CS525-01/02/03 - Fall 2018
Problem Set 2 KEY

Chapter 4: Index Structures

- Exercise 4.1.1
  - The data file: we will need $\frac{n}{3}$ blocks
  - Dense index:

    Should contain one index record per data record, a total of n index records.
    Since we can fit 10 of those in a block, we need a total of $\frac{n}{10}$ blocks
    $\Rightarrow$ The total number of blocks is $\frac{n}{10} + \frac{n}{3} = \frac{13n}{30}$
  - Sparse index

    Should contain one index record per data block, a total of $\frac{n}{3}$ index records.
    Since we can fit 10 of those in a block, we need a total of $\frac{\frac{n}{3}}{10} = \frac{n}{30}$ blocks
    $\Rightarrow$ The total number of blocks is $\frac{n}{30} + \frac{n}{3} = \frac{11n}{30}$

- Exercise 4.3.3

  We need first to find the largest integer value of n such that $12n+4(n+1) \leq 16384$. $\Rightarrow$ $n = 1023$

  $\therefore$ 1023 keys and 1024 pointers.

- Exercise 4.3.4

  B+-tree Non-leaf: minimum #: $\lceil \frac{n+1}{2} \rceil$ pointers (to nodes), Leaf: $\lfloor \frac{n+1}{2} \rfloor$ pointers (to data)

  a. 5 keys and 6 pointers for the interior nodes, 5 keys and 5 pointers in the leaf nodes

  b. 5 keys and 6 pointers for the interior nodes, 6 keys and 6 pointers in the leaf nodes

- Exercise 4.3.5

  a. Start at the root. $41 \geq 13$ so follow the second pointer. $31 \leq 41 < 43$ so follow the third pointer. We find one of the keys is 41 so we follow the third pointer to the data block.

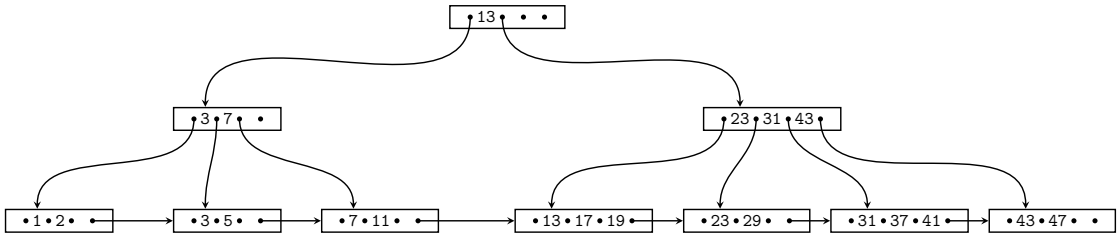
  b. Start at the root. $40 \geq 13$ so follow the second pointer. $31 \leq 41 < 43$ so follow the third pointer. We do not find any of the keys is 40, so we find that there are no records with key 40.

  c. Start at the root. $20 \geq 13$ so follow the second pointer. $20 < 23$ so follow the first pointer. No keys in the range are found so follow the next leaf pointer. 23 is in the range, so follow the first pointer to the data block. Key 29 is in the range so follow second pointer to the data block. No more keys on the leaf so follow next leaf pointer. Key 31 is not in the range so we are done.

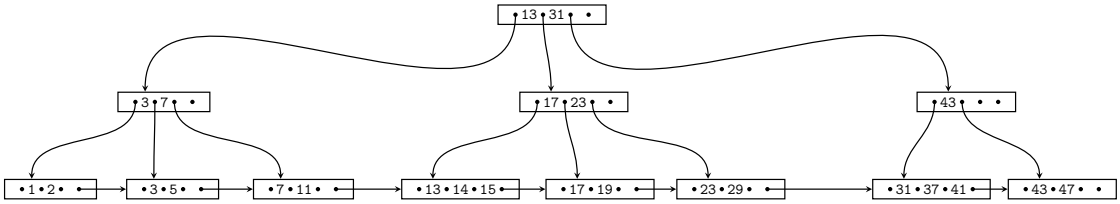
  d. Start at the root. Follow leftmost pointers until the leaf node. For each of the keys, if it less than 30, follow the pointer to the data block. Thus we will follow data pointers for keys 2, 3, 5. Since $30 \geq 5$ follow the next leaf pointer and repeat. Thus we will follow the data pointers for keys 7, 11, then next leaf, 13, 17, 19, next, 23, 29, next. $30 < 31$ so we are done.

