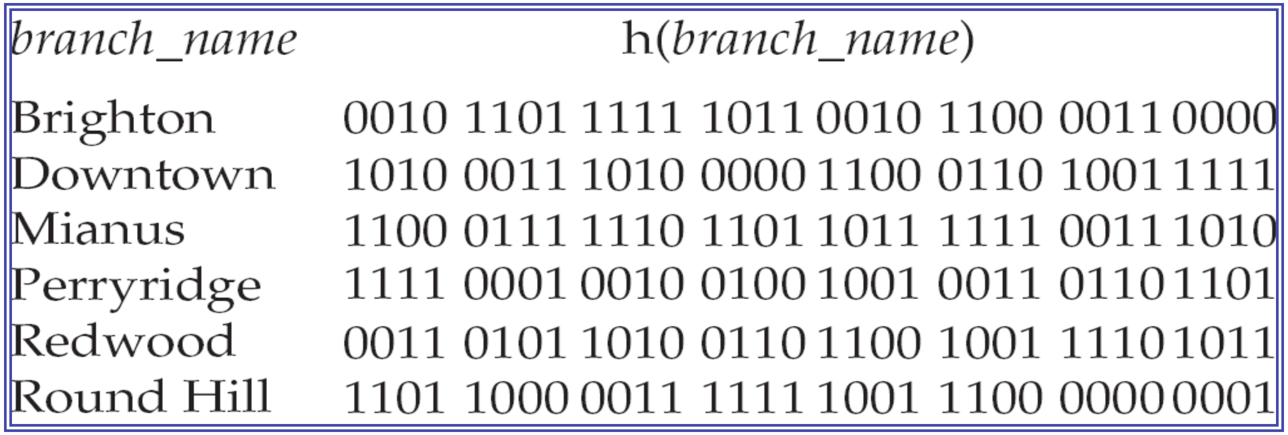
Database Development and Design (CSE210)

Tutorial 3

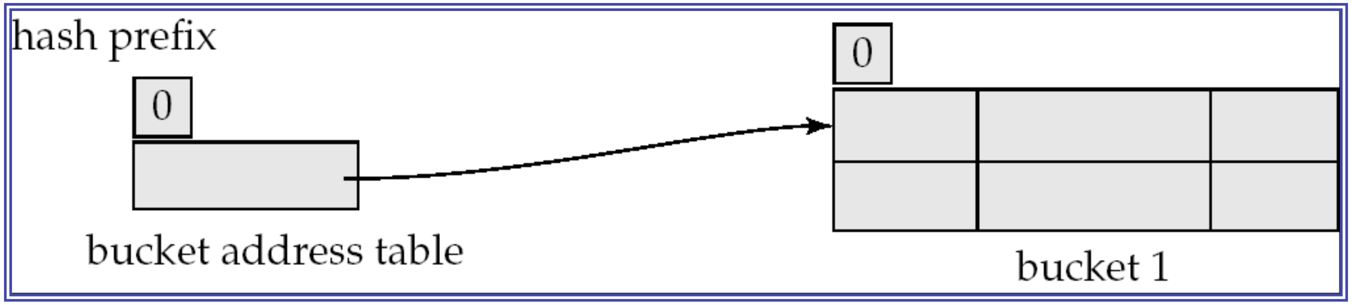
Dr. Wei Wang

Department of Computing

Q1



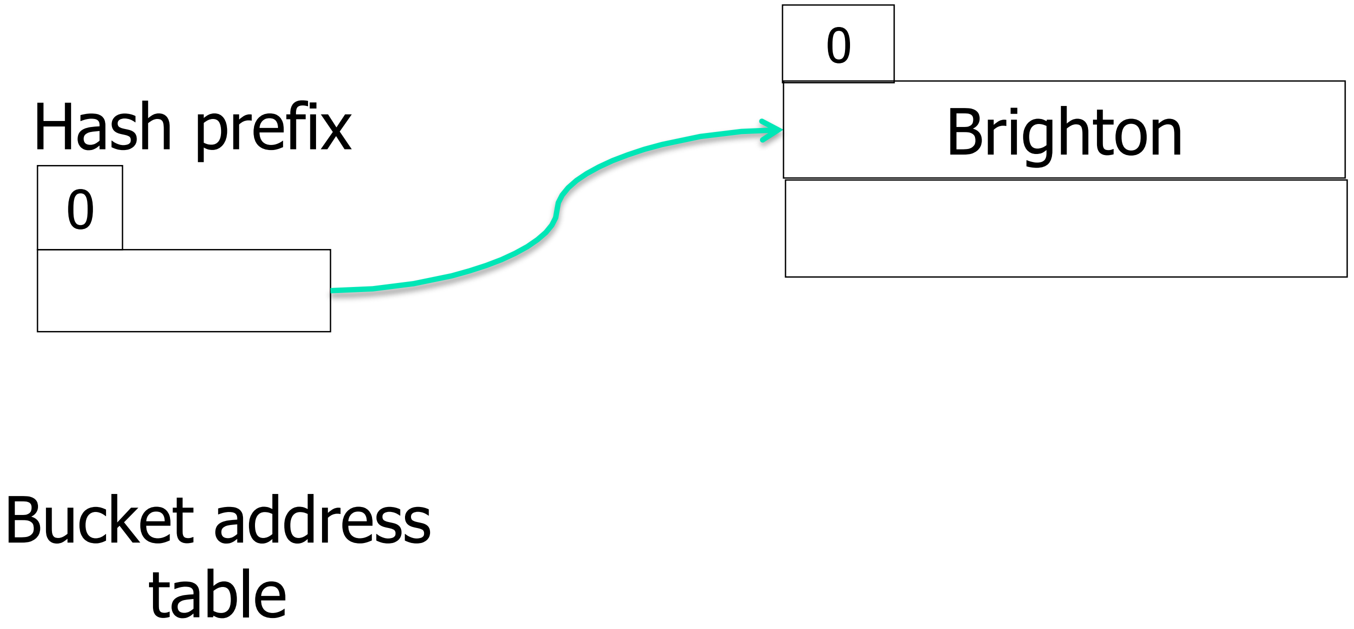
branch_name	h(*branch_name*)
Brighton	0010 1101 1111 1011 0010 1100 0011 0000
Downtown	1010 0011 1010 0000 1100 0110 1001 1111
Mianus	1100 0111 1110 1101 1011 1111 0011 1010
Perryridge	1111 0001 0010 0100 1001 0011 0110 1101
Redwood	0011 0101 1010 0110 1100 1001 1110 1011
Round Hill	1101 1000 0011 1111 1001 1100 0000 0001



- Given the above hash values for the branch_name and the initial hash index, draw the hash indices after inserting the following 8 values. Assume that one block can hold two tuples.
    - Brighton, Downtown, Downtown, Mianus, Perryridge, Perryridge, Perryridge, Redwood

Q1 Solution

- Insert "Brighton"



