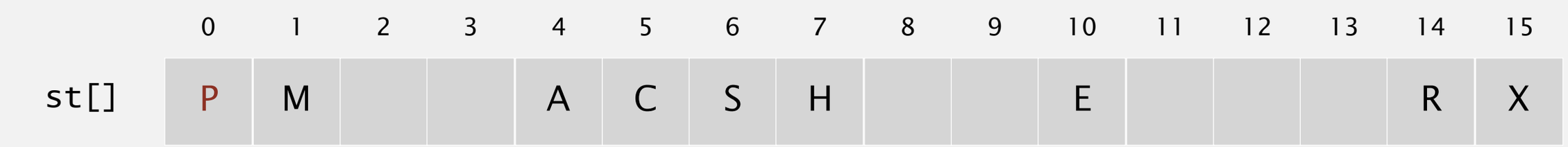
Linear-probing hash table demo: insert

Hash. Map key to integer i between 0 and M-1.

Insert. Put at table index i if free; if not try i+1, i+2, etc.

insert P

hash(P) = 14

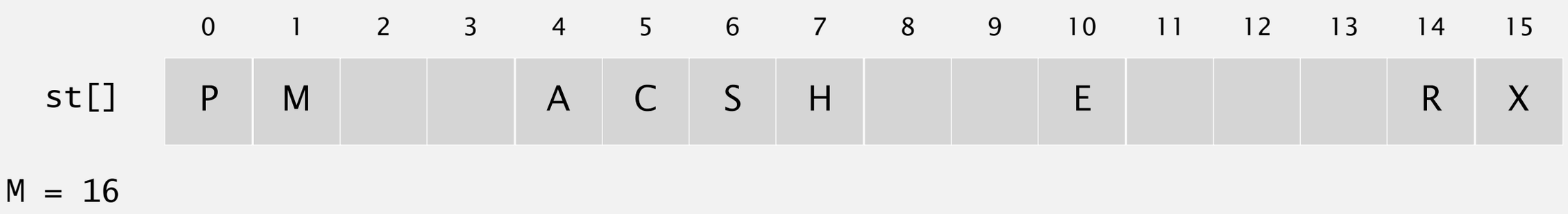


Linear-probing hash table demo: insert

Hash. Map key to integer i between 0 and M-1.

Insert. Put at table index i if free; if not try i+1, i+2, etc.

linear-probing hash table



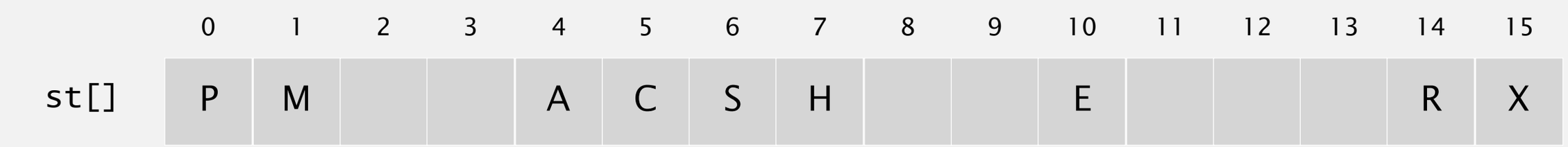
Linear-probing hash table demo: insert

Hash. Map key to integer i between 0 and M-1.

Insert. Put at table index i if free; if not try i+1, i+2, etc.

insert L

hash(L) = 6



Linear-probing hash table demo: insert

Hash. Map key to integer i between 0 and M-1.

Insert. Put at table index i if free; if not try i+1, i+2, etc.

insert L

hash(L) = 6

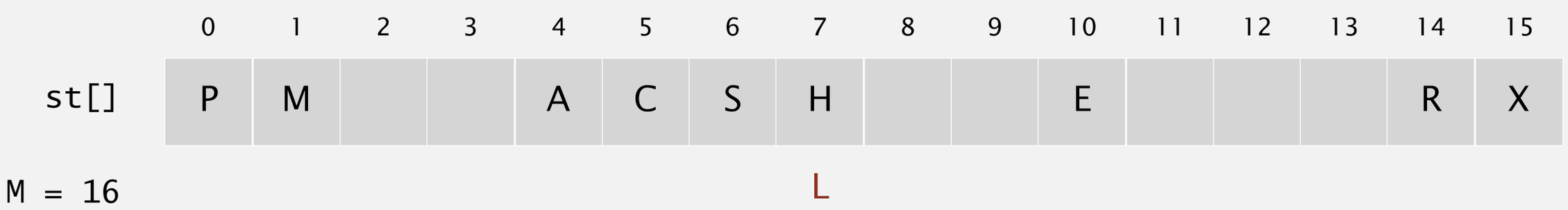
Linear-probing hash table demo: insert

Hash. Map key to integer i between 0 and M-1.