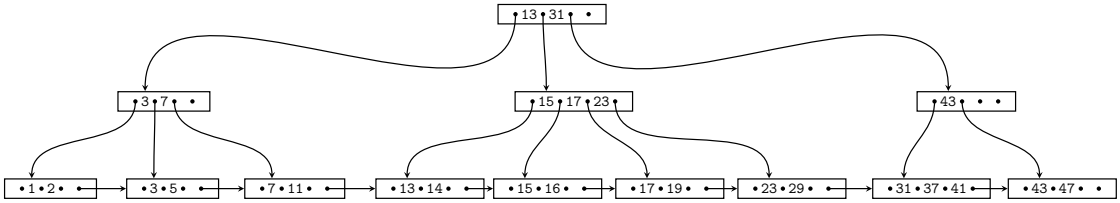
e. Start at the root. 30 $\geq$ 13 so follow the second pointer. 23$\leq$30 $<$ 31 so follow the second pointer. We find none of the keys is greater than 30 so we follow the next leaf pointer. We find 31, which is the first key greater than 30 so we follow every pointer to data starting from 31 to the end of all keys in the leaves (i.e. we will follow data pointers for keys: 31,37,41,43,47).

f. insert(1)

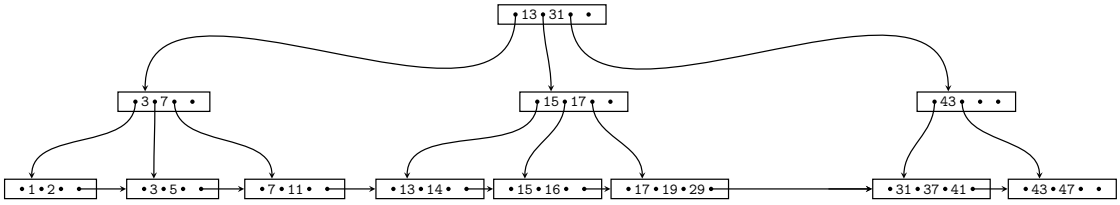


g. insert(14 through 16)

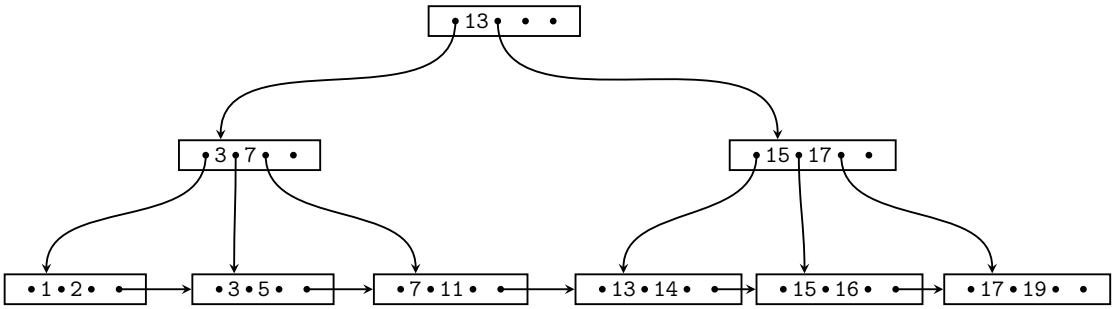




h. delete(23)