Hash prefix

0

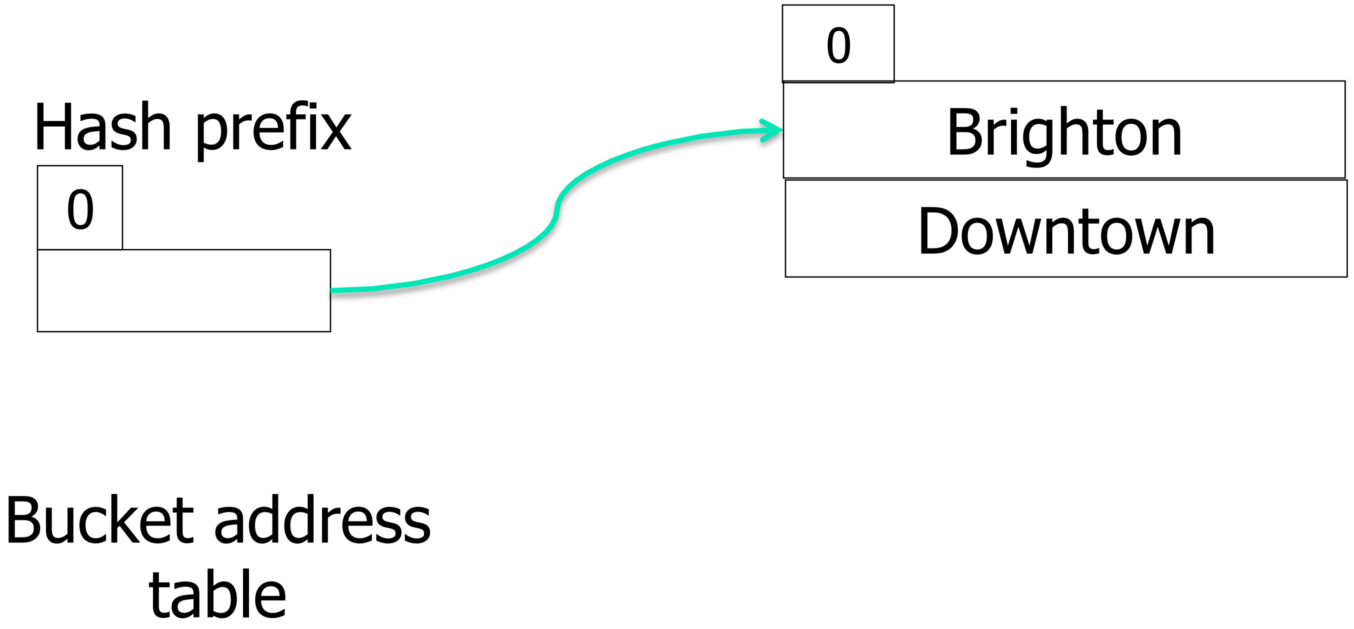
Bucket address table

0

Brighton

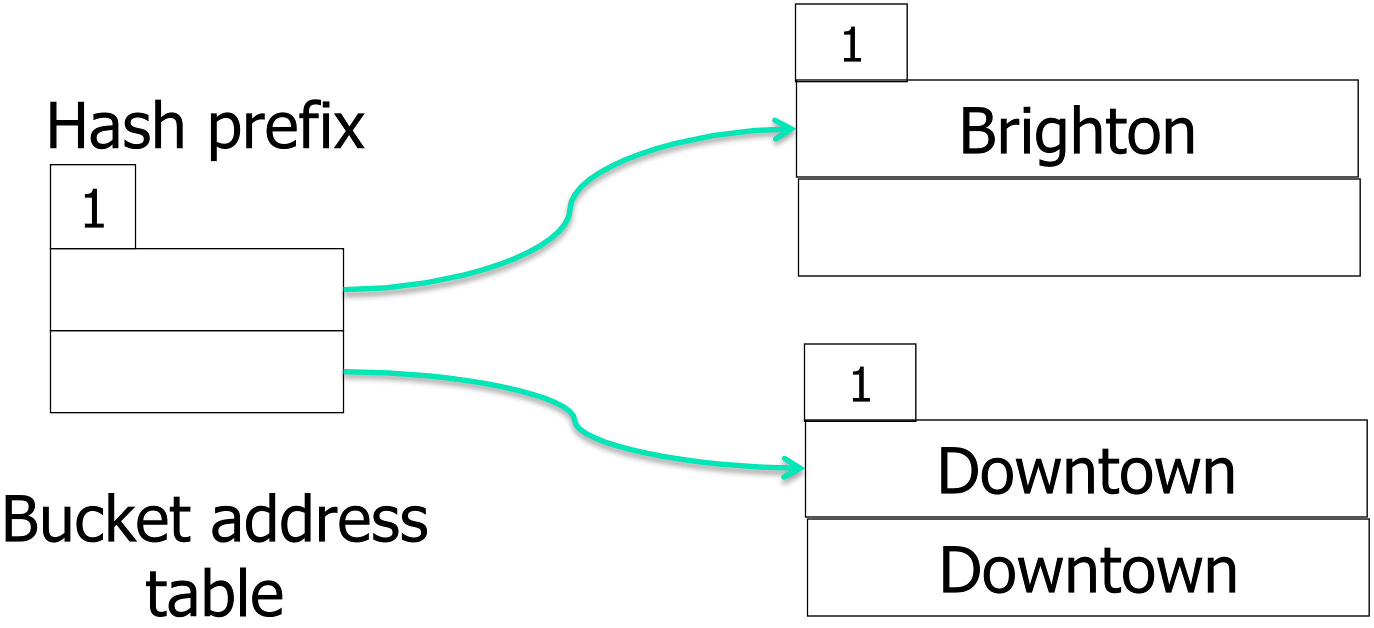
Q1 Solution cont’d

- Insert “Downtown”



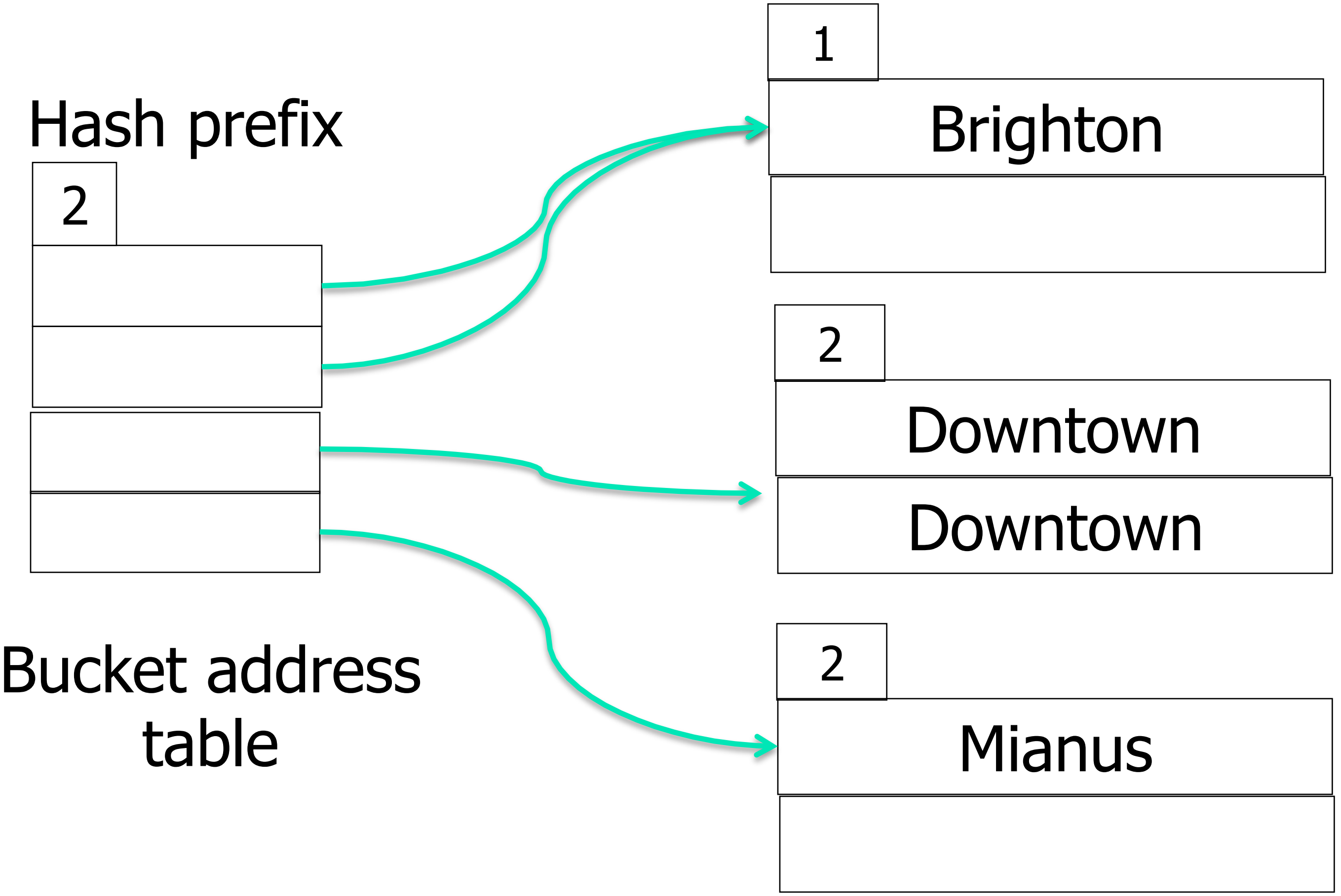
Q1 Solution cont'd

- Insert "Downtown"



