Q1 Solutions

- Insert 11

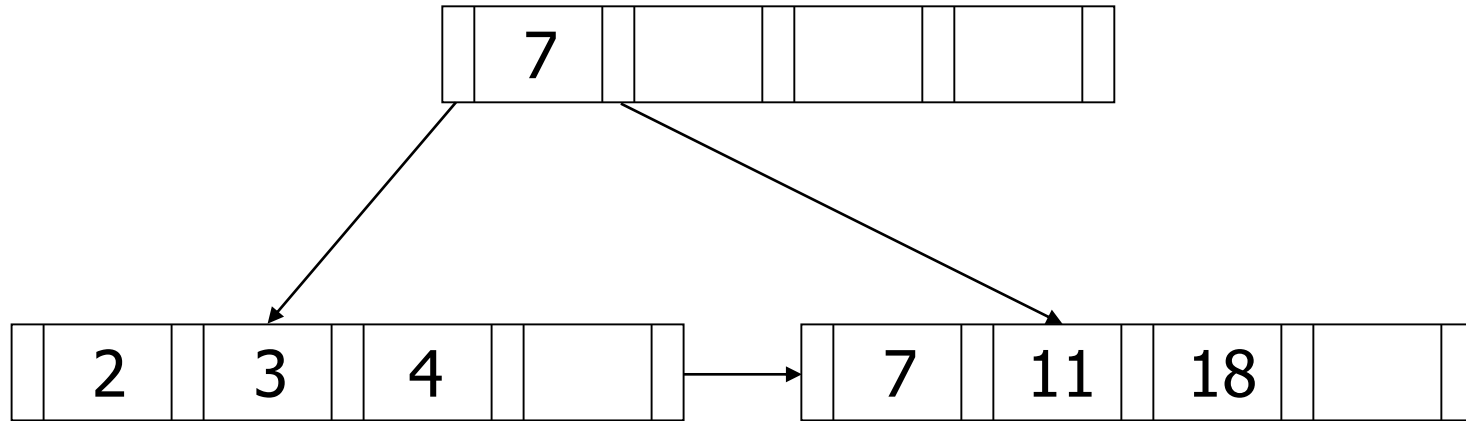|	2	3	4	11	

Q1 Solutions – cont'd
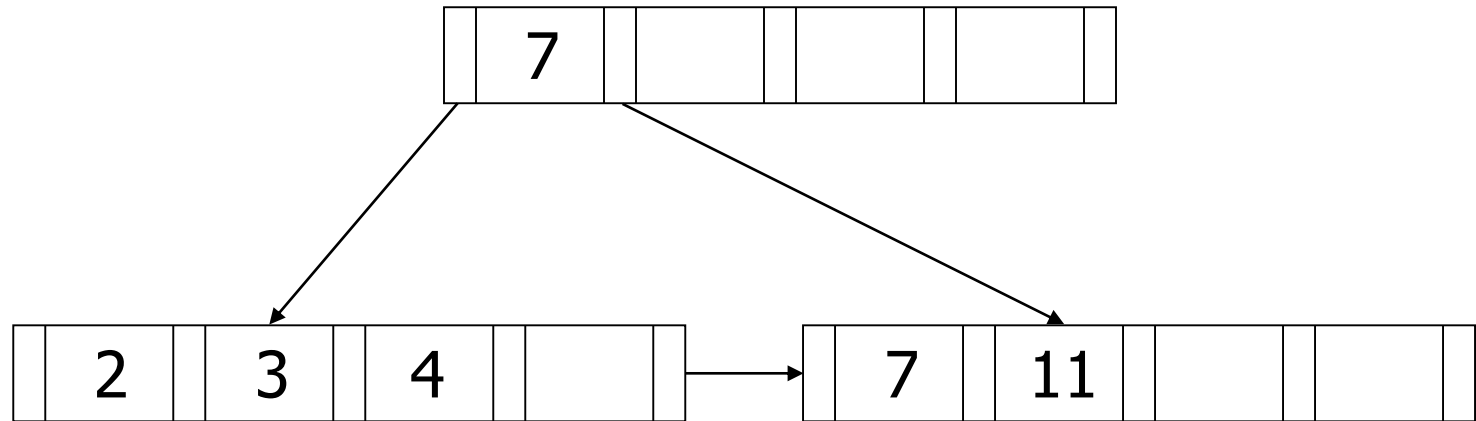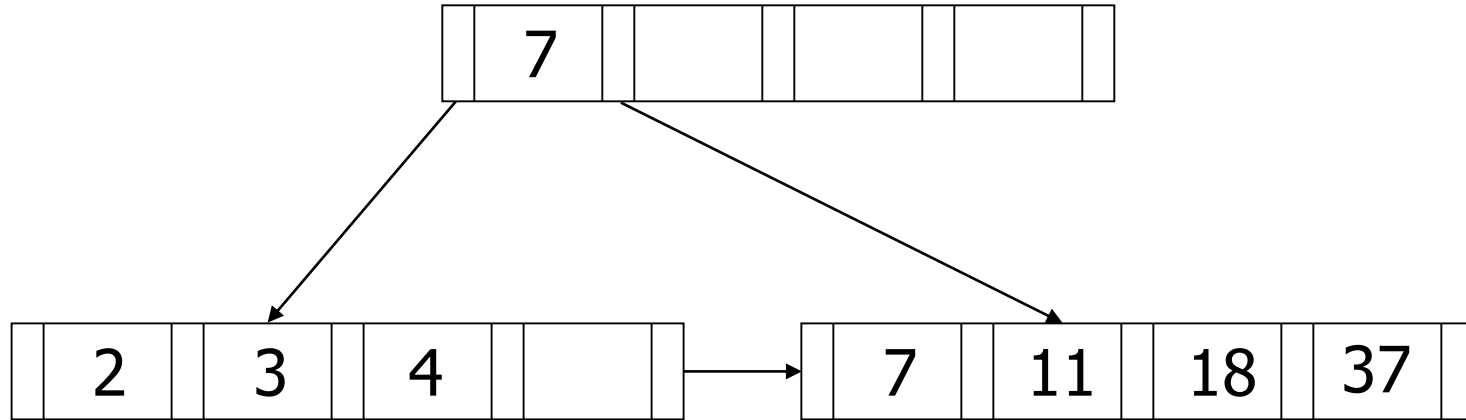
- Insert 7