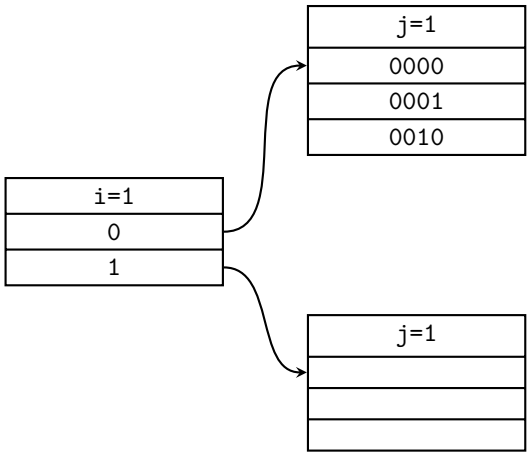


i. `delete(all >23)`

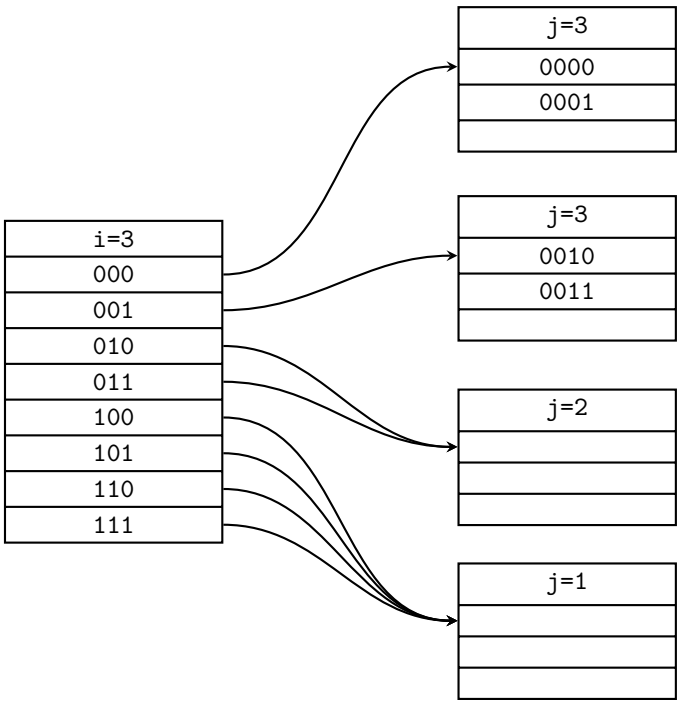


- Exercise 4.4.6:

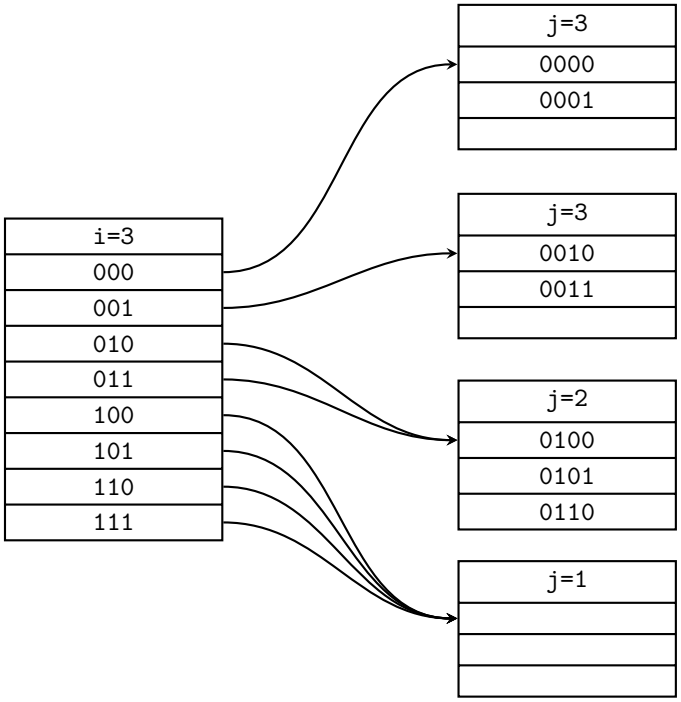
  a. `Extensible hashing`
  insert(0000), (0001), (0010)