Insert. Put at table index i if free; if not try i+1, i+2, etc.

insert L

hash(L) = 6



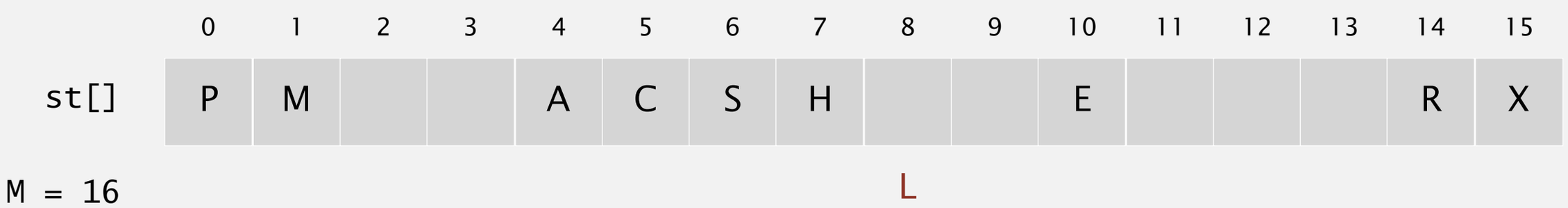
Linear-probing hash table demo: insert

Hash. Map key to integer i between 0 and M-1.

Insert. Put at table index i if free; if not try i+1, i+2, etc.

insert L

hash(L) = 6



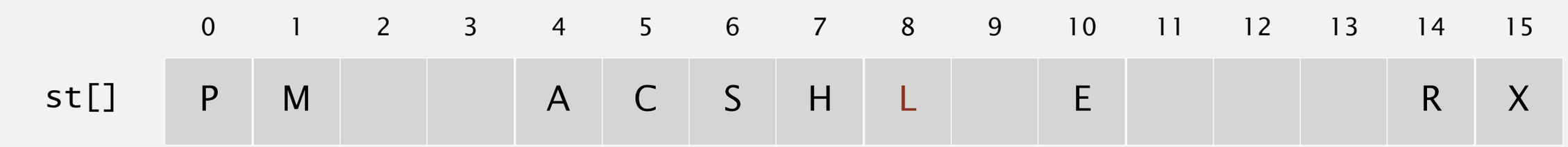
Linear-probing hash table demo: insert

Hash. Map key to integer i between 0 and M-1.

Insert. Put at table index i if free; if not try i+1, i+2, etc.

insert L

hash(L) = 6

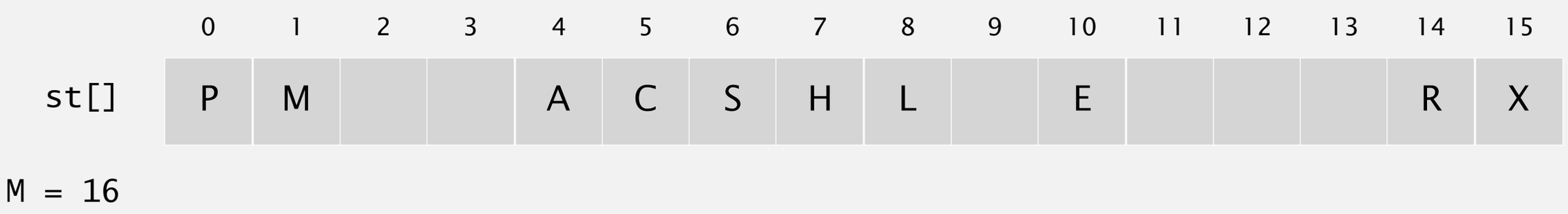


Linear-probing hash table demo: insert

Hash. Map key to integer i between 0 and M-1.