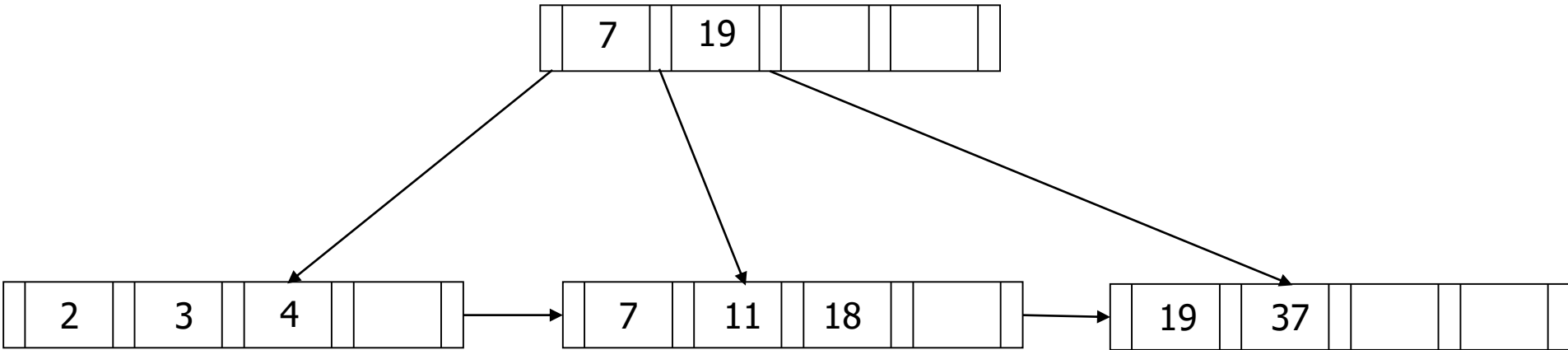# Q1 Solutions – cont'd

- Insert 18

Q1 Solutions – cont'd

- Insert 37

Q1 Solutions – cont'd

- Insert 19

Q1 Solutions – cont'd

- **Insert 8**

	7		19				

	2		3		4		

	7		8		11		18	

	19		37				

Q1 Solutions – cont'd

- Insert 77

Q1 Solutions – cont'd

- Insert 80

	7		19			

2	3	4		→	7	8	11	18		→	19	37	77	80

Q1 Solutions – cont'd

- Insert 28

Q1 Solutions – cont'd

- Insert 79

The B+ tree diagram:
- Root node: 7 | 19 | 77
- Leaf nodes: [2 | 3 | 4] → [7 | 8 | 11 | 18] → [19 | 28 | 37] → [77 | 79 | 80]
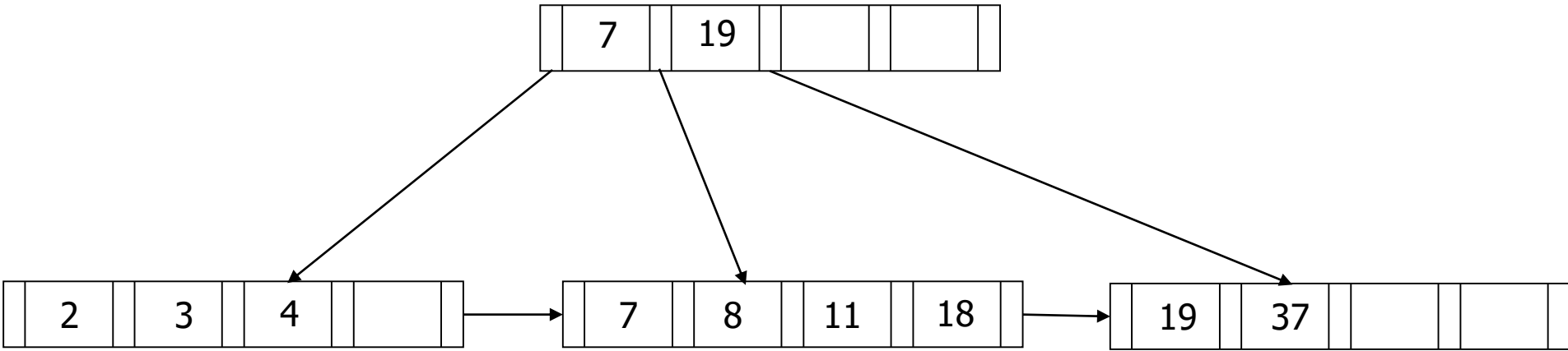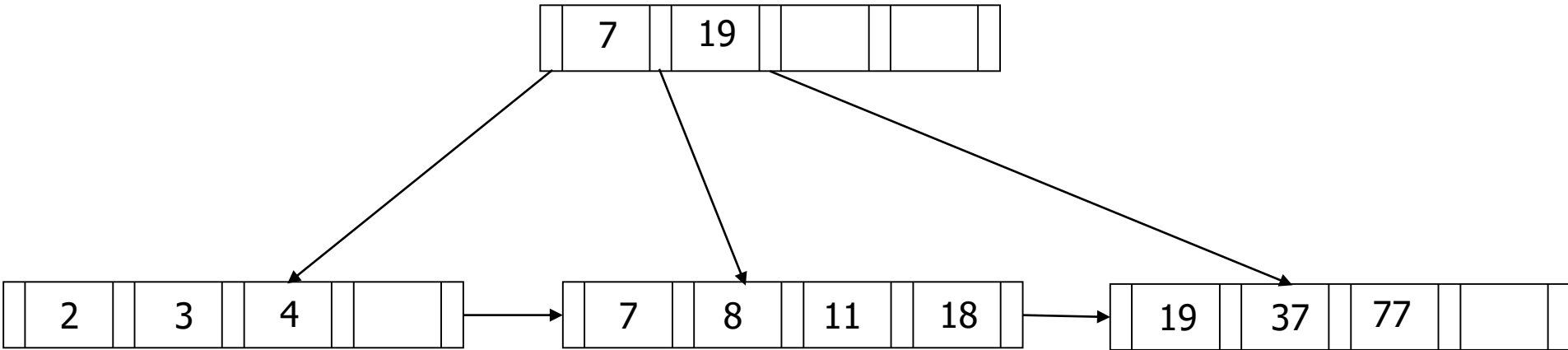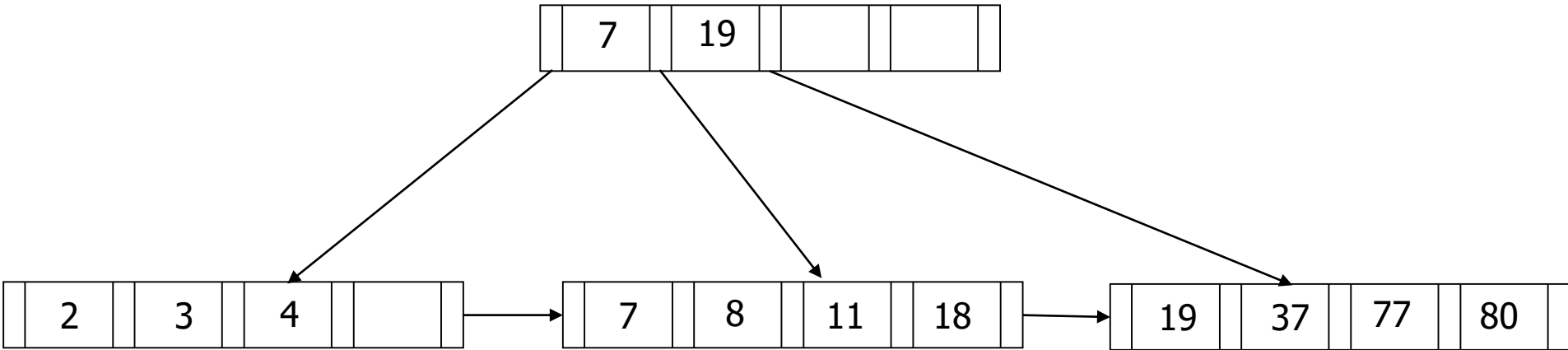
Q1 Solutions – cont'd
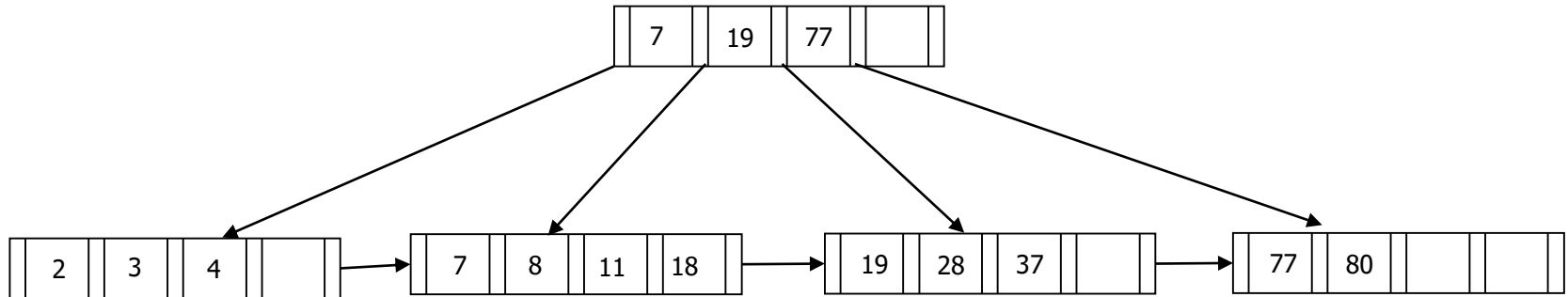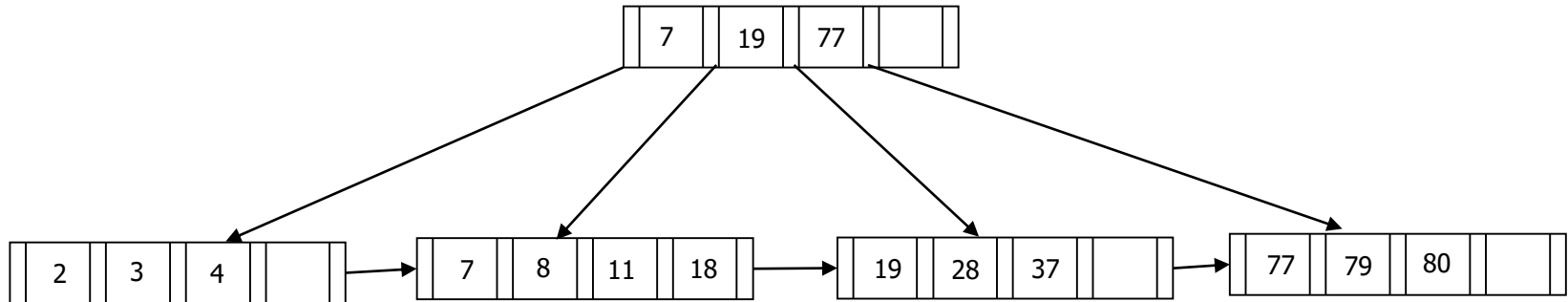
- Insert 88

