

## Linear Hashing

For this example, we are going to use the same data file as the Extendible hashing example. It looks like the following:

Record	K	H(K)	Bits of H(K)
Record 1	2639	1	00001
Record 2	3760	16	10000
Record 3	4692	20	10100
Record 4	4871	7	00111
Record 5	5659	27	11011
Record 6	1821	29	11101
Record 7	1074	18	10010
Record 8	2115	11	01011
Record 9	1620	20	10100

In this case, we start with our current level = 1 bit, our  $i+1$  level is = 2 bits of the algorithm above. That looks like:

Next = 0

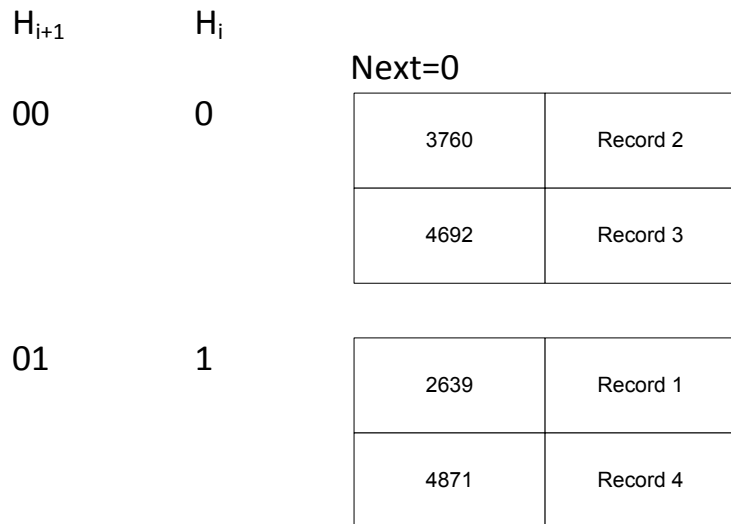
$H_{i+1}$	$H_i$					
00	0	<table border="1"><tr><td></td><td></td></tr><tr><td></td><td></td></tr></table>				

01	1	<table border="1"><tr><td></td><td></td></tr><tr><td></td><td></td></tr></table>				

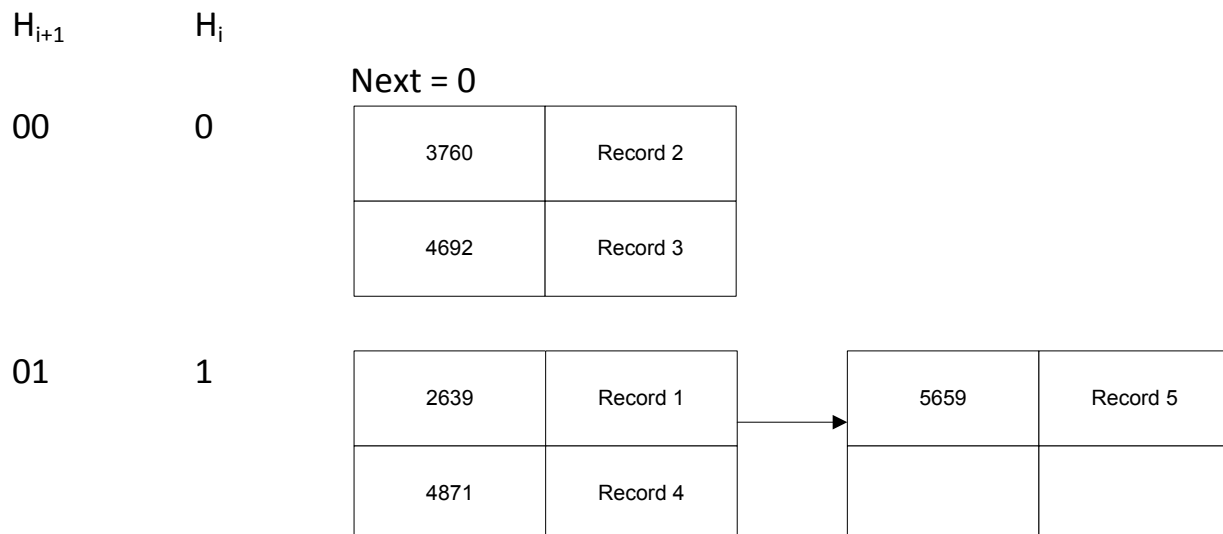
We are going to process a split every time we create an overflow bucket. We hash Record 1's key using  $H_i$  and get a value of 1. Comparing that to the Next value, we see it is not less than Next, so we use the  $H_i$  value to place the data entry. That gives us:

