

3.4 LINEAR PROBING DEMO



<http://algs4.cs.princeton.edu>



<http://algs4.cs.princeton.edu>

3.4 LINEAR PROBING DEMO

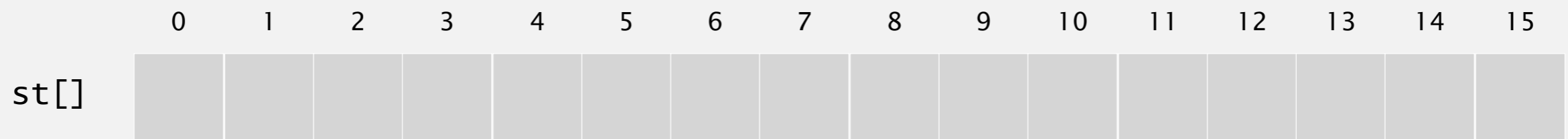
- *insert*
- *search*

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



$M = 16$

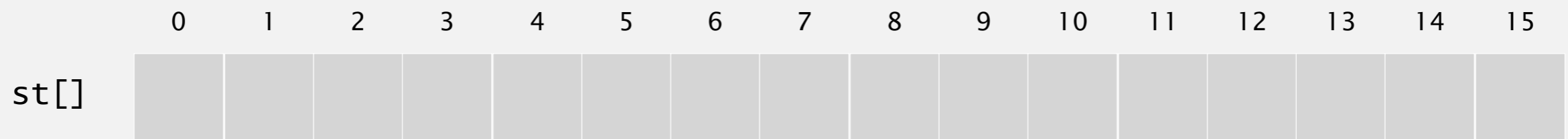
Linear-probing hash table demo: insert

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

$\text{hash}(S) = 6$



$M = 16$

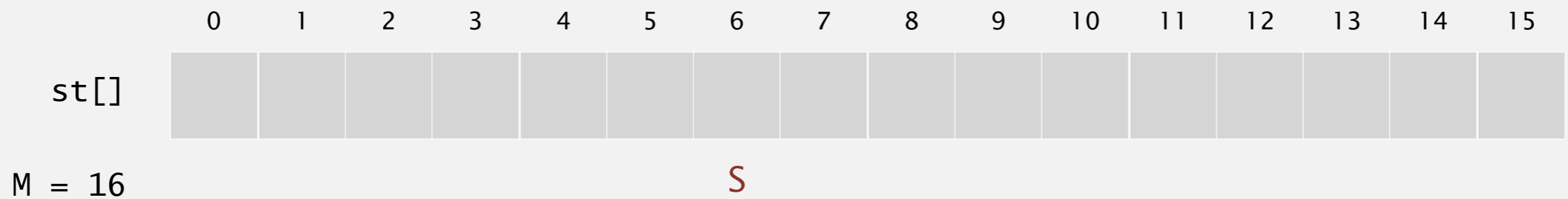
Linear-probing hash table demo: insert

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

$\text{hash}(S) = 6$



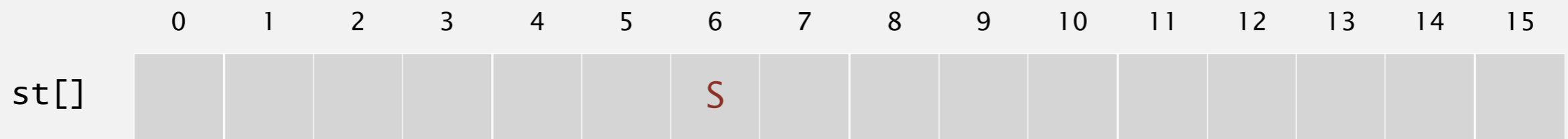
Linear-probing hash table demo: insert

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

$\text{hash}(S) = 6$



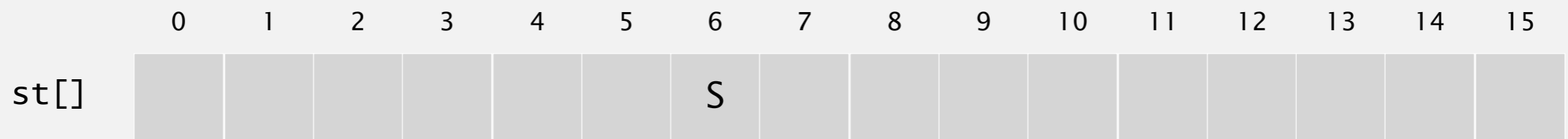
$M = 16$

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



$M = 16$

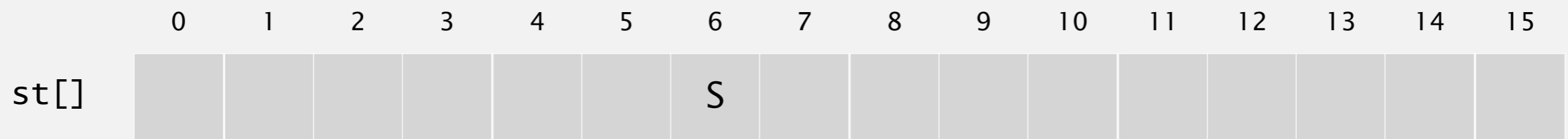
Linear-probing hash table demo: insert

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



M = 16

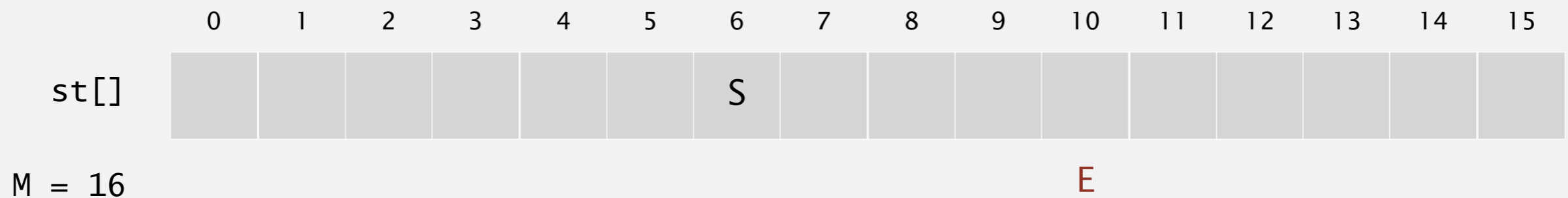
Linear-probing hash table demo: insert