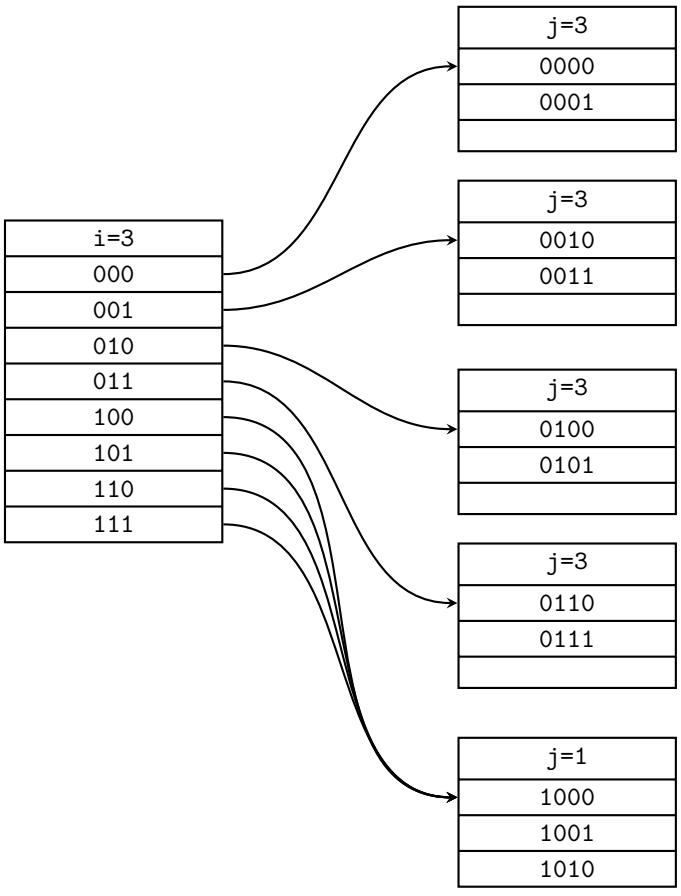
  insert(0011)



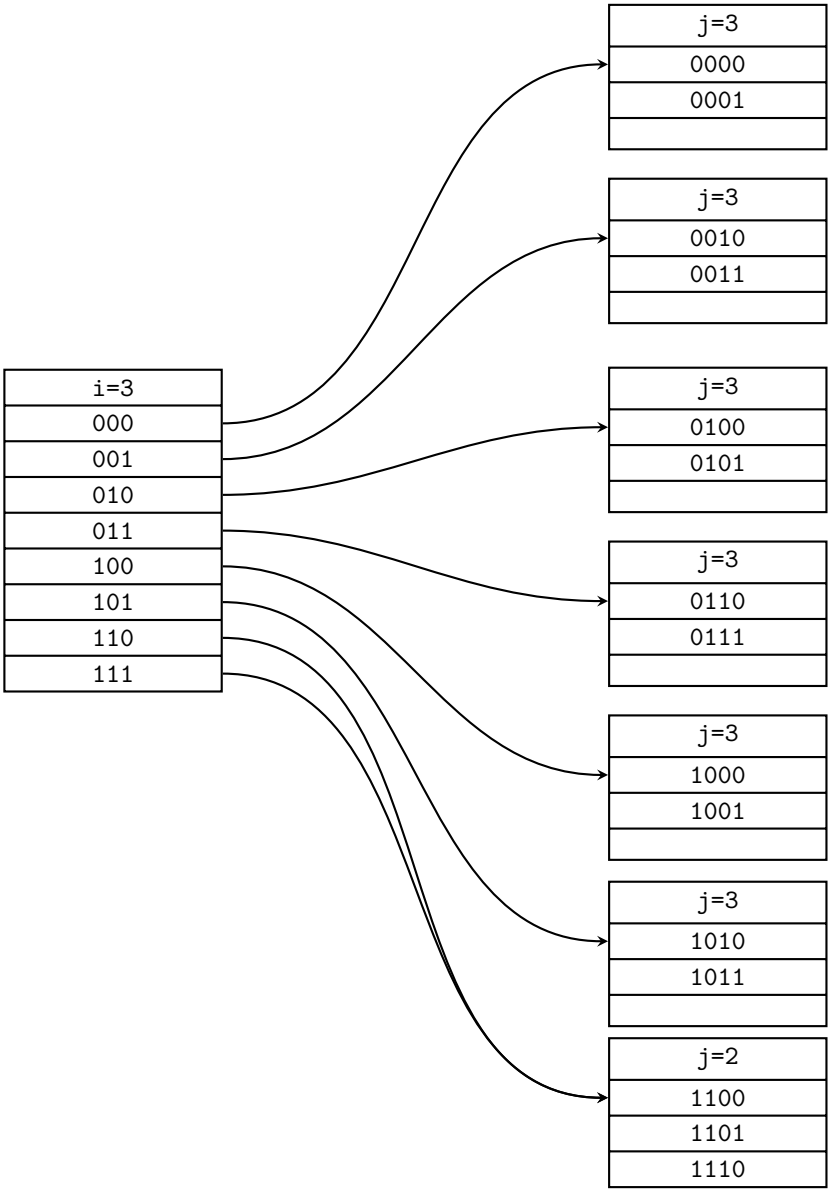
insert(0100), (0101), (0110)