Insert. Put at table index i if free; if not try i+1, i+2, etc.

linear-probing hash table



http://algs4.cs.princeton.edu

3.4 Linear Probing Demo

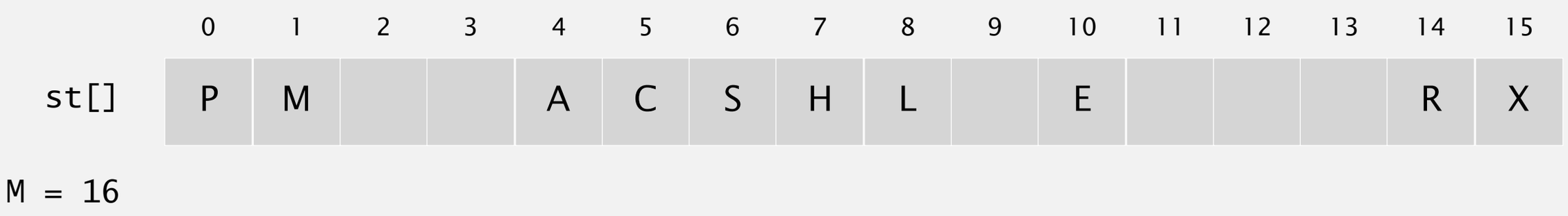
- *insert*
- *search*

Linear-probing hash table demo: search

Hash. Map key to integer i between 0 and M-1.

Search. Search table index i; if occupied but no match, try i+1, i+2, etc.

linear-probing hash table



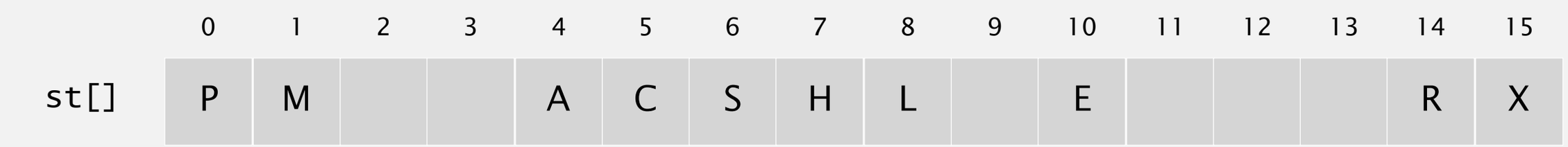
Linear-probing hash table demo: search

Hash. Map key to integer i between 0 and M-1.

Search. Search table index i; if occupied but no match, try i+1, i+2, etc.

search E

hash(E) = 10



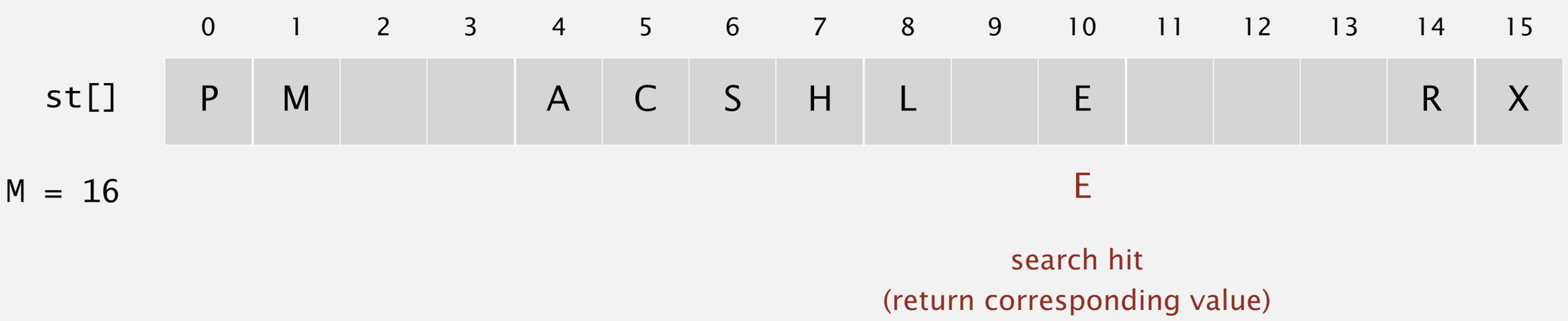
Linear-probing hash table demo: search

Hash. Map key to integer i between 0 and M-1.