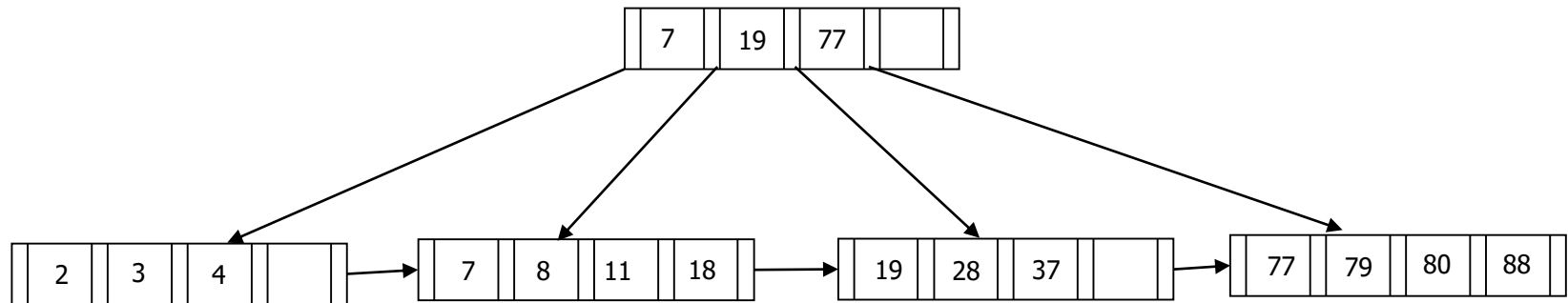# Q1 Solutions – cont'd

- Insert 4

Q1 Solutions – cont'd

■ Insert 6

Q1 Solutions – cont'd

■ Insert 44

7	19	77	

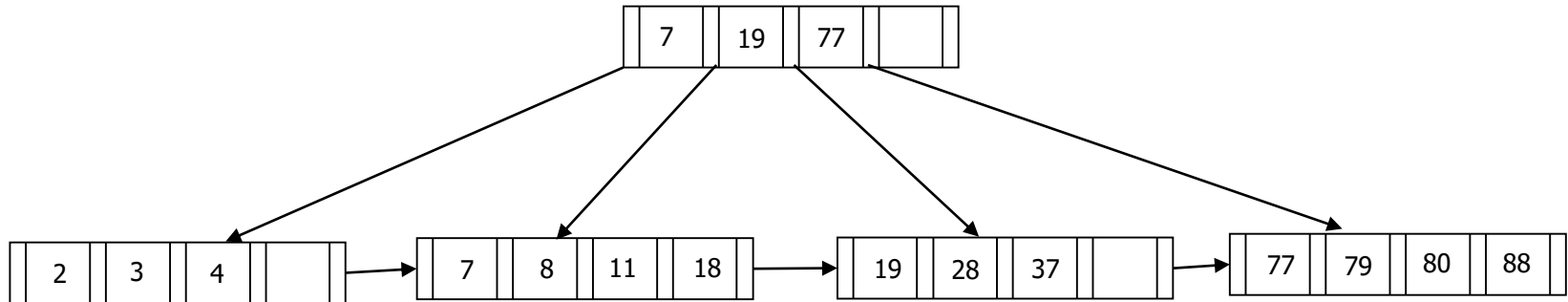| 2 | 3 | 4 | 6 | → | 7 | 8 | 11 | 18 | → | 19 | 28 | 37 | 44 | → | 77 | 79 | 80 | 88 |
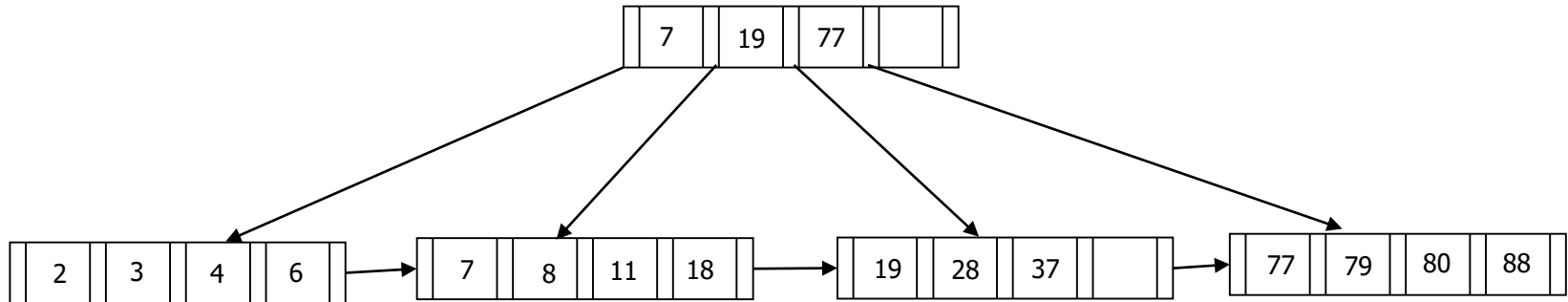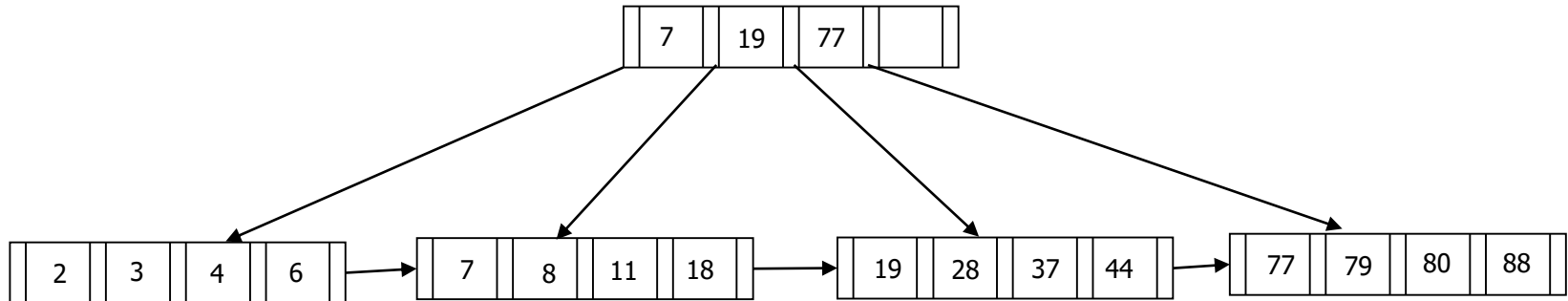
Q1 Solutions – cont'd
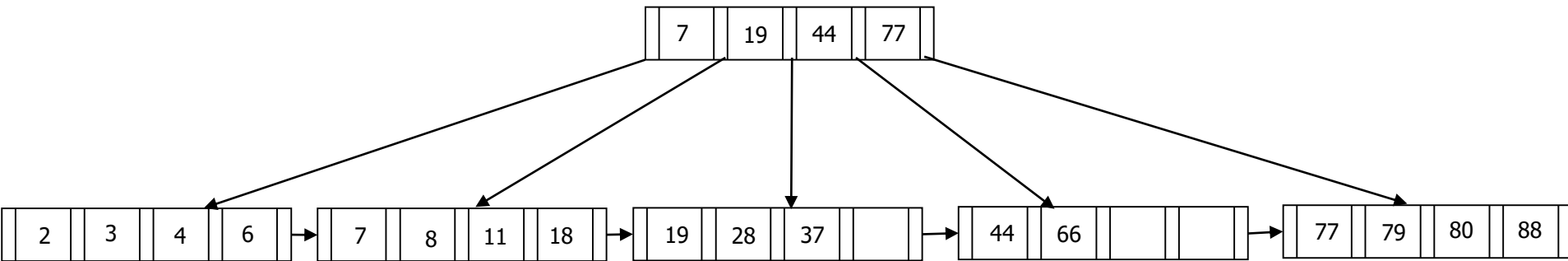
- Insert 66