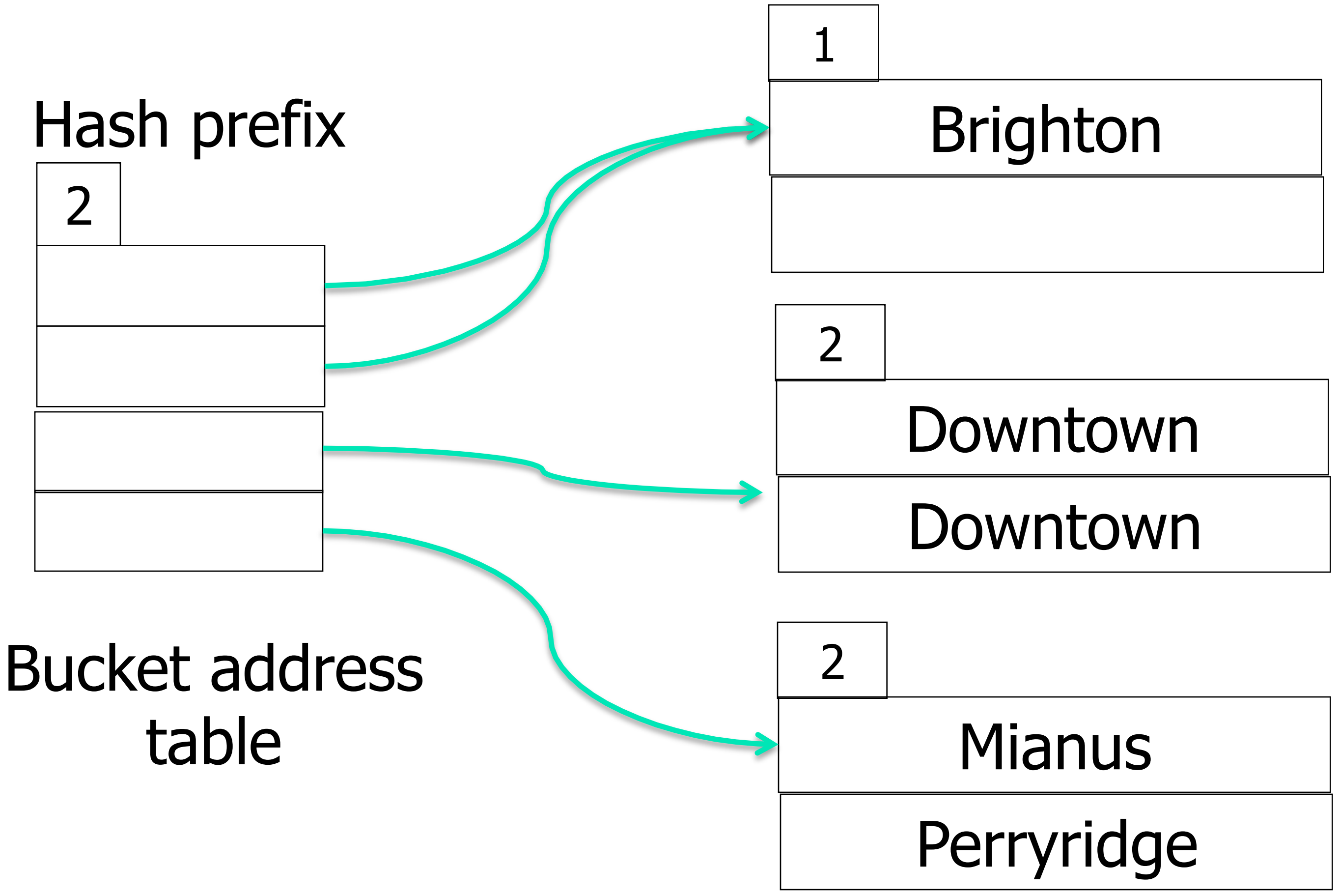
Q1 Solution cont'd

- Insert "Mianus"



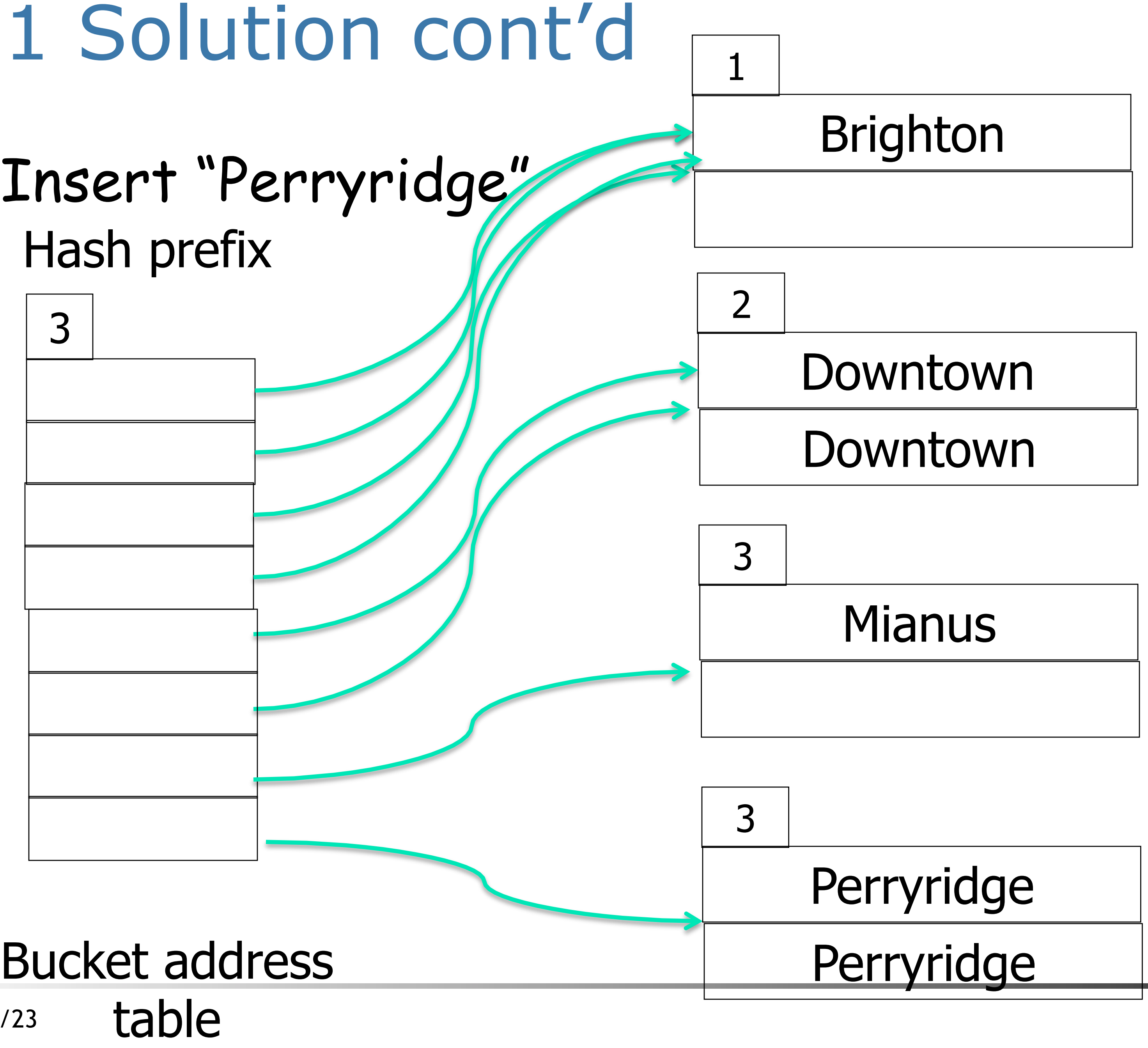
Q1 Solution cont’d

- Insert “Perryridge”



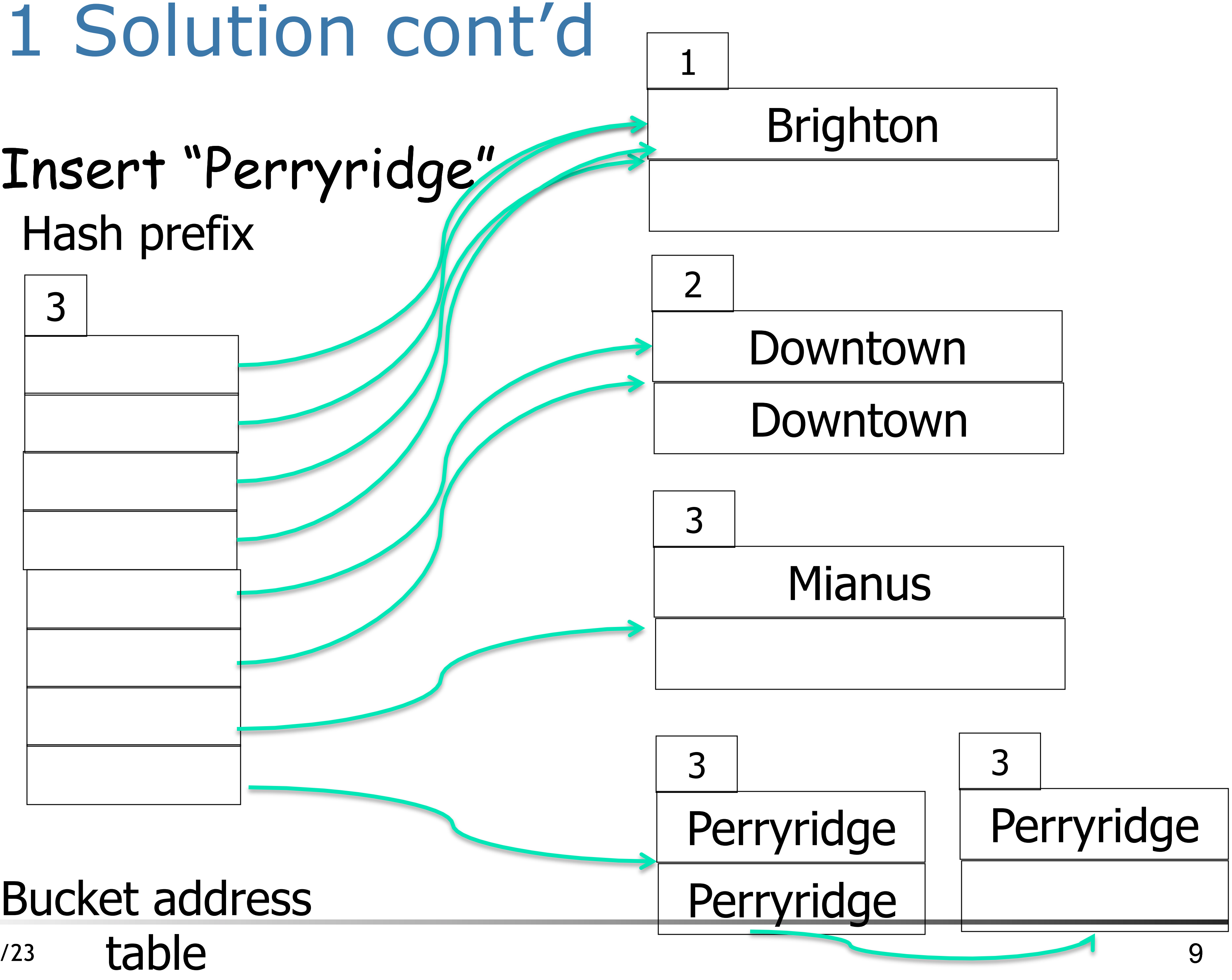
Q1 Solution cont'd

- Insert "Perryridge"