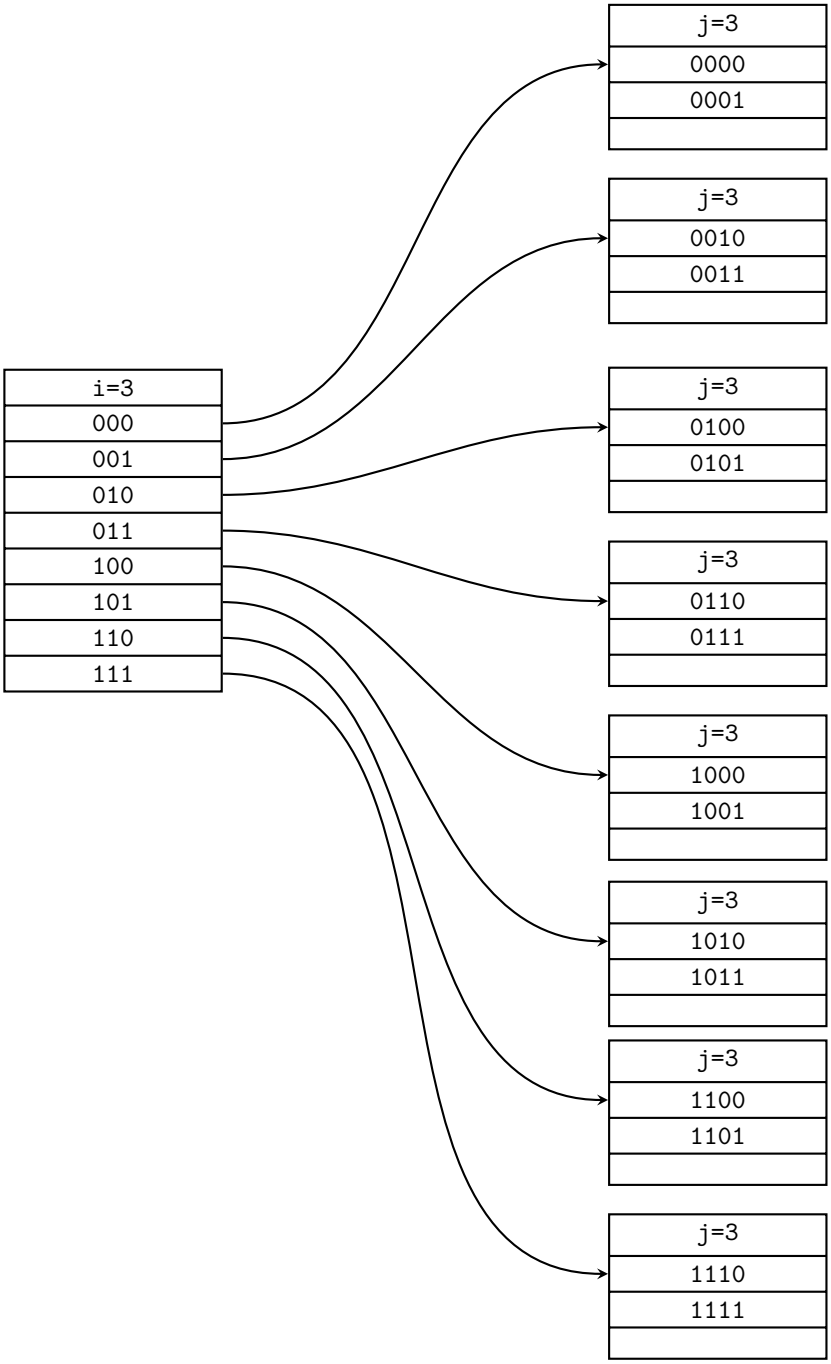
insert(0111), then (1000), (1001), (1010)