Search. Search table index i; if occupied but no match, try i+1, i+2, etc.

search E

hash(E) = 10

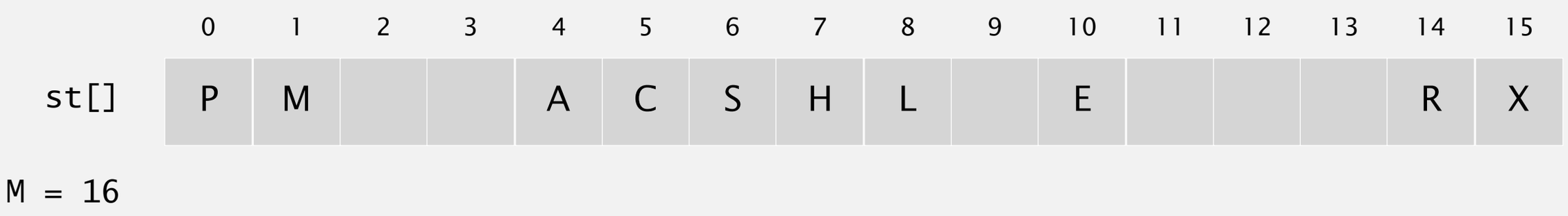


Linear-probing hash table demo: search

Hash. Map key to integer i between 0 and M-1.

Search. Search table index i; if occupied but no match, try i+1, i+2, etc.

linear-probing hash table



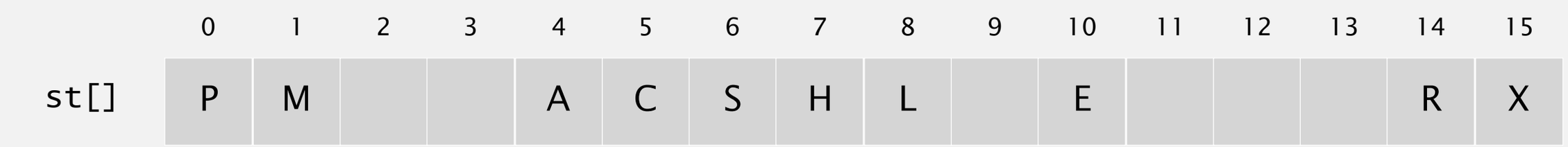
Linear-probing hash table demo: search

Hash. Map key to integer i between 0 and M-1.

Search. Search table index i; if occupied but no match, try i+1, i+2, etc.

search L

hash(L) = 6



Linear-probing hash table demo: search

Hash. Map key to integer i between 0 and M-1.

Search. Search table index i; if occupied but no match, try i+1, i+2, etc.

search L

hash(L) = 6

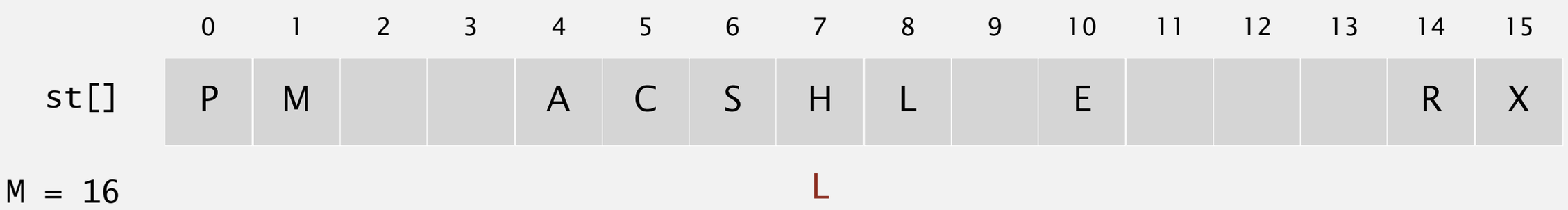
Linear-probing hash table demo: search

Hash. Map key to integer i between 0 and M-1.