Q1 Solutions – cont'd

- Insert 98
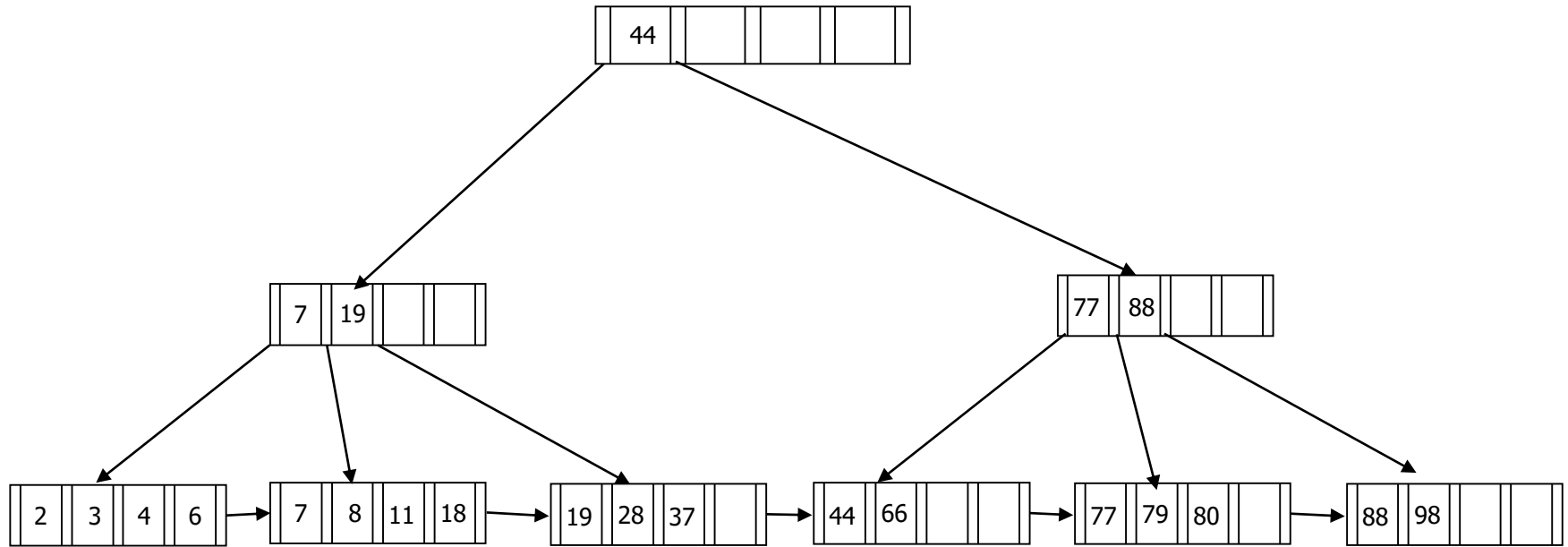
Q1 Solutions – cont'd

- Delete 19
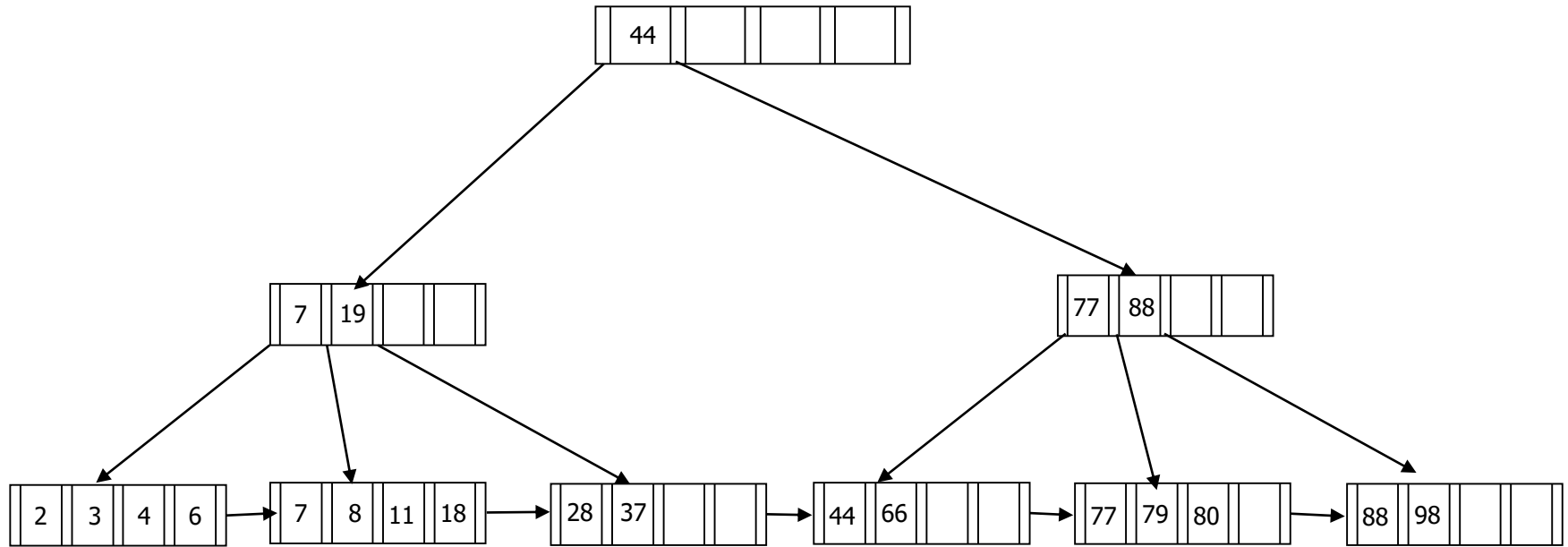
Q1 Solutions – cont'd

- Delete 21

Q1 Solutions – cont'd

- Delete 88

| | 7 | 19 | 44 | 77 | |

| 2 | 3 | 4 | 6 | → | 7 | 8 | 11 | 18 | → | 28 | 37 | | | → | 44 | 66 | | | → | 77 | 79 | 80 | 98 |
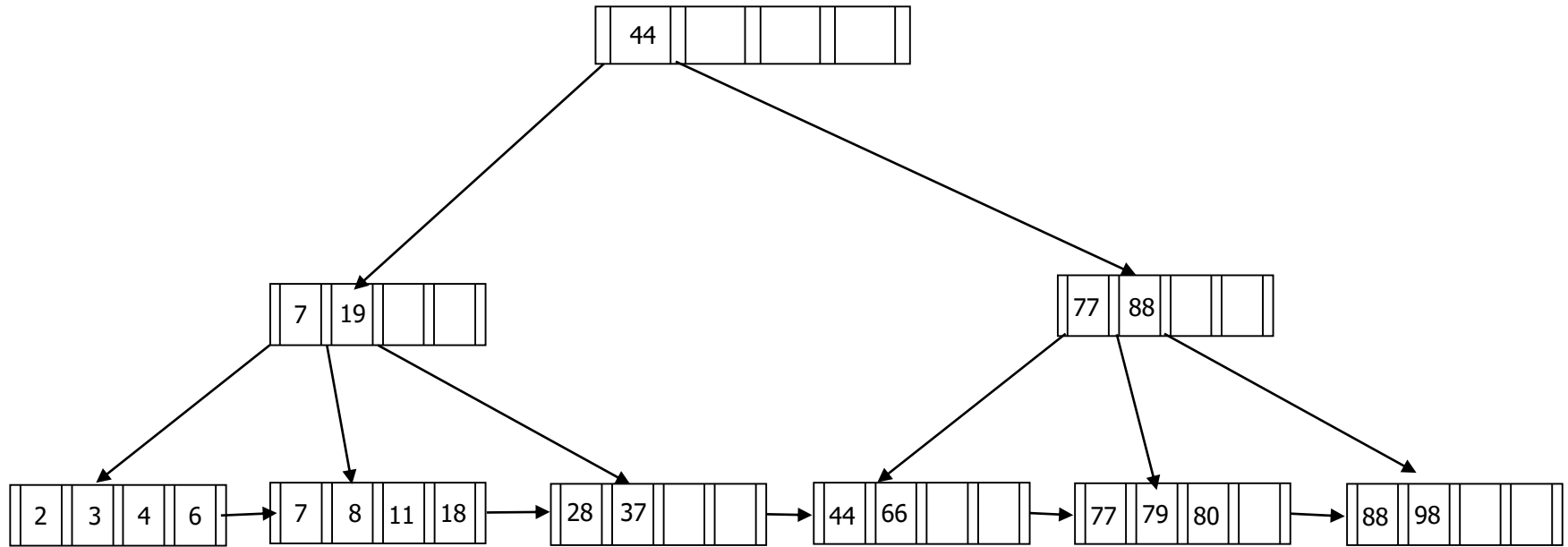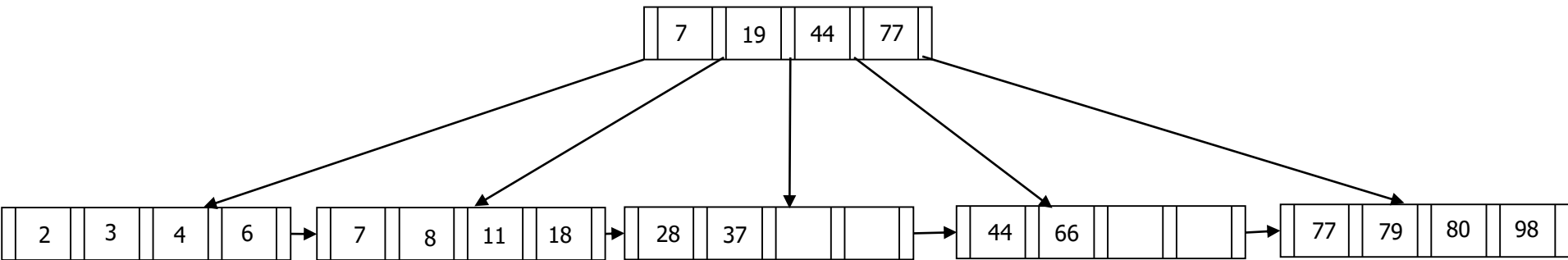
Q1 Solutions – cont'd

- Delete 3
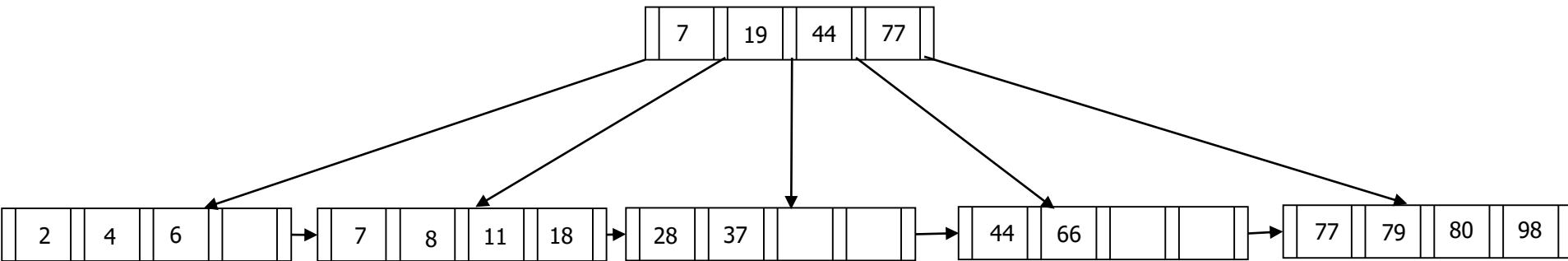
Q2 Solution

- Insert "Taiyuan"

  i = 0
  Hash Prefix: $2^0 = 1$

0

  Bucket Address Table

0
Taiyuan

Q2 Solution cont'd

- Insert "Cambridge"

i = 0
Hash Prefix: $2^0 = 1$

0

Bucket Address Table
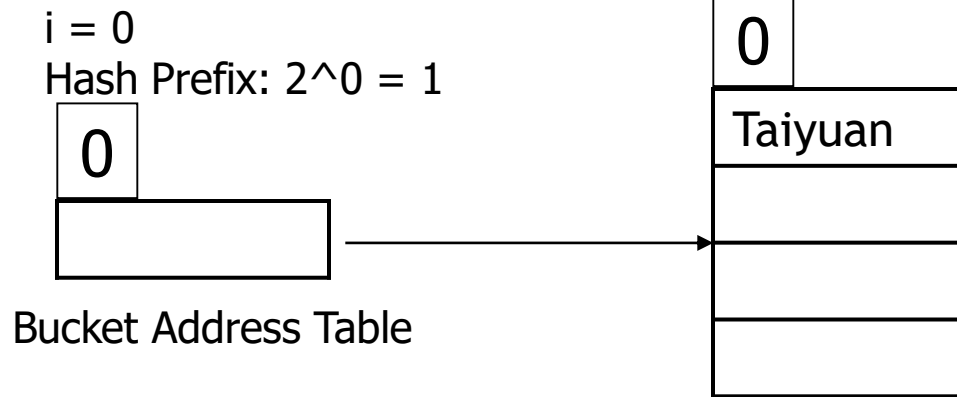
0
Taiyuan
Cambridge

Q2 Solution cont'd

- ■ Insert "Kuala Lumpur"

i = 0
Hash Prefix: $2^0 = 1$

0

0	

Bucket Address Table

0

Taiyuan
Cambridge
Kuala Lumpur

Q2 Solution cont'd

- **Insert "Florence"**

i = 0
Hash Prefix: $2^0 = 1$

0

0

Bucket Address Table
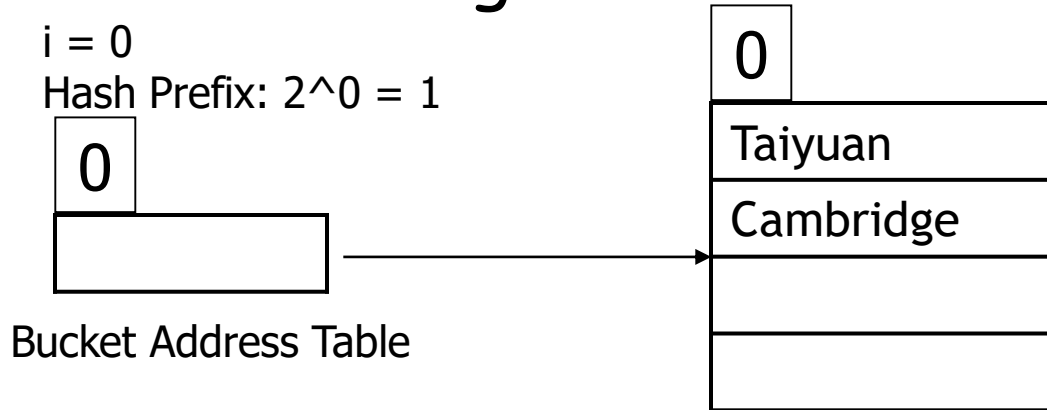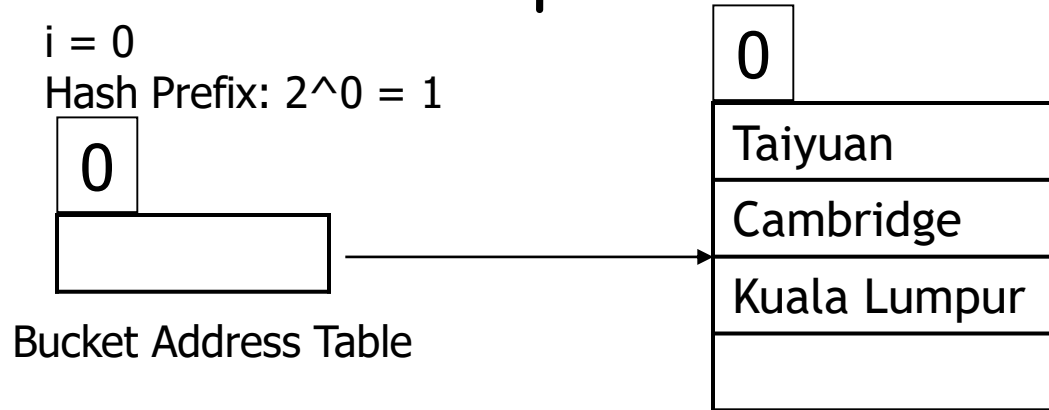
0
Taiyuan
Cambridge
Kuala Lumpur
Florence
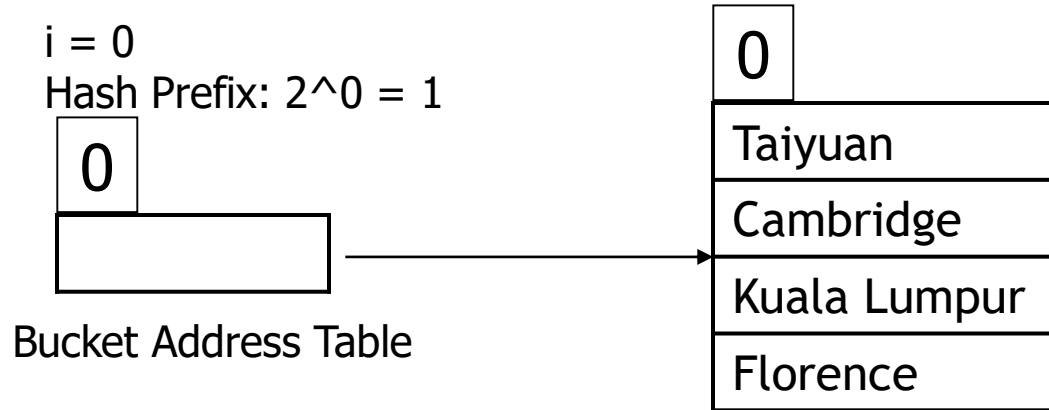
Q2 Solution cont'd