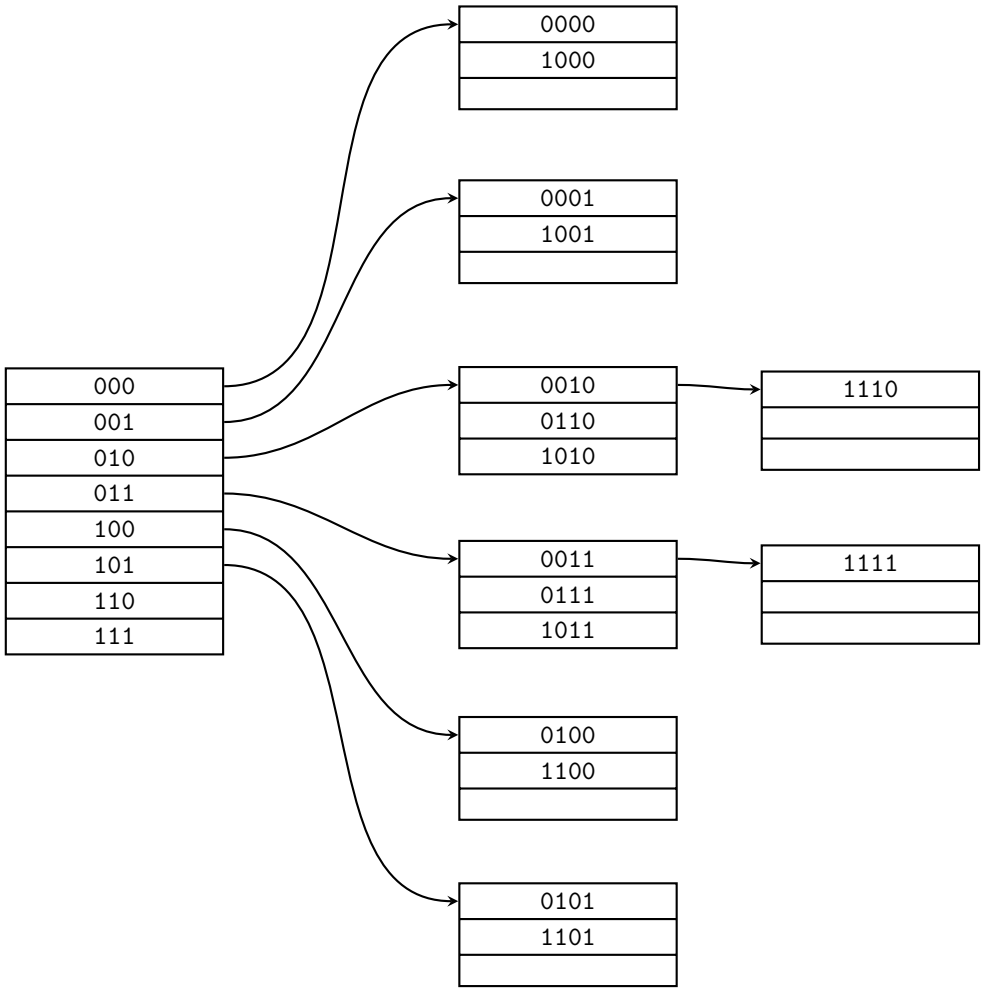
insert(1011), then (1100), (1101), (1110)



insert (1111);