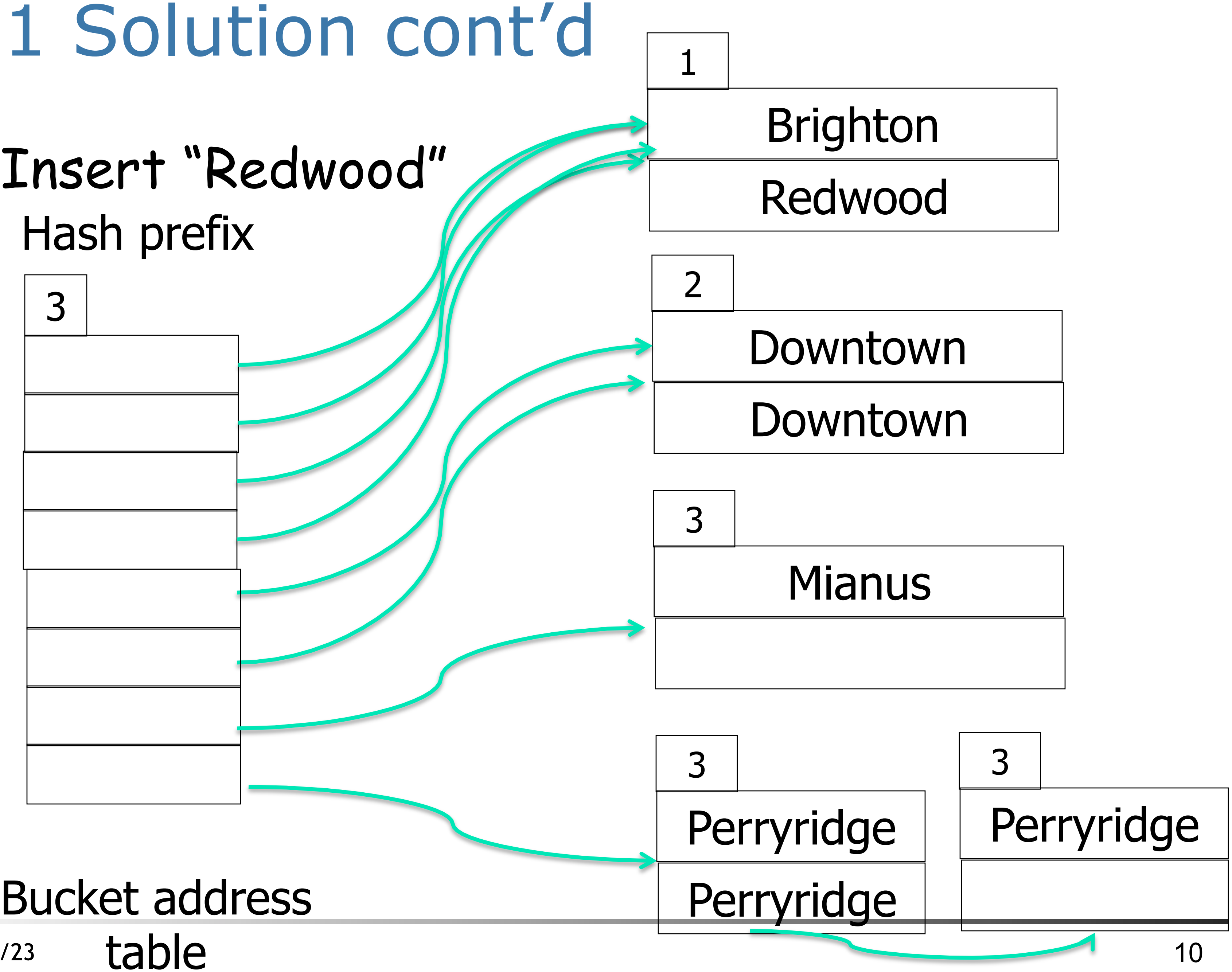
Q1 Solution cont'd

- Insert "Perryridge"



Q1 Solution cont’d

- Insert “Redwood”