- Insert "Lianyungang"

i = 1
Hash Prefix: $2^1 = 2$

1

0
1

Bucket Address Table

1

Taiyuan
Cambridge
Florence

1

Kuala Lumpur
Lianyungang

Q2 Solution cont'd

- Insert "Nottingham"

i = 1
Hash Prefix: 2^1 = 2

1
0
1

Bucket Address Table

1
Taiyuan
Cambridge
Florence
Nottingham

1
Kuala Lumpur
Lianyungang
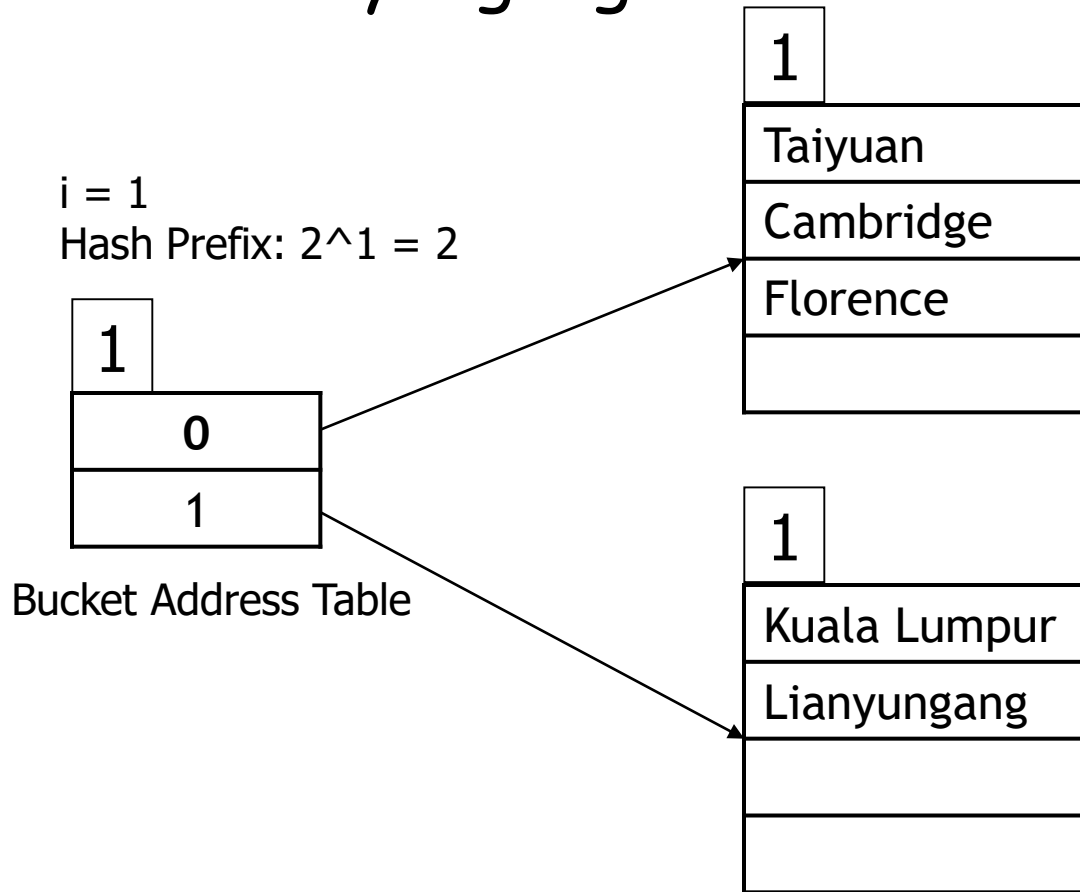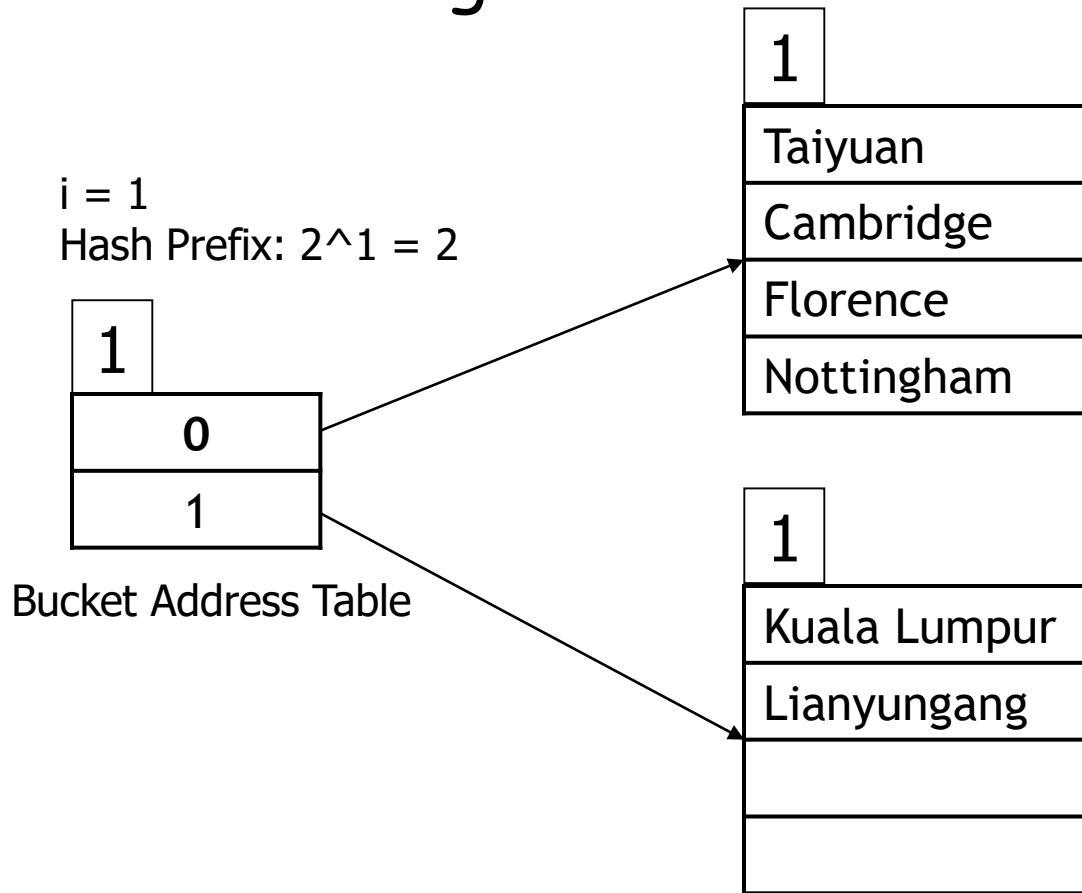
Q2 Solution cont'd

- Insert "Taiyuan"

i = 2
Hash Prefix: $2^2 = 4$

Bucket Address Table

2
00
01
10
11

2
Taiyuan
Florence
Nottingham
Taiyuan

2
Cambridge

1
Kuala Lumper
Lianyungang
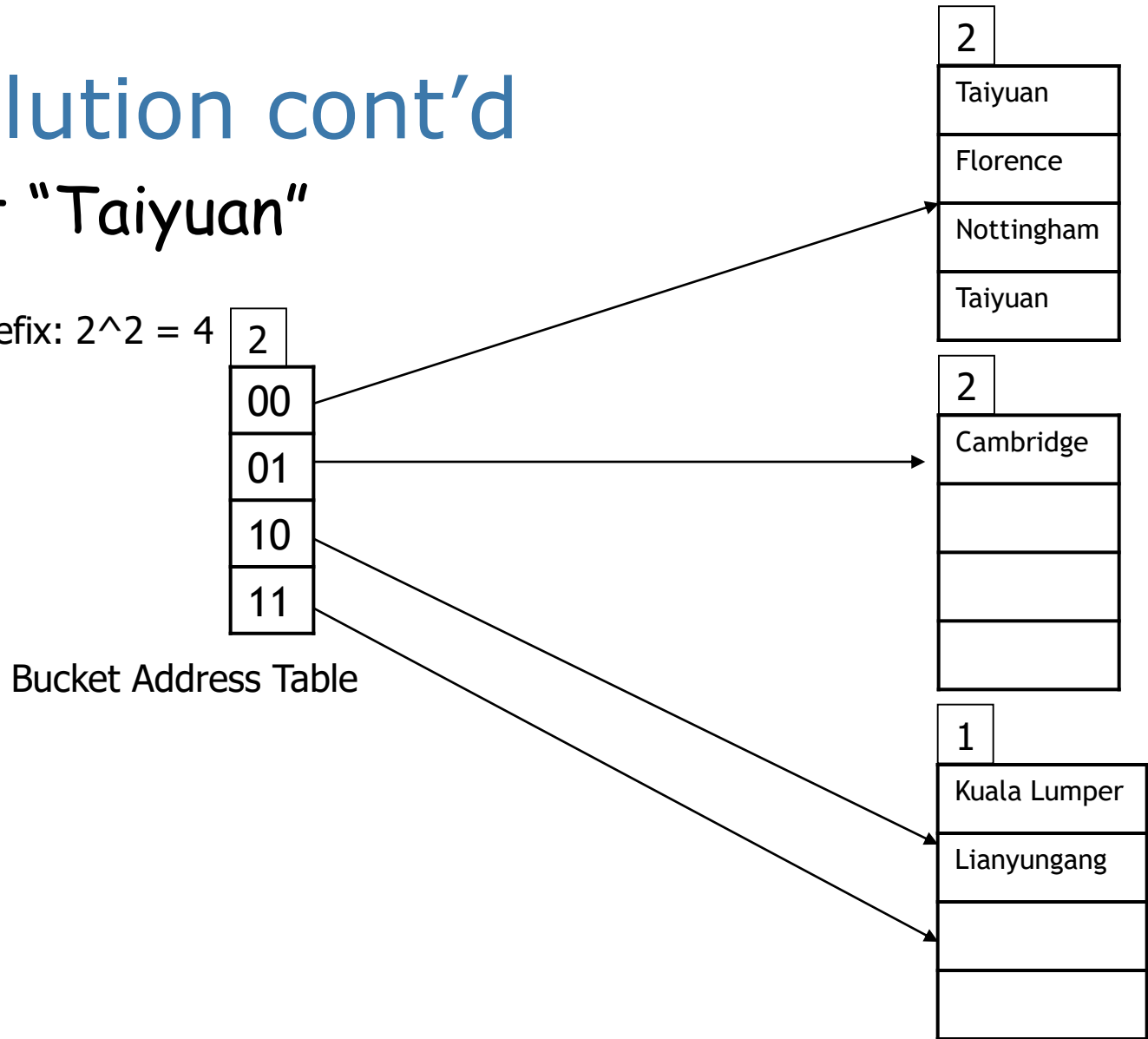
Q2 Solution cont'd

- ## Insert "Cambridge"