Search. Search table index i; if occupied but no match, try i+1, i+2, etc.

search L

hash(L) = 6



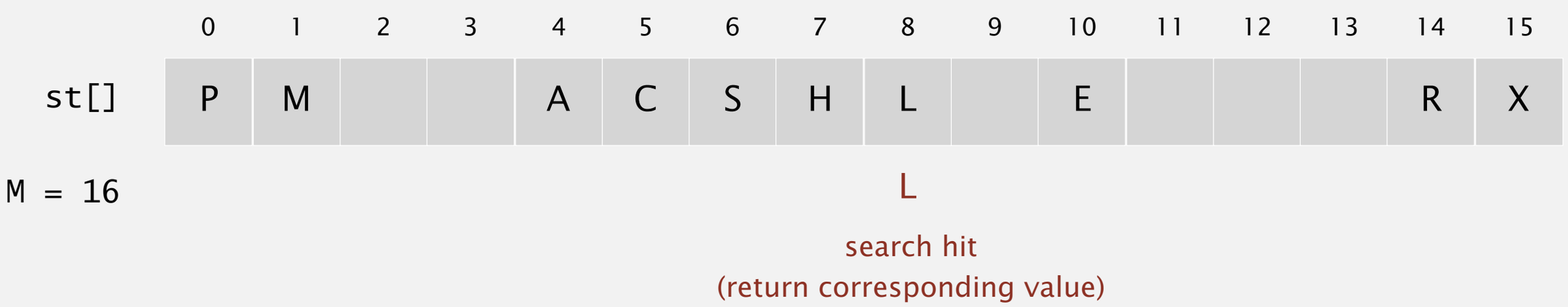
Linear-probing hash table demo: search

Hash. Map key to integer i between 0 and M-1.

Search. Search table index i; if occupied but no match, try i+1, i+2, etc.

search L

hash(L) = 6

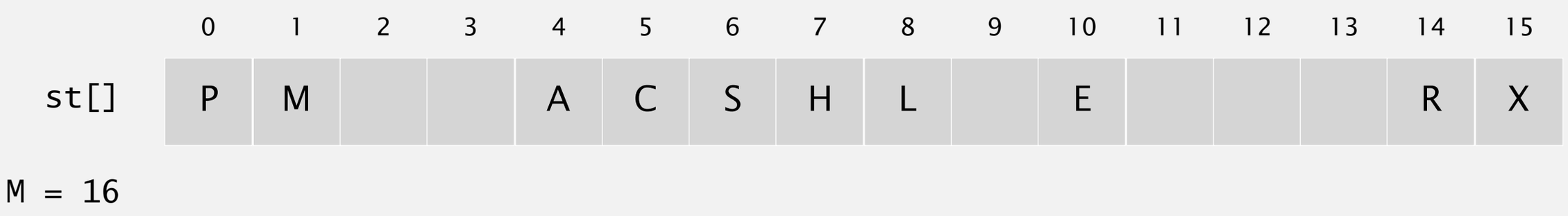


Linear-probing hash table demo: search

Hash. Map key to integer i between 0 and M-1.

Search. Search table index i; if occupied but no match, try i+1, i+2, etc.

linear-probing hash table



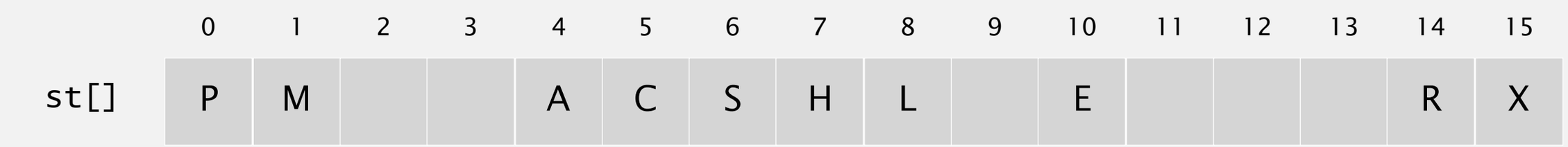
Linear-probing hash table demo: search

Hash. Map key to integer i between 0 and M-1.