Hash prefix

3

Bucket address table

1

Brighton

Redwood

2

Downtown

Downtown

3

Mianus

3

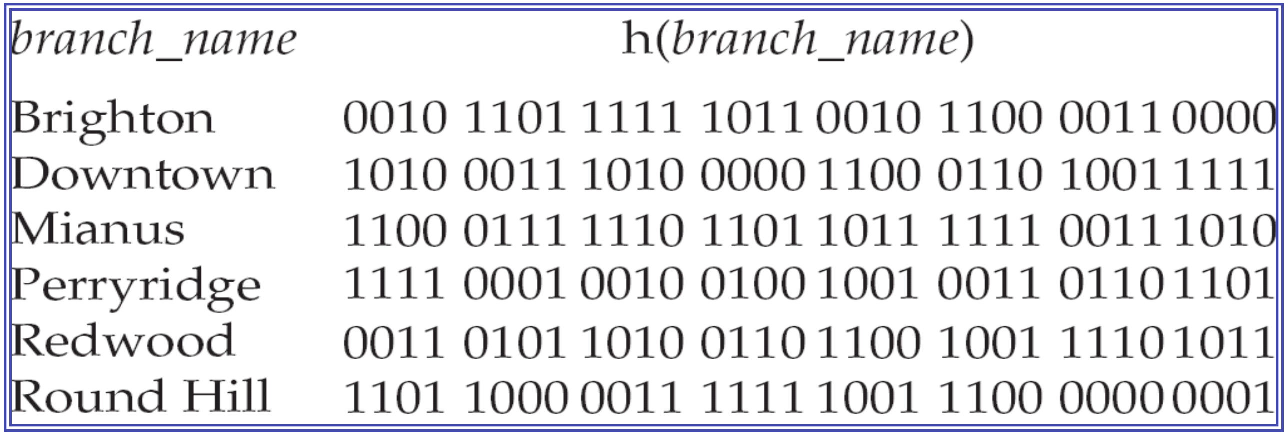
Perryridge

Perryridge

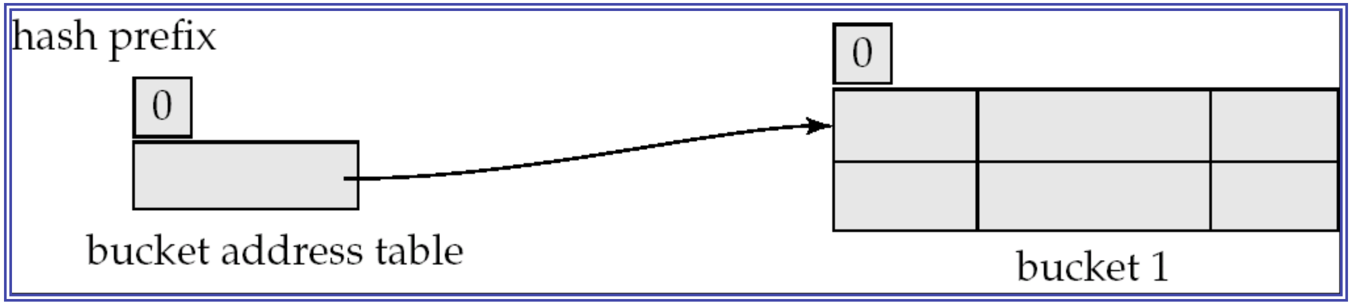
3

Perryridge

Q2



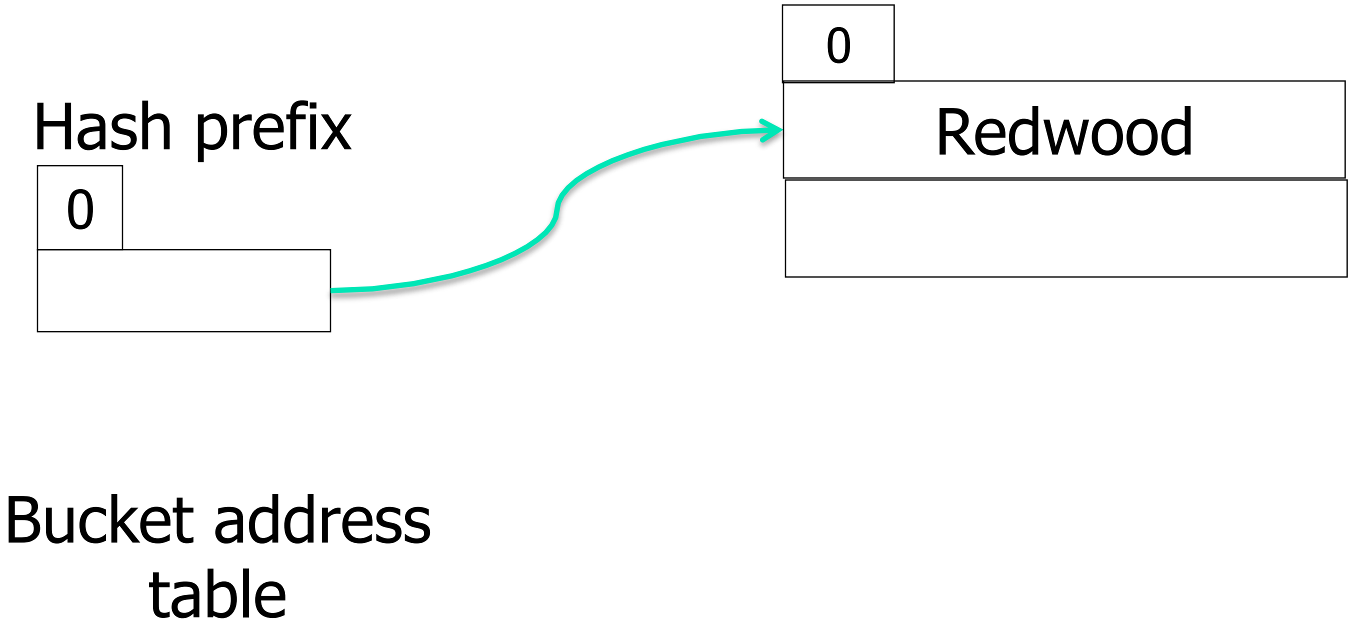
branch_name	h(*branch_name*)
Brighton	0010 1101 1111 1011 0010 1100 0011 0000
Downtown	1010 0011 1010 0000 1100 0110 1001 1111
Mianus	1100 0111 1110 1101 1011 1111 0011 1010
Perryridge	1111 0001 0010 0100 1001 0011 0110 1101
Redwood	0011 0101 1010 0110 1100 1001 1110 1011
Round Hill	1101 1000 0011 1111 1001 1100 0000 0001



- Given the above hash values for the branch_name and the initial hash index, draw the hash indices after inserting the following 8 values. Assume that one block can hold two tuples.
    - Redwood, Perryridge, Perryridge, Perryridge, Mianus, Downtown, Downtown, Brighton

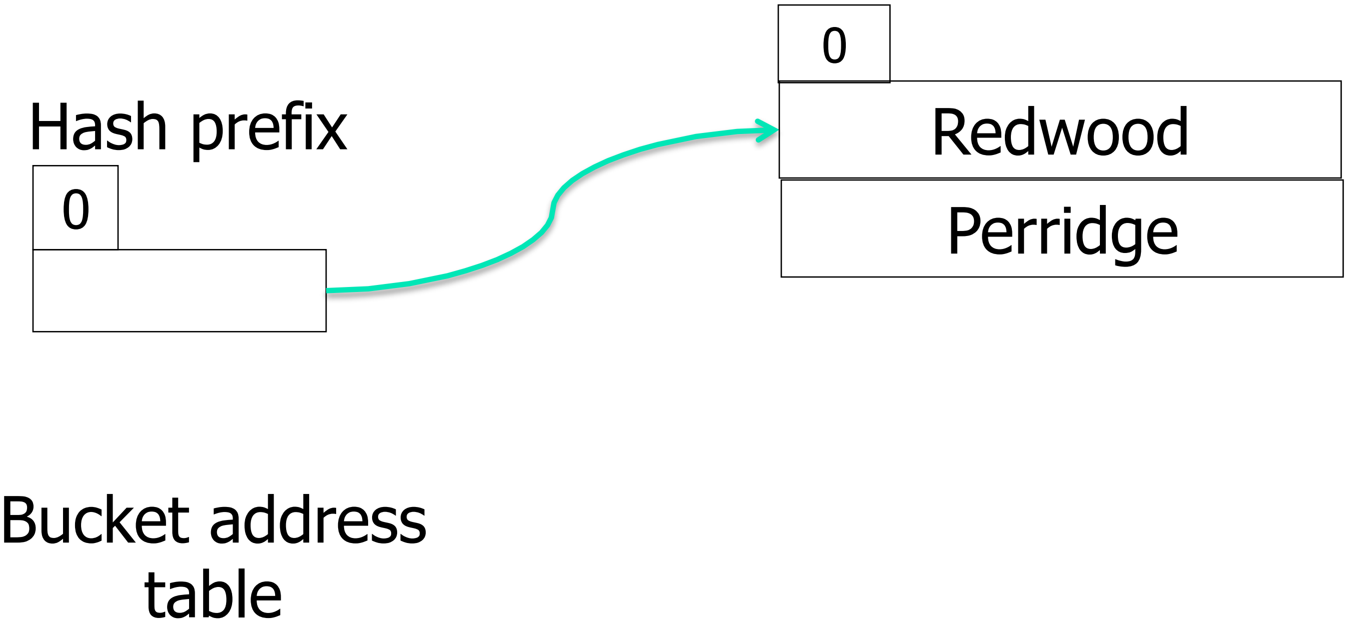
Q2 Solution

- Insert “Redwood”



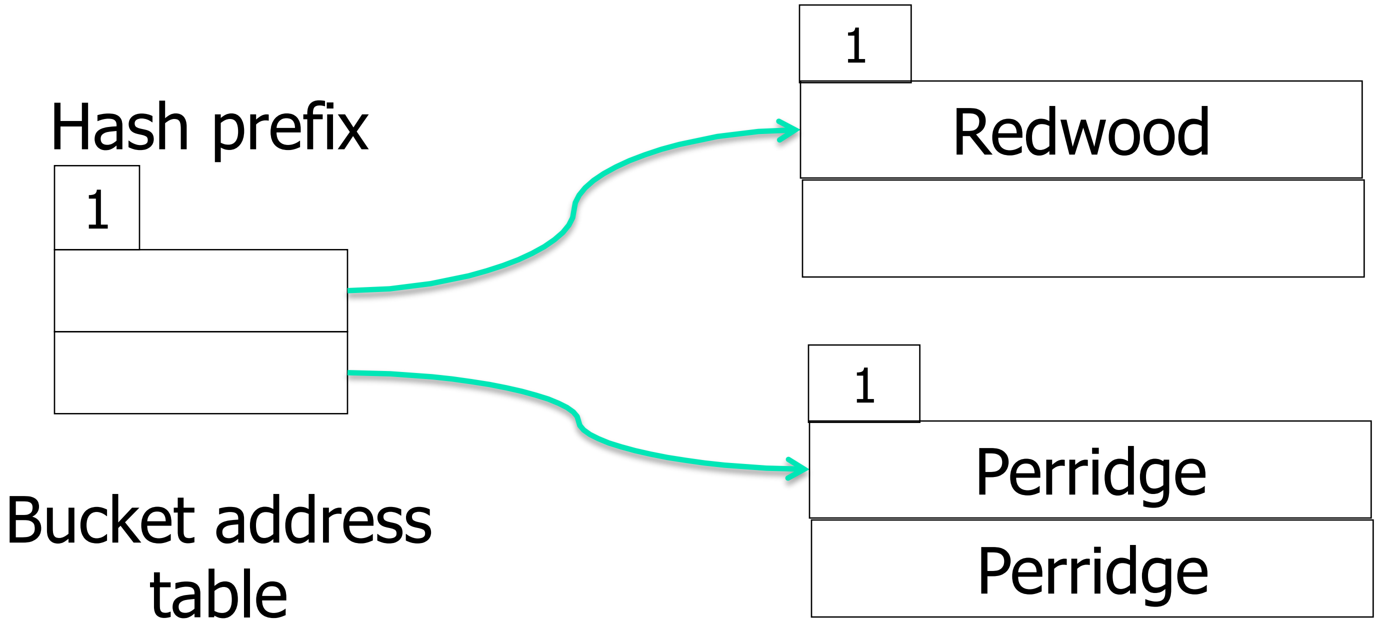
Q2 Solution cont'd

- Insert "Perridge"