i = 2
Hash Prefix: 2^2 = 4

Bucket Address Table

2
00
01
10
11

2
Taiyuan
Florence
Nottingham
Taiyuan

2
Cambridge
Cambridge

1
Kuala Lumper
Lianyungang
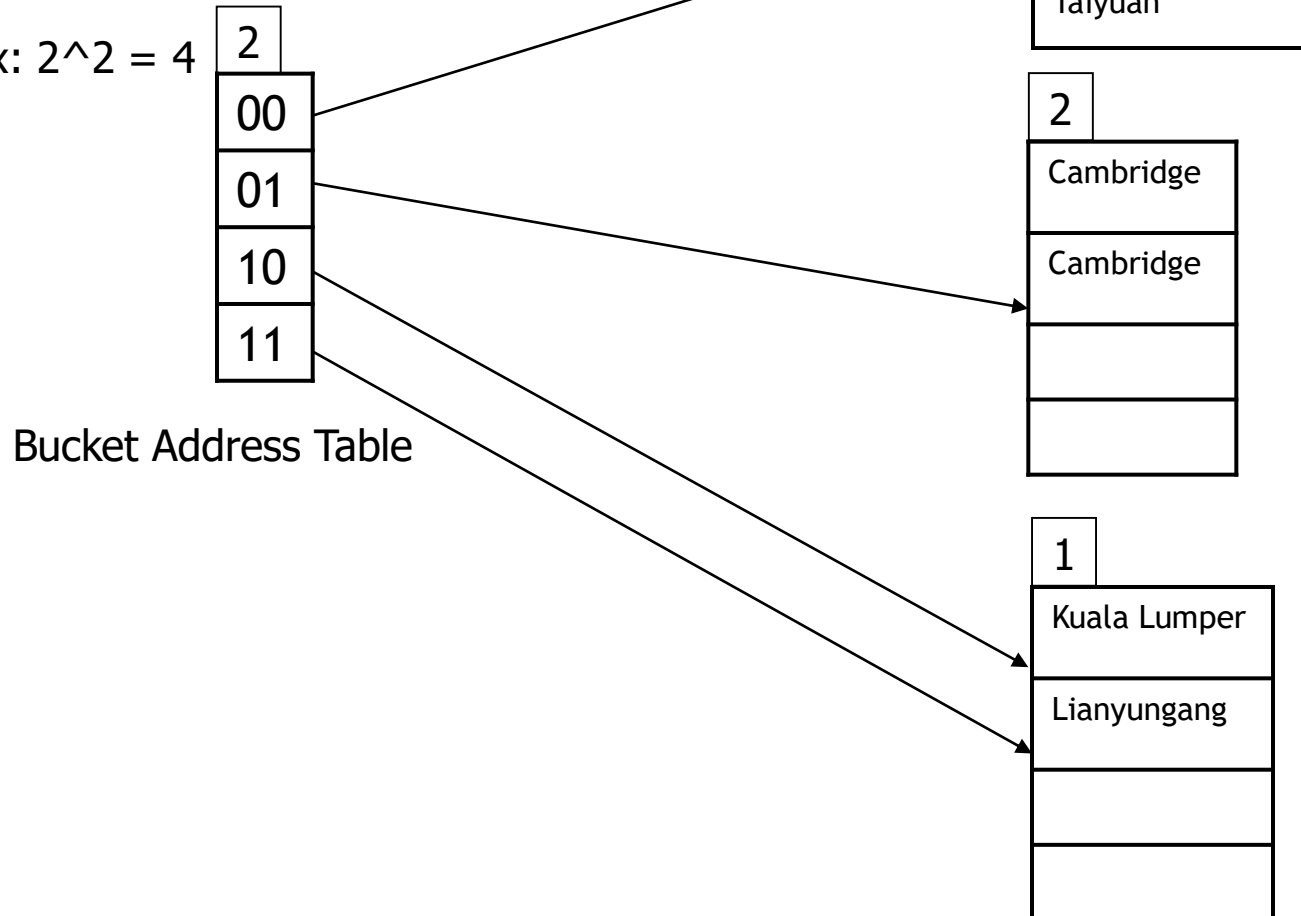
Q2 Solution cont'd

- Insert "Venice"

i = 4
Hash Prefix: 2^4 = 16

Bucket Address Table

4
0000
0001
0010
0011
0100
0101
0110
0111
1000
1001
1010
1011
1100
1101
1110
1111

3

4

Taiyuan
Florence
Nottingham
Taiyuan

4

Venice

2

Cambridge
Cambridge

1

Kuala Lumper
Lianyungang

Q2 Solution cont'd

- Insert "Kuala Lumpur"

i = 4
Hash Prefix: 2^4 = 16

Bucket Address Table

4

0000
0001
0010
0011
0100
0101
0110
0111
1000
1001
1010
1011
1100
1101
1110
1111

3

4

Taiyuan
Florence
Nottingham
Taiyuan

4

Venice

2

Cambridge
Cambridge

1

Kuala Lumper
Lianyungang
Kuala Lumpur
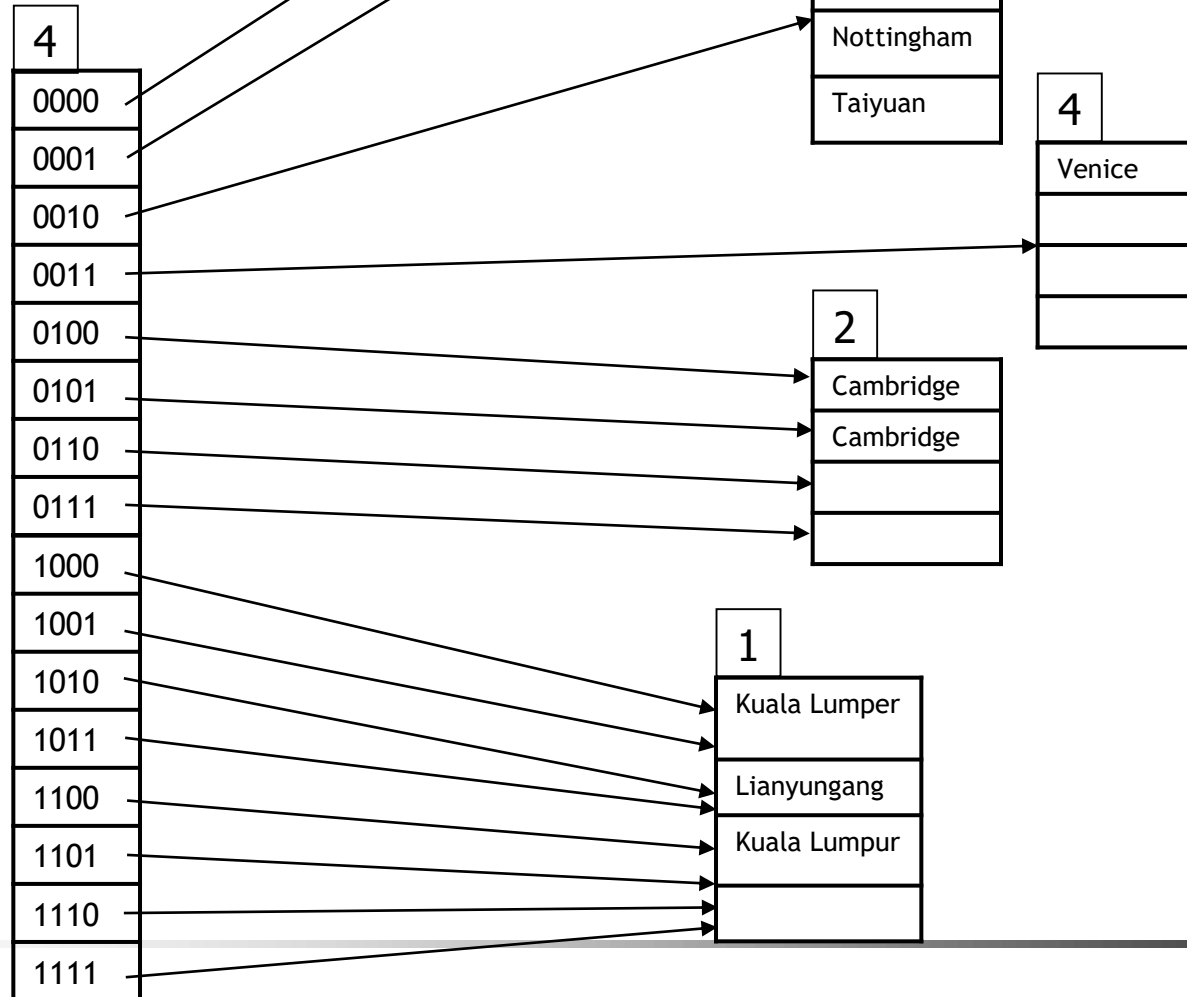
Q2 Solution cont'd

- Insert "Java Island"

    i = 4
    Hash Prefix: 2^4 = 16

Bucket Address Table

4
0000
0001
0010
0011
0100
0101
0110
0111
1000
1001
1010
1011
1100
1101
1110
1111

3

4

Taiyuan
Florence
Nottingham
Taiyuan

4

Venice

2

Cambridge
Cambridge
Java Island

1

Kuala Lumper
Lianyungang
Kuala Lumpur

Q2 Solution cont'd

- Insert "Valencia"

  i = 4
  Hash Prefix: $2^4 = 16$

Bucket Address Table

4
0000
0001
0010
0011
0100
0101
0110
0111
1000
1001
1010
1011
1100
1101
1110
1111