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

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
st[]							S				E					

$M = 16$

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

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
st[]							S				E					

$M = 16$

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

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
st[]							S				E					

M = 16

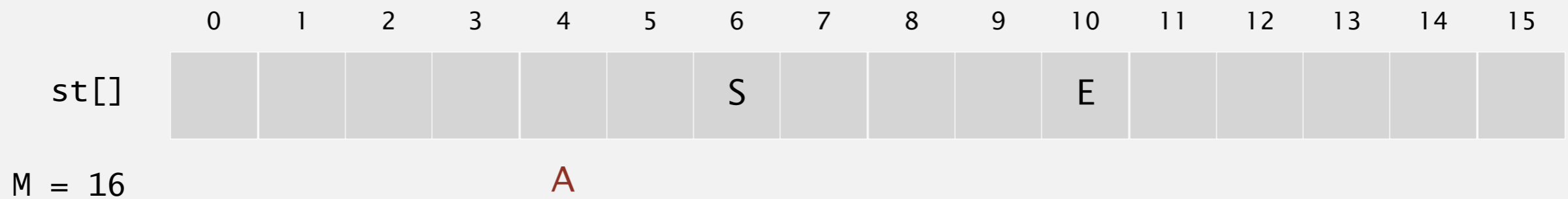
Linear-probing hash table demo: insert

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

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
st[]					A		S				E					

M = 16

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

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
st[]					A		S				E					

$M = 16$

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

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
st[]					A		S				E					

$M = 16$

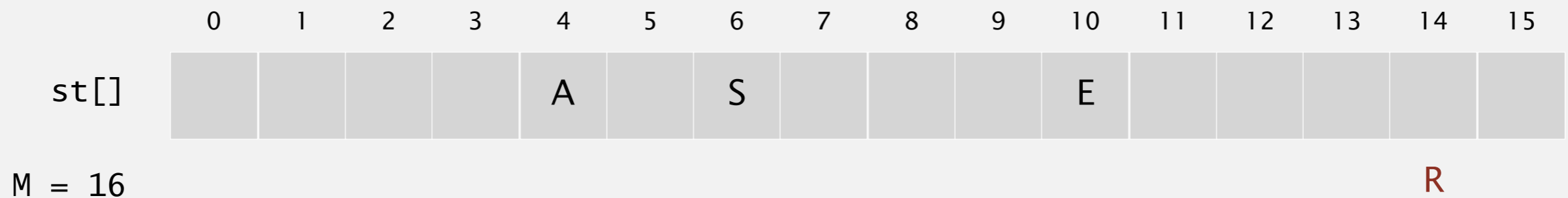
Linear-probing hash table demo: insert

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

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