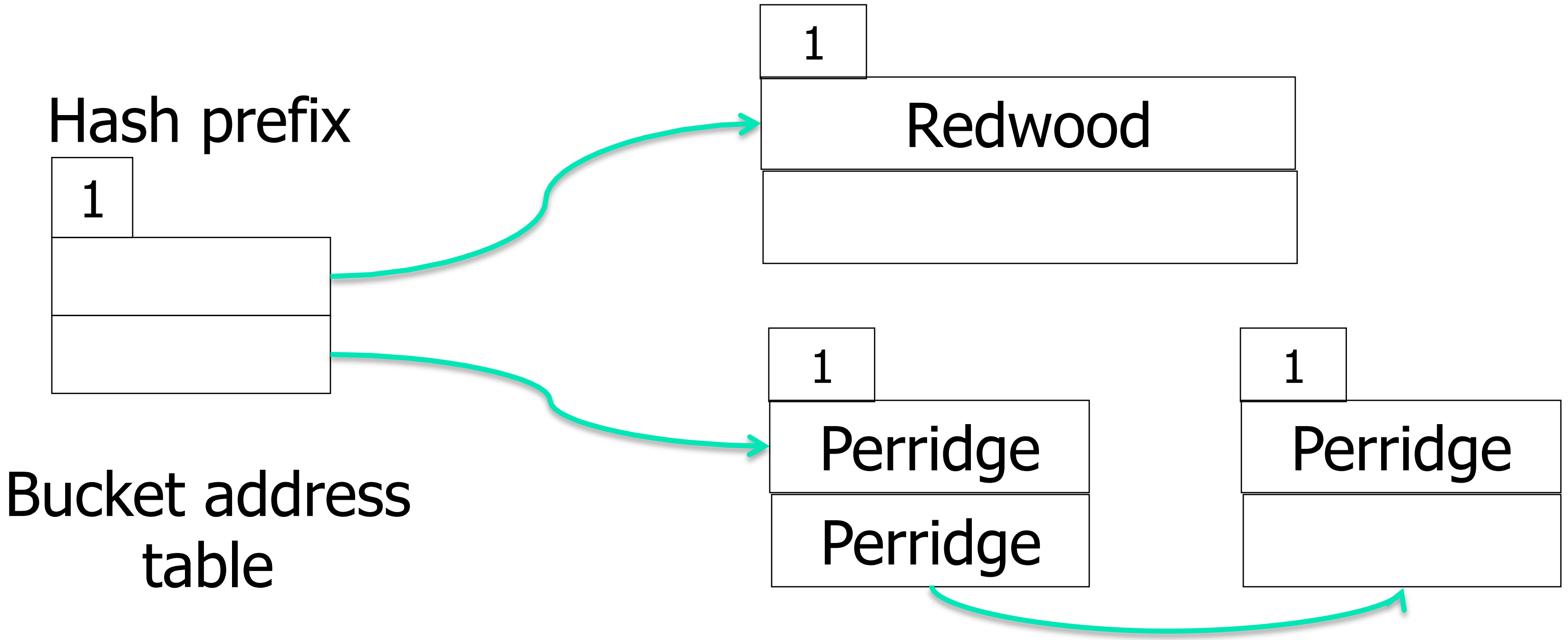
Q2 Solution cont’d

- Insert “Perridge”



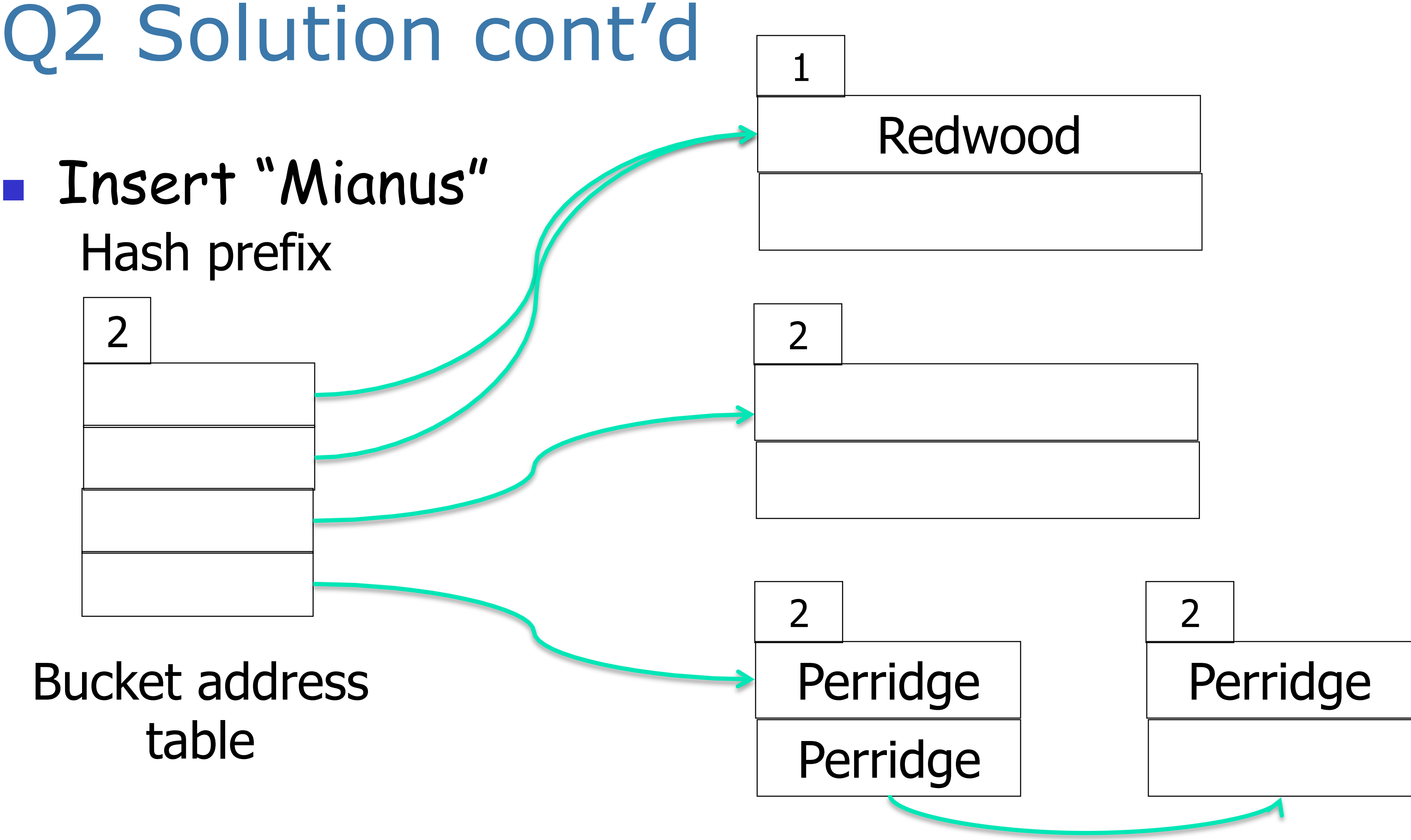
Q2 Solution cont'd

- Insert "Perridge"



Q2 Solution cont'd

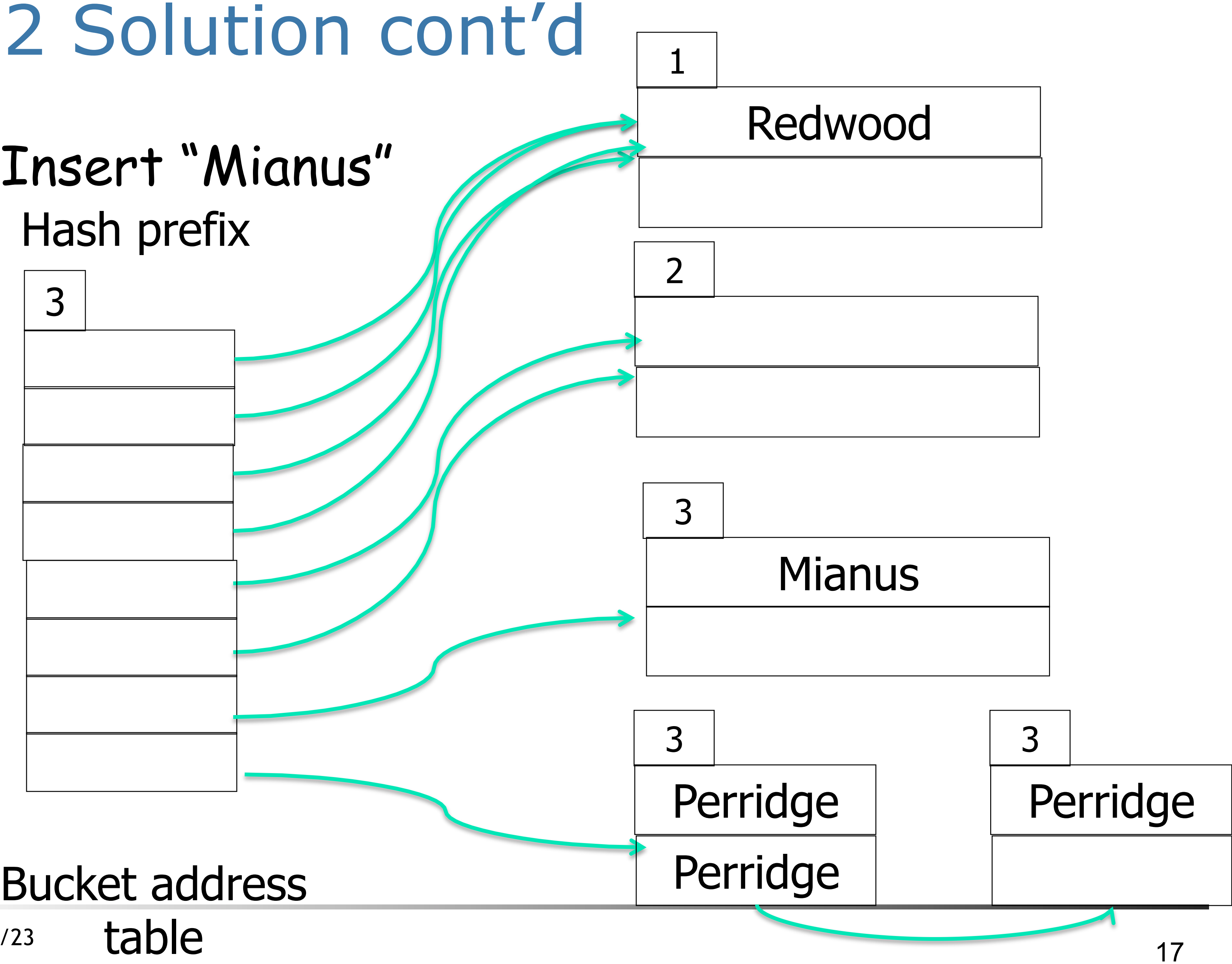
- Insert "Mianus"



NOT COMPLETE. STILL CANNOT INSERT MIANUS. SEE NEXT SLIDE.