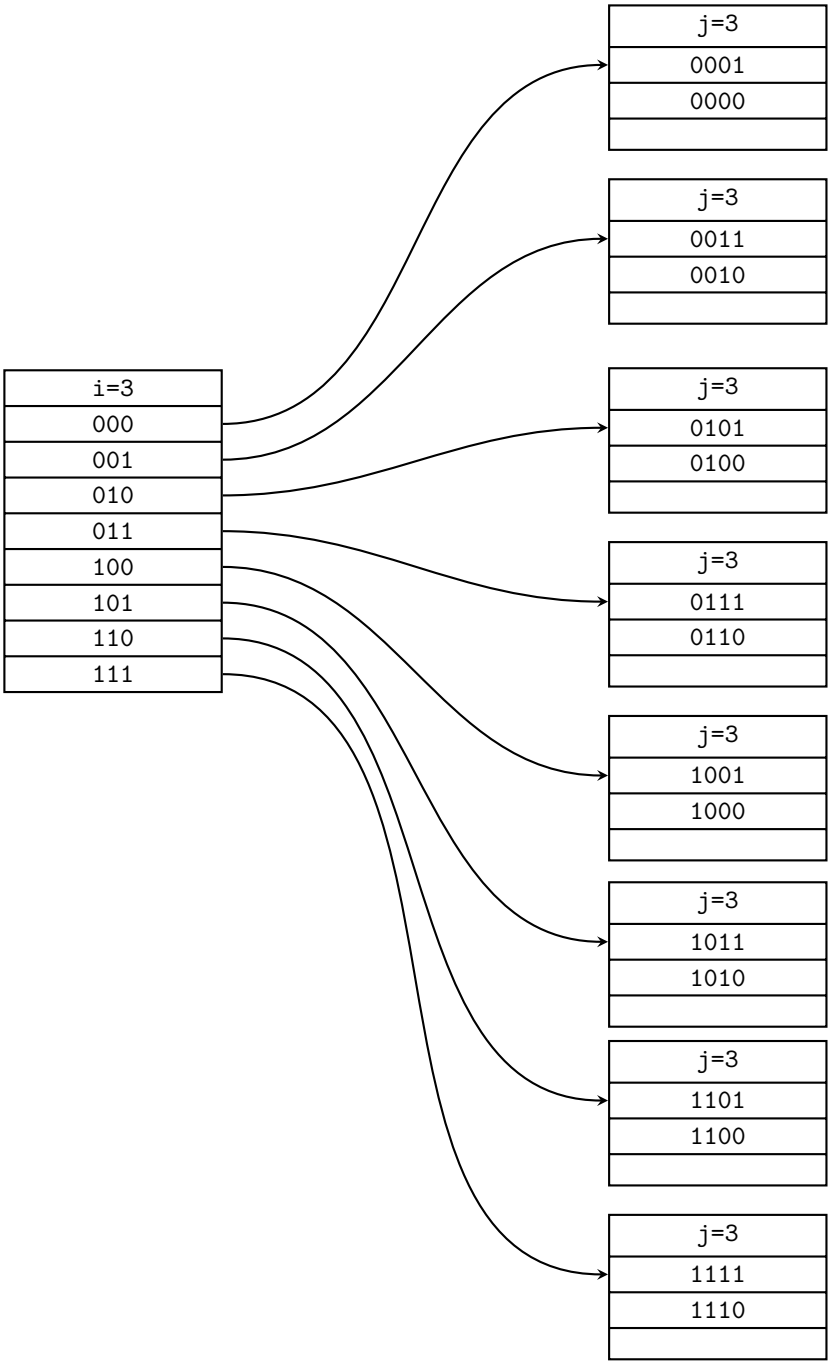

i=3
000
001
010
011
100
101
110
111
j=3
0000
0001
j=3
0010
0011
j=3
0100
0101
j=3
0110
0111
j=3
1000
1001
j=3
1010
1011
j=3
1100
1101
j=3
1110
1111

b. linear hashing with a capacity threshold of 100%.
`i = 3, n = 6, r = 16`
final result;



c. Extensible hashing

final result



d. linear hashing with a capacity threshold of 75%.
`i = 3, n = 8, r = 16`
final result;