st[]					A		S				E				R	

$M = 16$

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

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
st[]					A		S				E				R	

$M = 16$

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

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
st[]					A		S				E				R	

$M = 16$

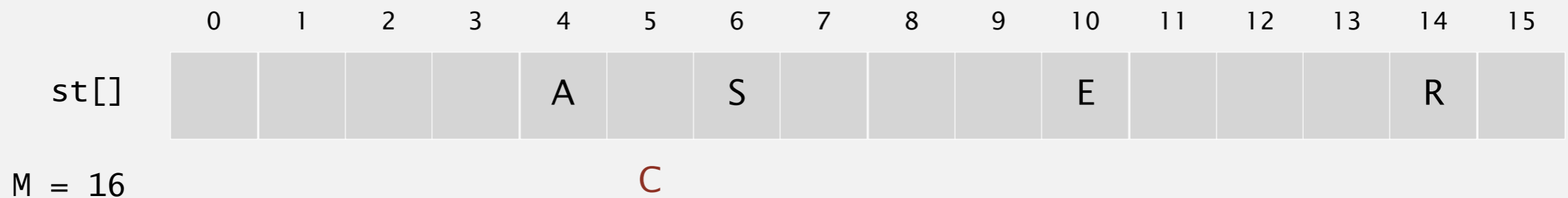
Linear-probing hash table demo: insert

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

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
st[]					A	C	S				E				R	

M = 16

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

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
st[]					A	C	S				E				R	

$M = 16$

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

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
st[]					A	C	S				E				R	

$M = 16$

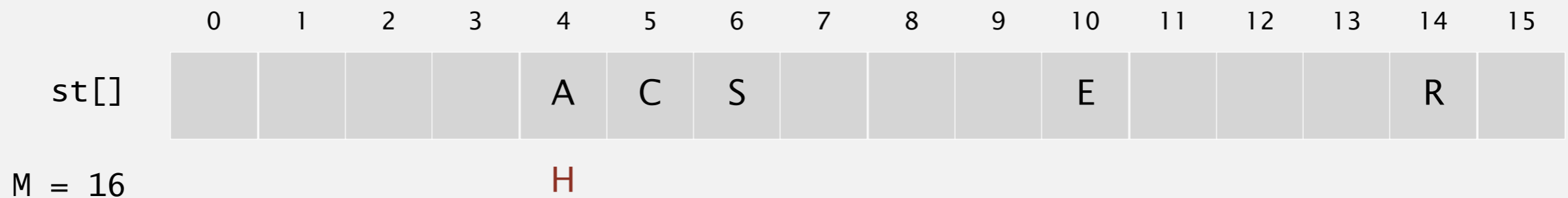
Linear-probing hash table demo: insert

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



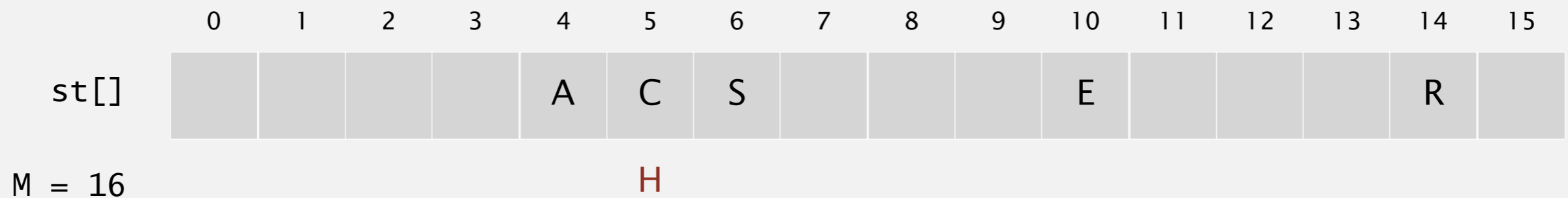
Linear-probing hash table demo: insert

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