Q2 Solution cont'd

- Insert "Mianus"



Hash prefix

3

Bucket address table

1

Redwood

2

3

Mianus

3

Perridge

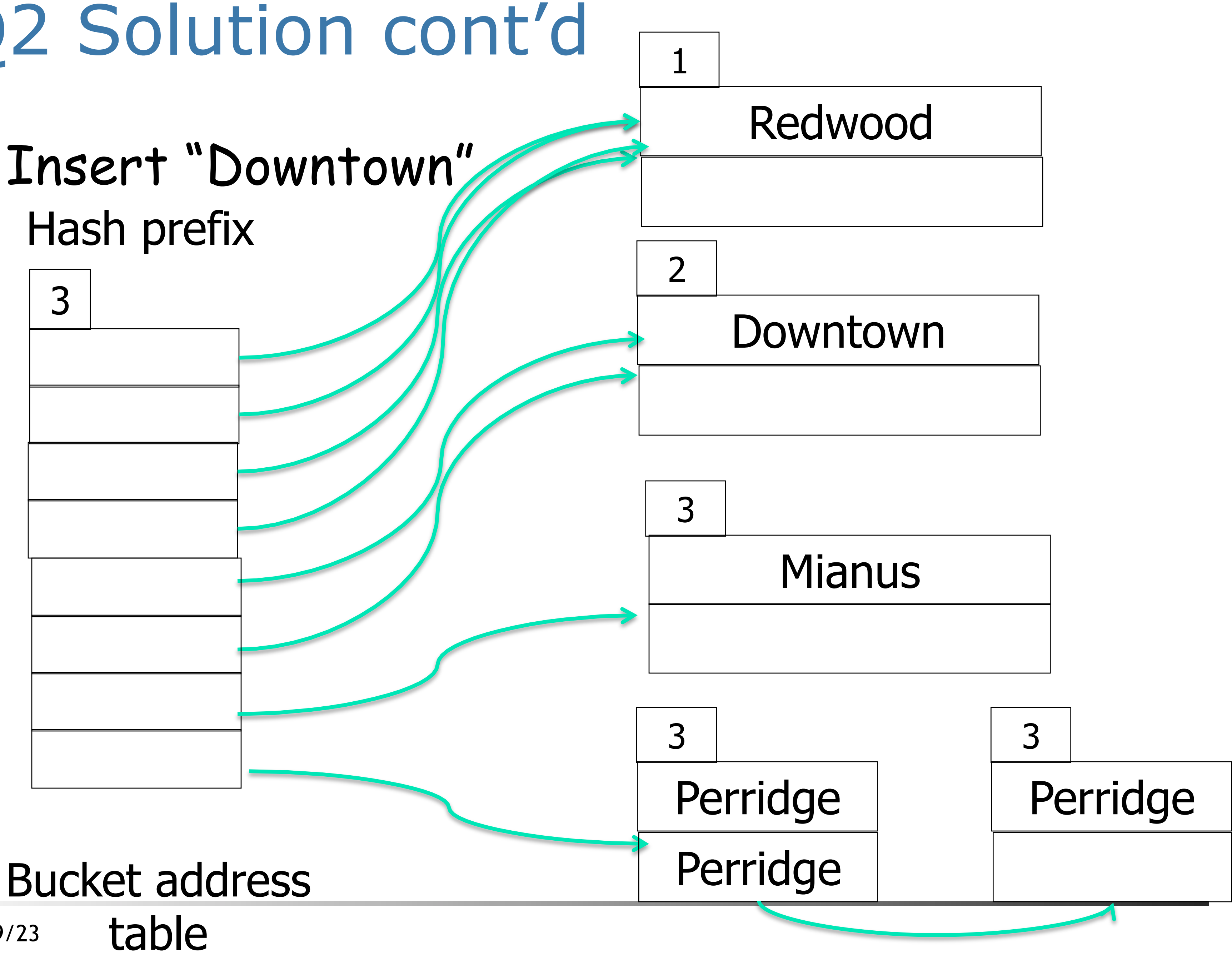
Perridge

3

Perridge

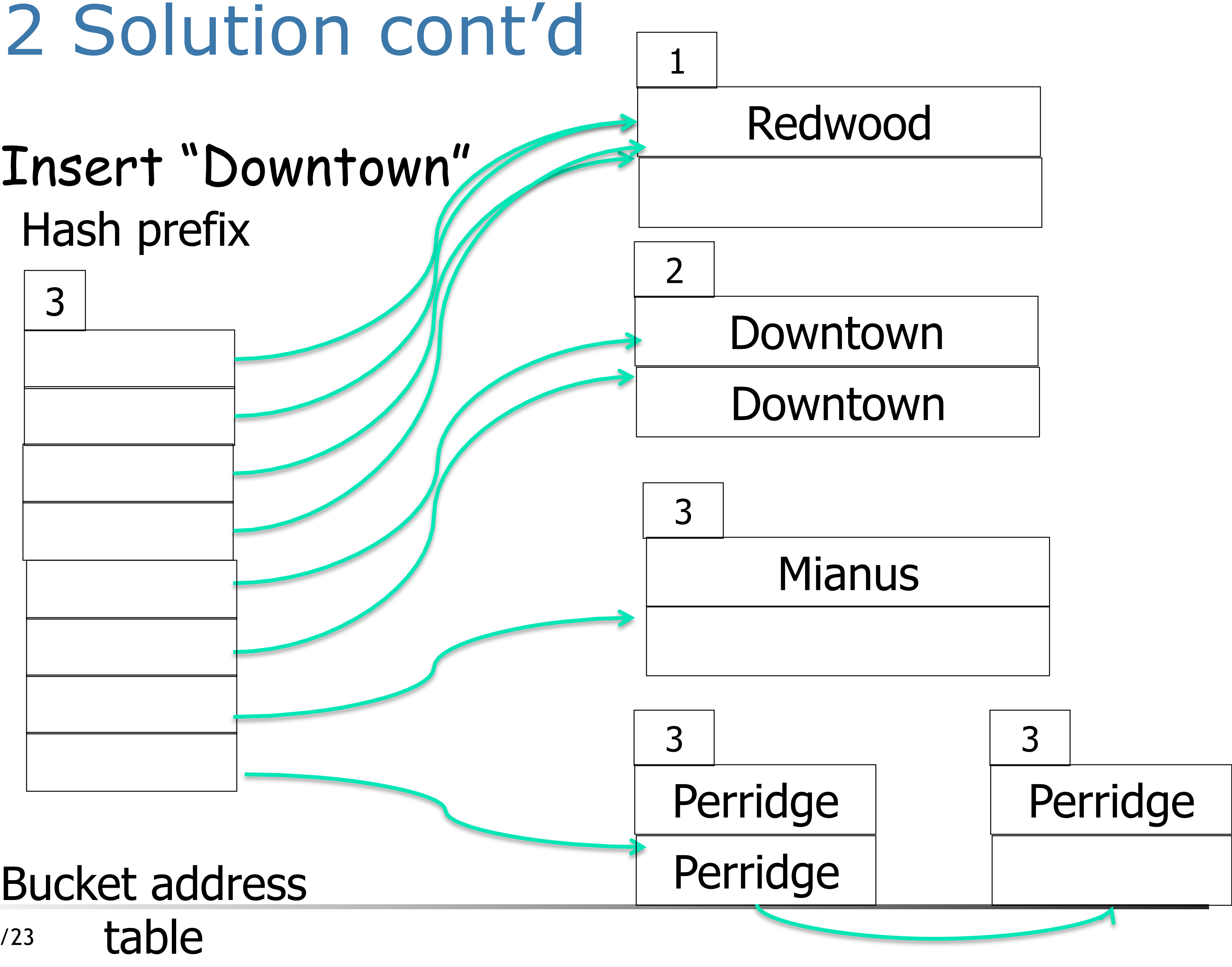
Q2 Solution cont’d

- Insert “Downtown”



Q2 Solution cont’d

- Insert “Downtown”