3

4
Taiyuan
Florence
Nottingham
Taiyuan

4
Venice

2
Cambridge
Cambridge
Java Island

1
Kuala Lumper
Lianyungang
Kuala Lumpur
Valencia

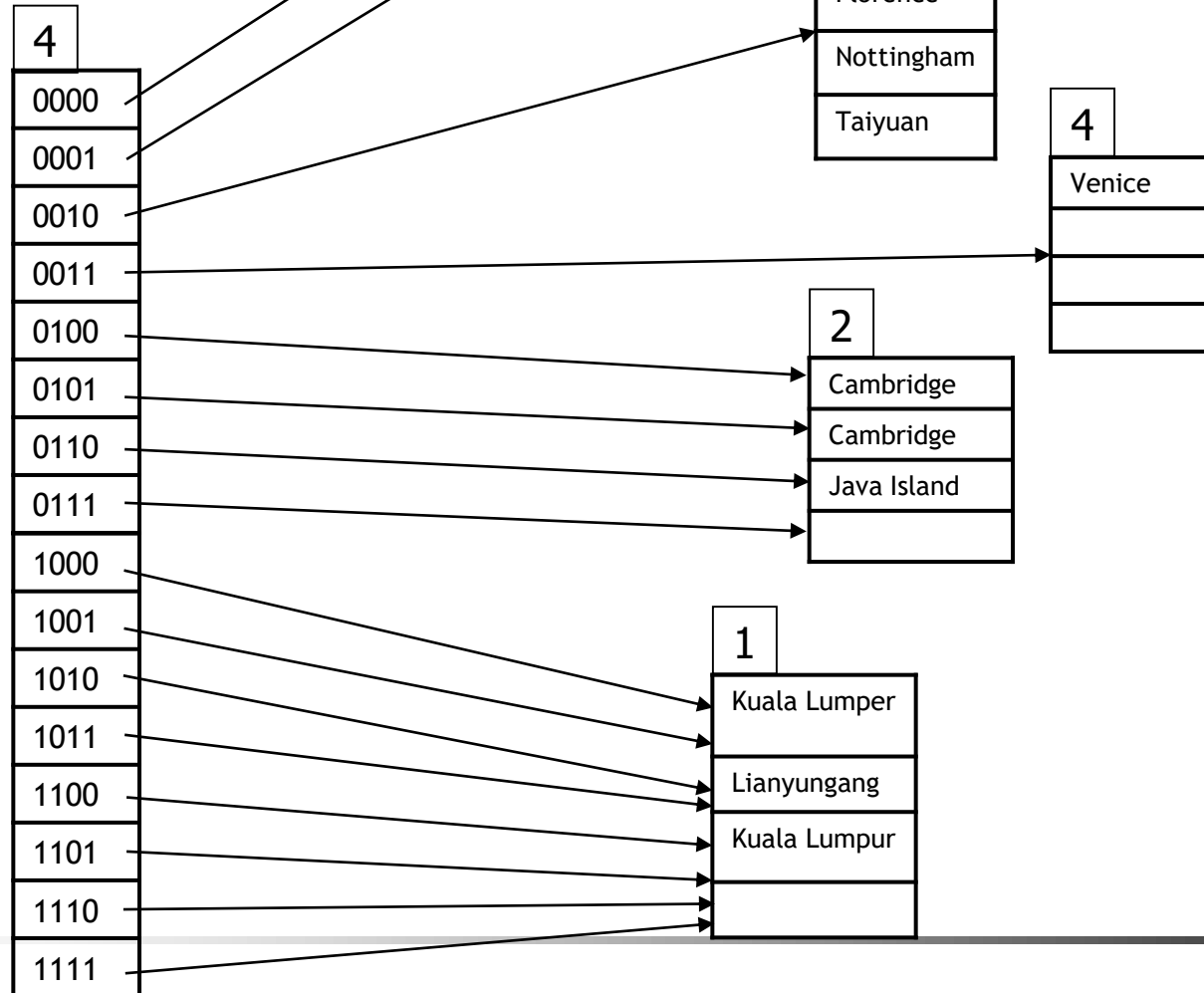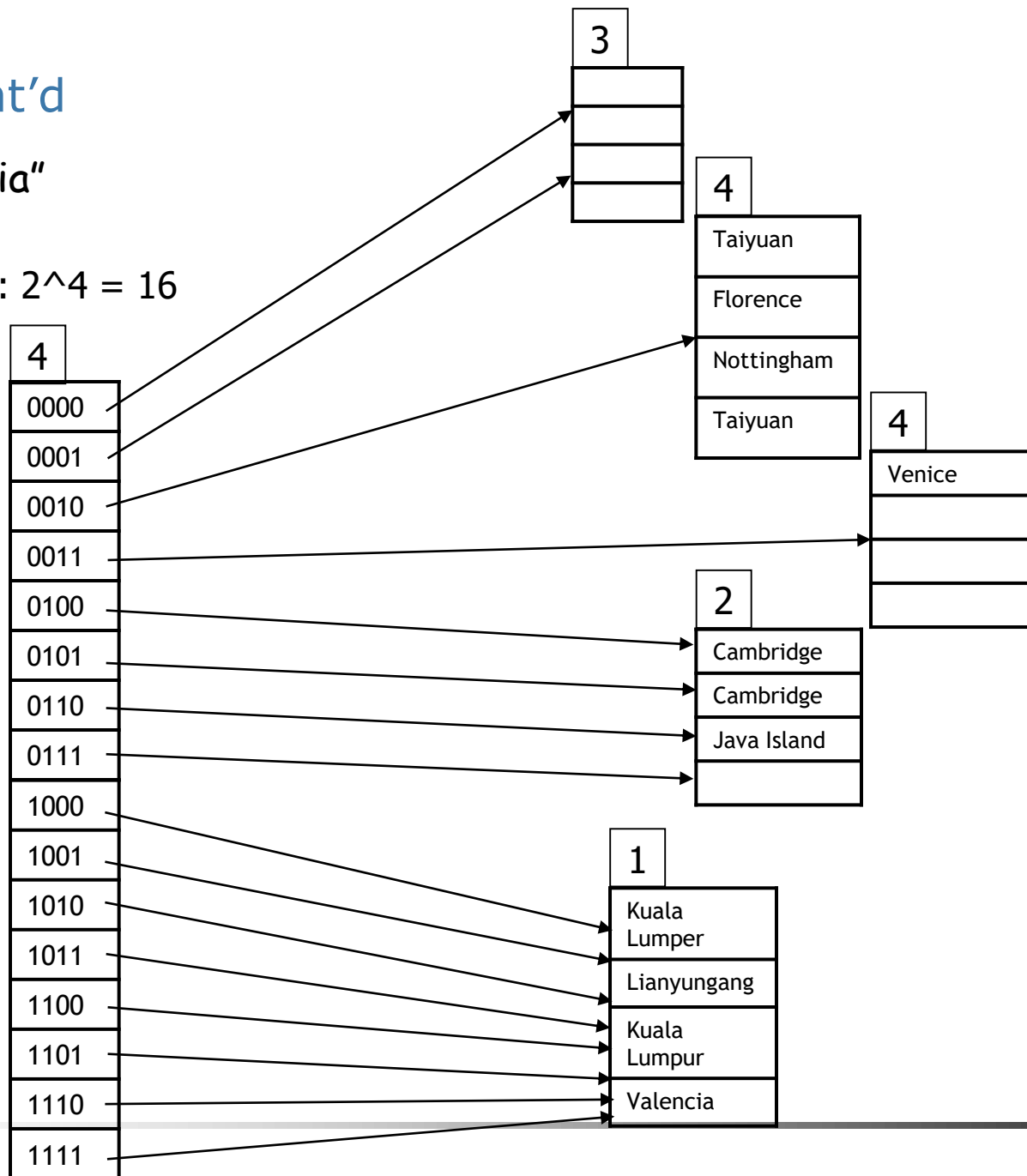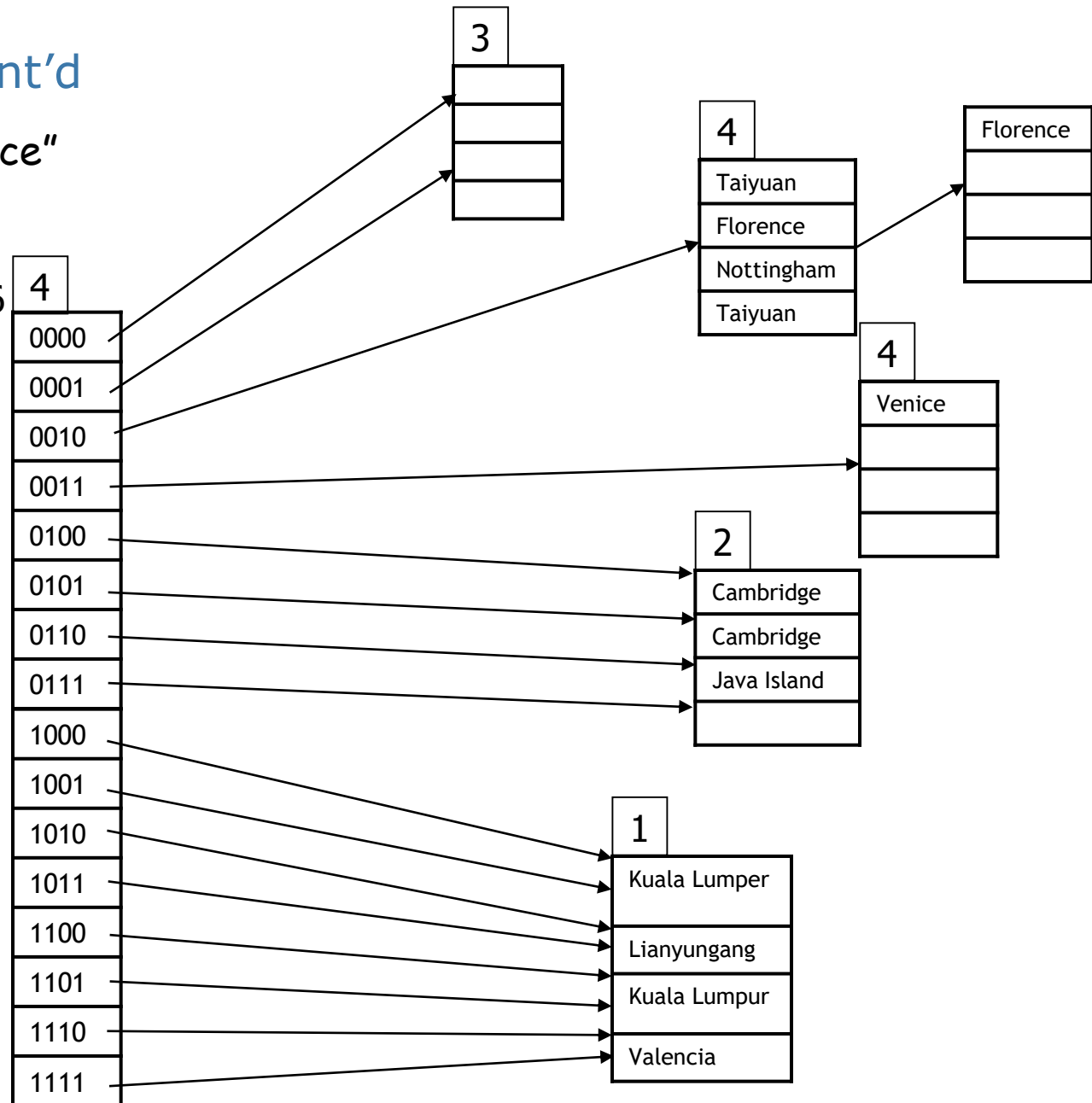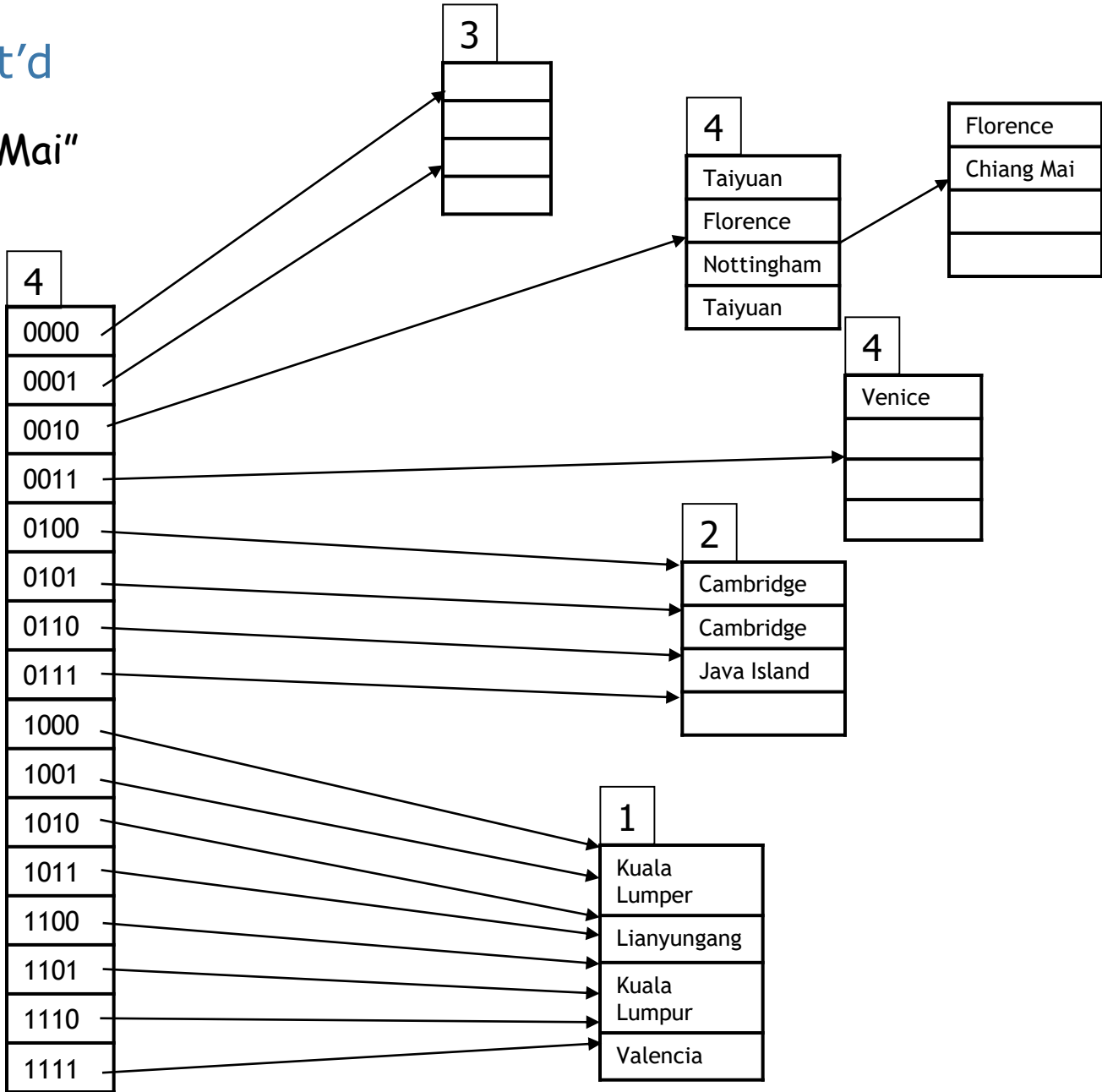# Q2 Solution cont'd