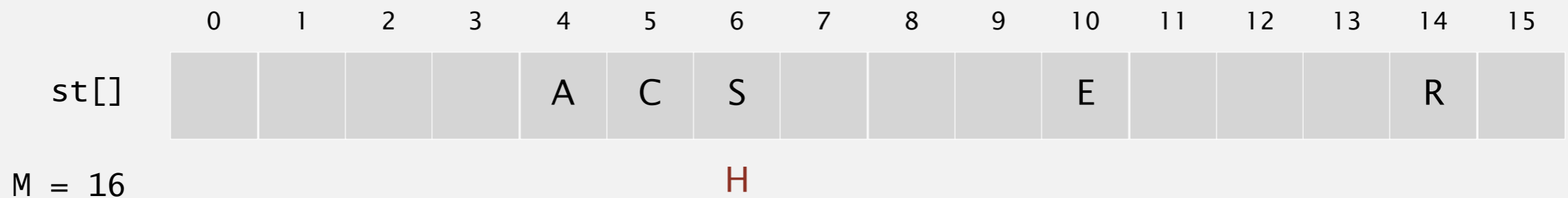
Linear-probing hash table demo: insert

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



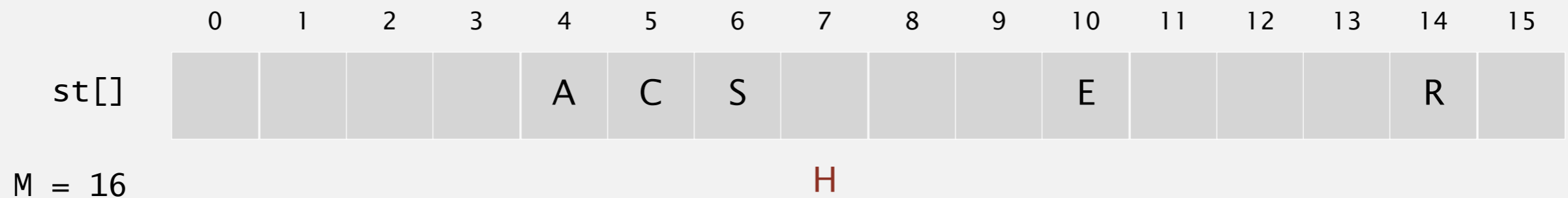
Linear-probing hash table demo: insert

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

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
st[]					A	C	S	H			E				R	

$M = 16$

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

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
st[]					A	C	S	H			E				R	

$M = 16$

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

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
st[]					A	C	S	H			E				R	

M = 16

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

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
st[]					A	C	S	H			E				R	

M = 16

X

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

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
st[]					A	C	S	H			E				R	X

M = 16

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

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
st[]					A	C	S	H			E				R	X