Hash prefix

3

1

Redwood

2

Downtown

Downtown

3

Mianus

3

Perridge

Perridge

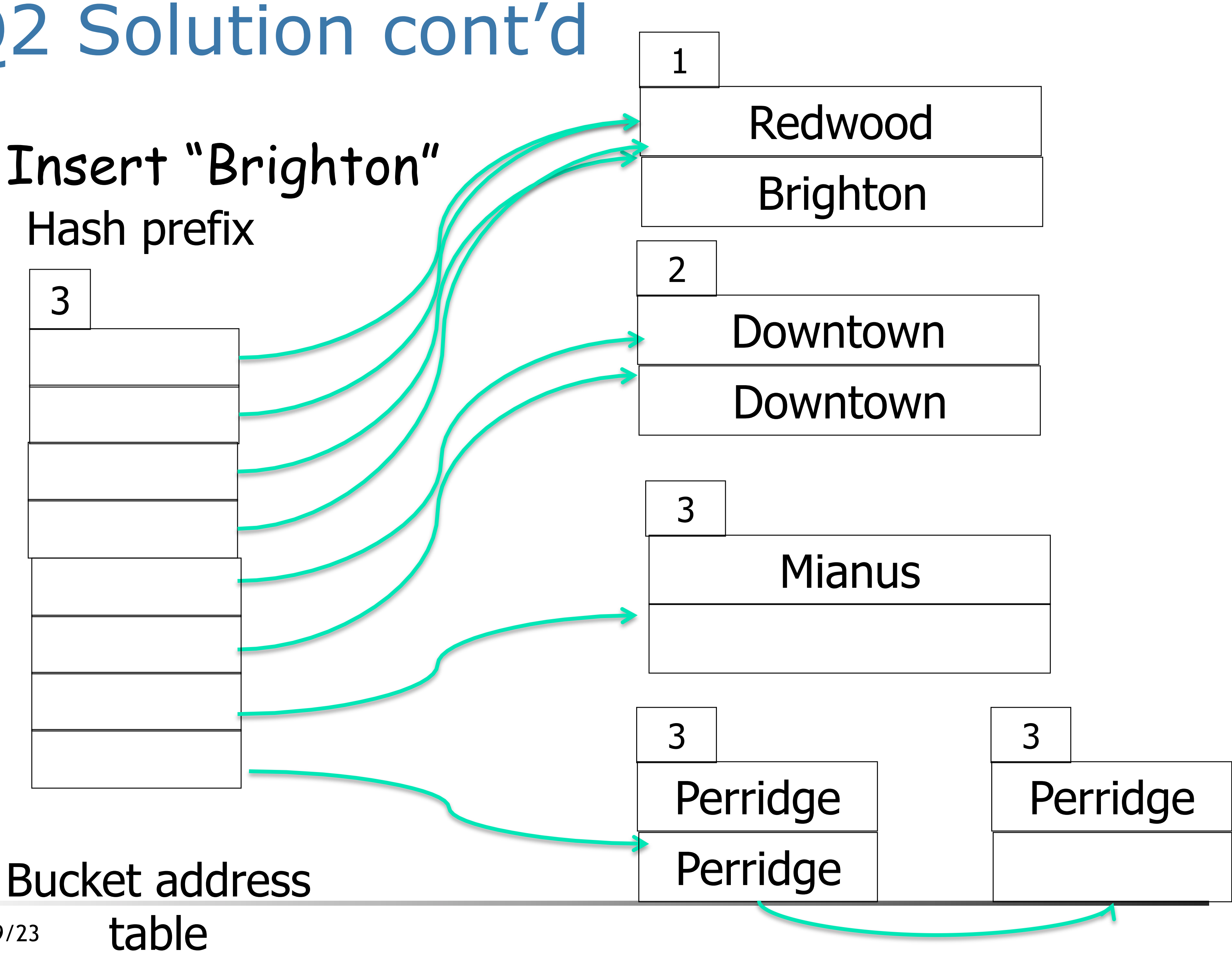
3

Perridge

Bucket address table

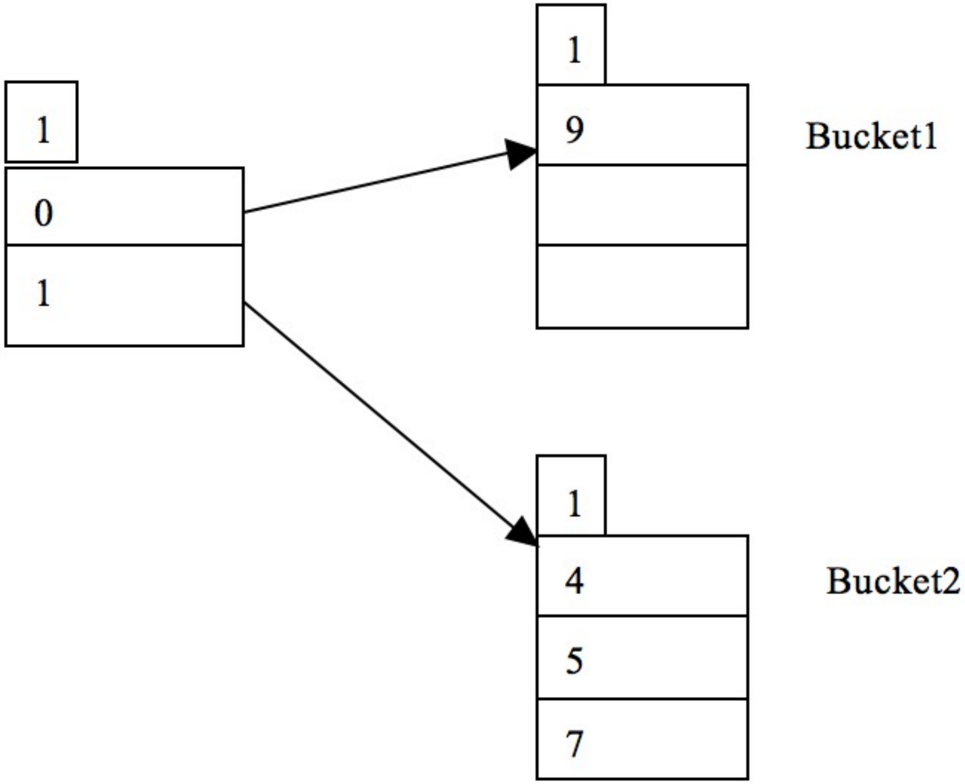
Q2 Solution cont'd

- Insert "Brighton"



Q3

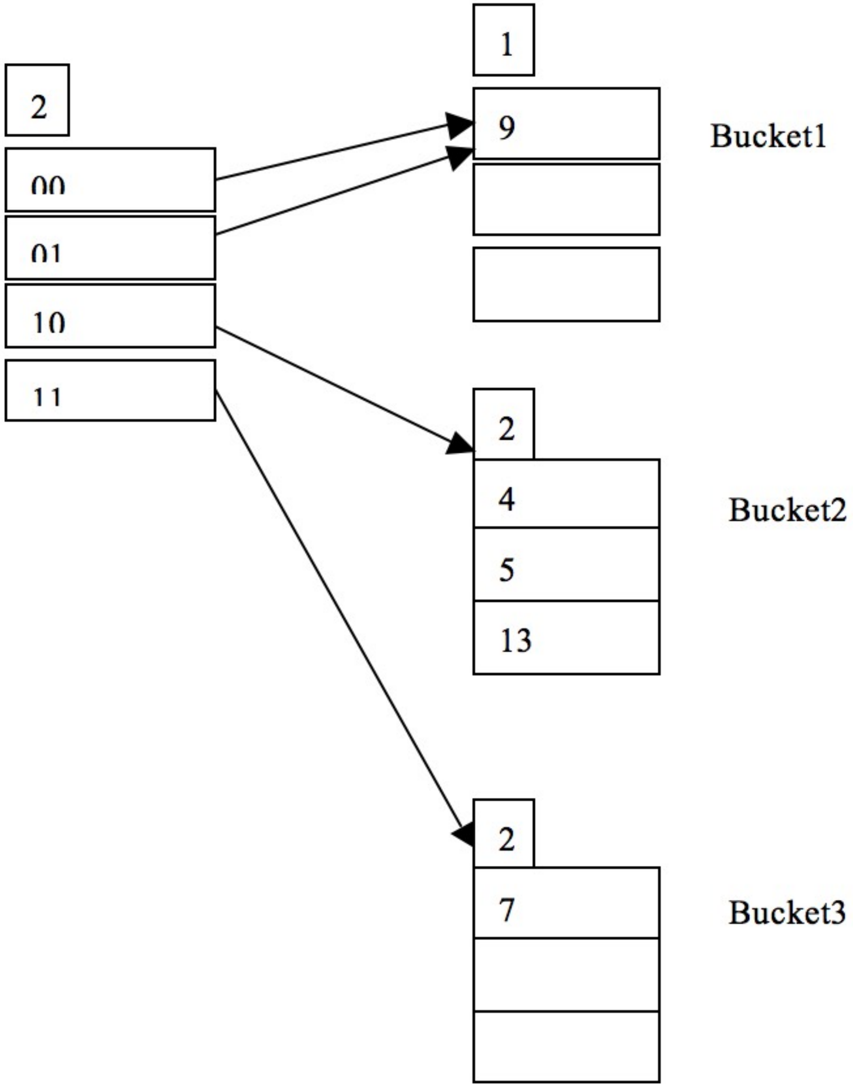
- Consider the extendable hashing with hash function h(x) = x mod 8 and a bucket can hold three records. Draw the hash index after inserting 13.



Q3 Solution

- Insertion of 13
  - h(13) = 13 mod 8 = 101, 13 should be in bucket 2, but there is no space and the bucket 2 also can't be split so we have to extend the structure by
  - Doubling the size of address table
  - Splitting the bucket 2 and redistributing the entries in it
  - Insert 13 in bucket 2

Q3 cont'd


1
2
00
01
10
11
9
Bucket1
2
4
5
13
Bucket2
2
7
Bucket3