- Insert "Florence"

i = 4
Hash Prefix: $2^4 = 16$

Bucket Address Table

Q2 Solution cont'd

- Insert "Chiang Mai"

i = 4
Hash Prefix: 2^4 = 16

Bucket Address Table

4
0000
0001
0010
0011
0100
0101
0110
0111
1000
1001
1010
1011
1100
1101
1110
1111

3

4
| Taiyuan |
| Florence |
| Nottingham |
| Taiyuan |

| Florence |
| Chiang Mai |

4
| Venice |

2
| Cambridge |
| Cambridge |
| Java Island |

1
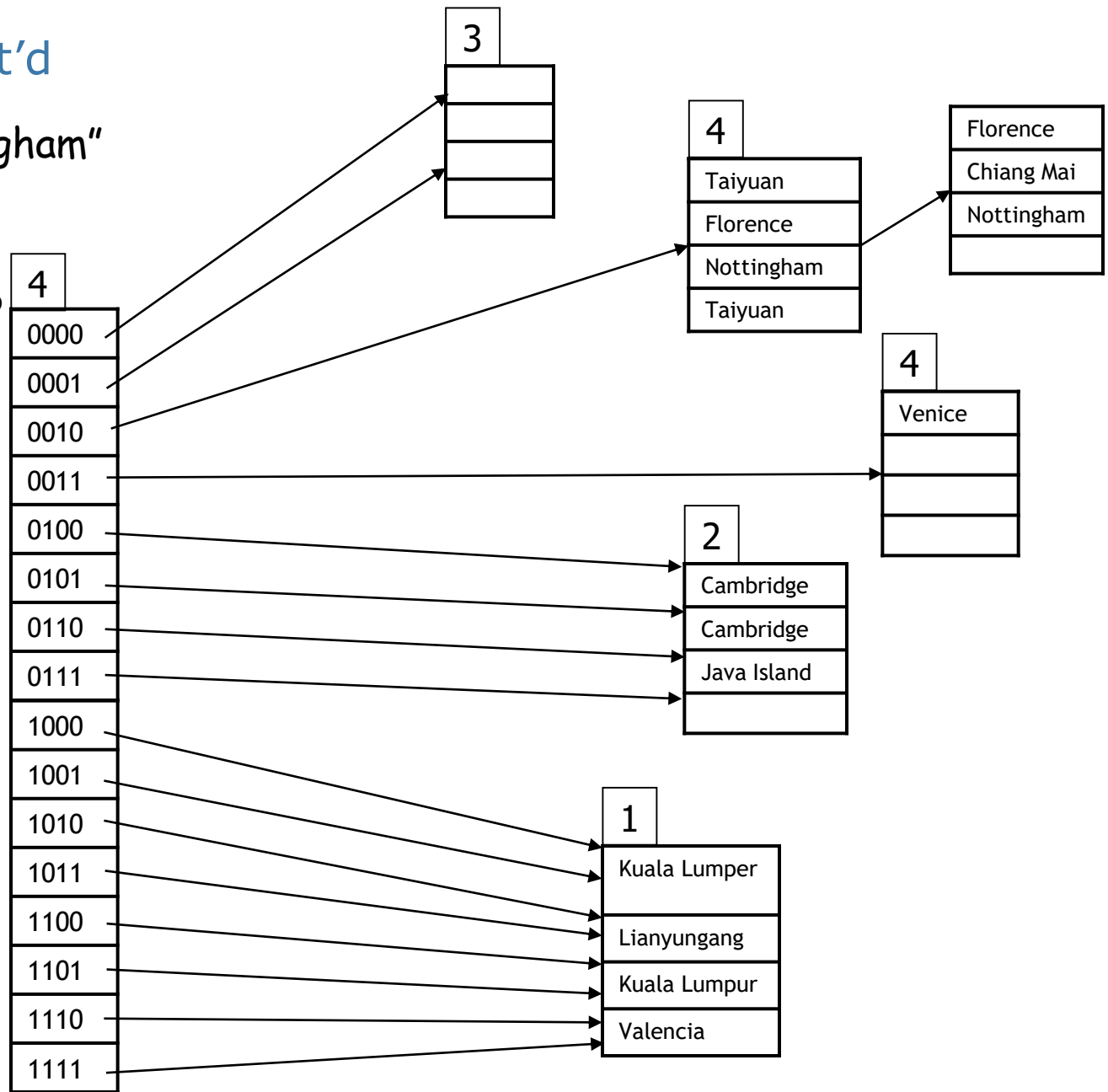| Kuala Lumper |
| Lianyungang |
| Kuala Lumpur |
| Valencia |

Q2 Solution cont'd

- Insert "Nottingham"

i = 4
Hash Prefix: 2^4 = 16

Bucket Address Table

Q3 Solutions

(1)Index on Year/Month: A B+ tree indexing is preferred for the year/month search, since B+ tree can be more efficient when is dealing with ordered retrievals. The leaf nodes of the B+ tree are linked sequentially so users can easily traverse all the records in chronological order of publication.

(2)Index on Author Names: A Extendable hash indexing is suitable for author names when searching by name. It is based on an exact match, and can use hash code to quickly locate a specific bucket and search for a specific name within it, also it can dynamically split its buckets to deal continuously increase data.

(3)Index on Abstract: B+ tree is more suited for range searches since it can find all the entries that match the given information after completing a lookup according to the linked list order of leaf nodes. Also, it reduces disk I/O operations, more suitable for databases that contain a lot of text data.