$M = 16$

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

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
st[]					A	C	S	H			E				R	X

M = 16

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

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
st[]		M			A	C	S	H			E				R	X

M = 16

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

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
st[]		M			A	C	S	H			E				R	X

$M = 16$

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

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
st[]		M			A	C	S	H			E				R	X

$M = 16$

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

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
st[]		M			A	C	S	H			E				R	X

M = 16

P

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

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
st[]	P	M			A	C	S	H			E				R	X

M = 16

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

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
st[]	P	M			A	C	S	H			E				R	X

$M = 16$

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

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
st[]	P	M			A	C	S	H			E				R	X

$M = 16$

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

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
st[]	P	M			A	C	S	H			E				R	X

M = 16

L

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

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
st[]	P	M			A	C	S	H			E				R	X

M = 16

L

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

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
st[]	P	M			A	C	S	H			E				R	X

M = 16

L

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

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
st[]	P	M			A	C	S	H	L		E				R	X

$M = 16$

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

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
st[]	P	M			A	C	S	H	L		E				R	X

$M = 16$



<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

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
st[]	P	M			A	C	S	H	L		E				R	X

$M = 16$

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

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
st[]	P	M			A	C	S	H	L		E				R	X

M = 16

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

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
st[]	P	M			A	C	S	H	L		E				R	X

$M = 16$

E

search hit
(return corresponding value)

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

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
st[]	P	M			A	C	S	H	L		E				R	X

$M = 16$

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

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
st[]	P	M			A	C	S	H	L		E				R	X

$M = 16$

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

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
st[]	P	M			A	C	S	H	L		E				R	X

M = 16

L

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

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
st[]	P	M			A	C	S	H	L		E				R	X

M = 16

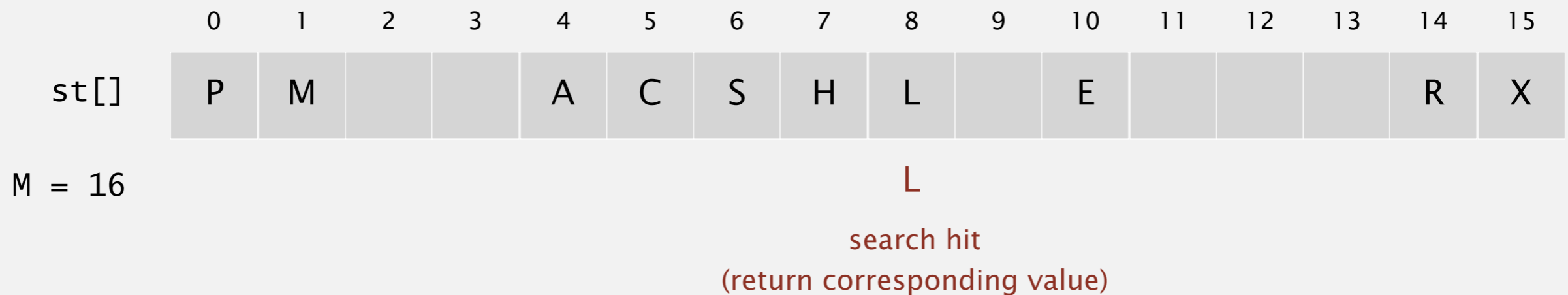
L

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

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
st[]	P	M			A	C	S	H	L		E				R	X

$M = 16$

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

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
st[]	P	M			A	C	S	H	L		E				R	X

$M = 16$

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

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
st[]	P	M			A	C	S	H	L		E				R	X

M = 16

K

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

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
st[]	P	M			A	C	S	H	L		E				R	X

M = 16

K

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

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
st[]	P	M			A	C	S	H	L		E				R	X

M = 16

K

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

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
st[]	P	M			A	C	S	H	L		E				R	X

M = 16

K

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
 $\text{hash}(K) = 5$

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
st[]	P	M			A	C	S	H	L		E				R	X

$M = 16$

K

search miss
(return null)