Search. Search table index i; if occupied but no match, try i+1, i+2, etc.

search K

hash(K) = 5



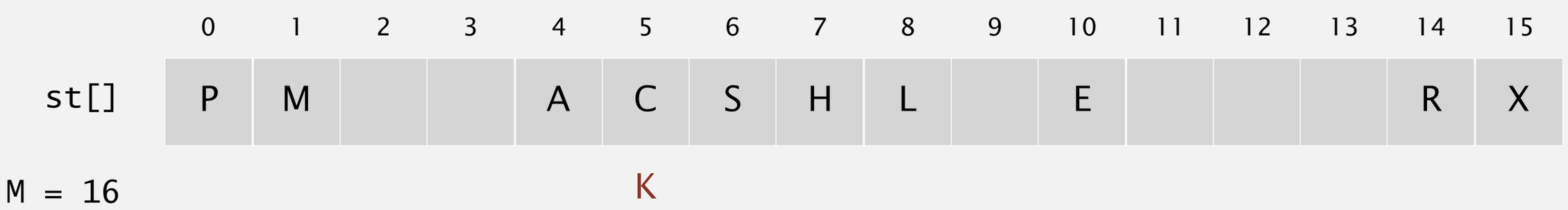
Linear-probing hash table demo: search

Hash. Map key to integer i between 0 and M-1.

Search. Search table index i; if occupied but no match, try i+1, i+2, etc.

search K

hash(K) = 5



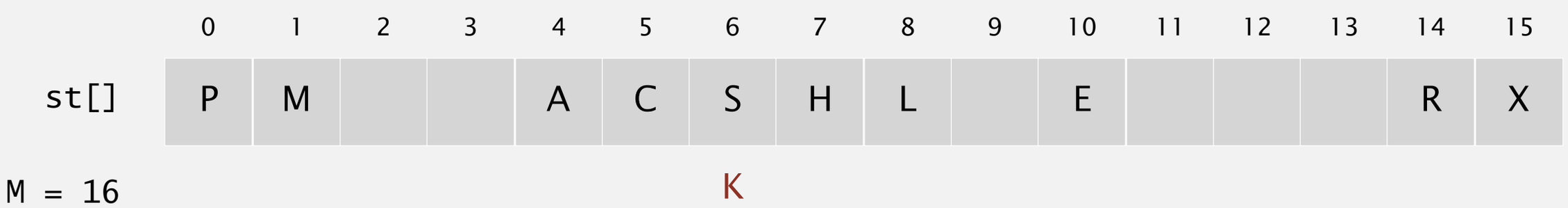
Linear-probing hash table demo: search

Hash. Map key to integer i between 0 and M-1.

Search. Search table index i; if occupied but no match, try i+1, i+2, etc.

search K

hash(K) = 5



Linear-probing hash table demo: search

Hash. Map key to integer i between 0 and M-1.

Search. Search table index i; if occupied but no match, try i+1, i+2, etc.

search K

hash(K) = 5

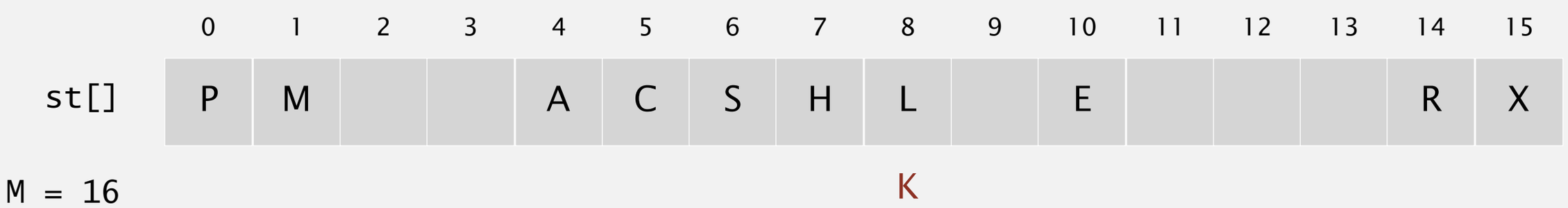
Linear-probing hash table demo: search

Hash. Map key to integer i between 0 and M-1.

Search. Search table index i; if occupied but no match, try i+1, i+2, etc.

search K

hash(K) = 5



Linear-probing hash table demo: search

Hash. Map key to integer i between 0 and M-1.

Search. Search table index i; if occupied but no match, try i+1, i+2, etc.

search K

hash(K) = 5