$H_{i+1}$	$H_i$	Next=0				
00	0	<table border="1"><tr><td></td><td></td></tr><tr><td></td><td></td></tr></table>				
01	1	<table border="1"><tr><td>2639</td><td>Record 1</td></tr><tr><td></td><td></td></tr></table>	2639	Record 1		
2639	Record 1					

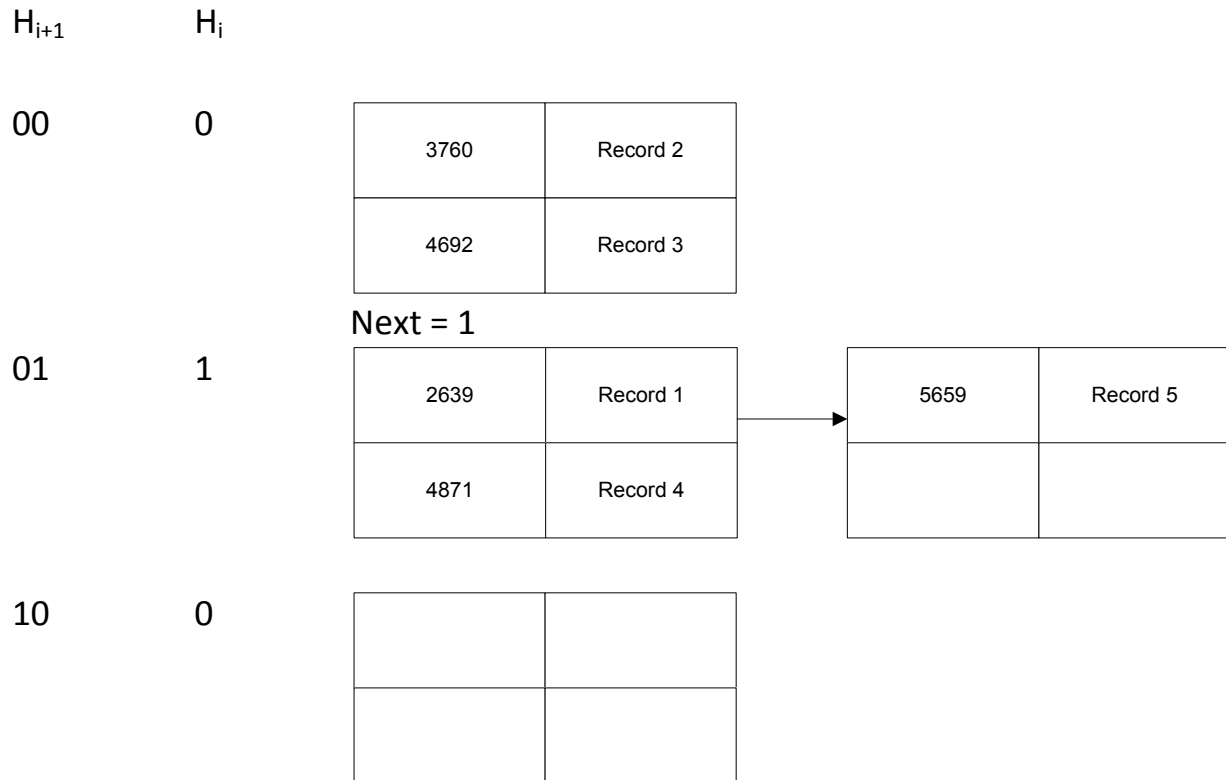
Using the same formula, after inserting the first four records, we look like this:



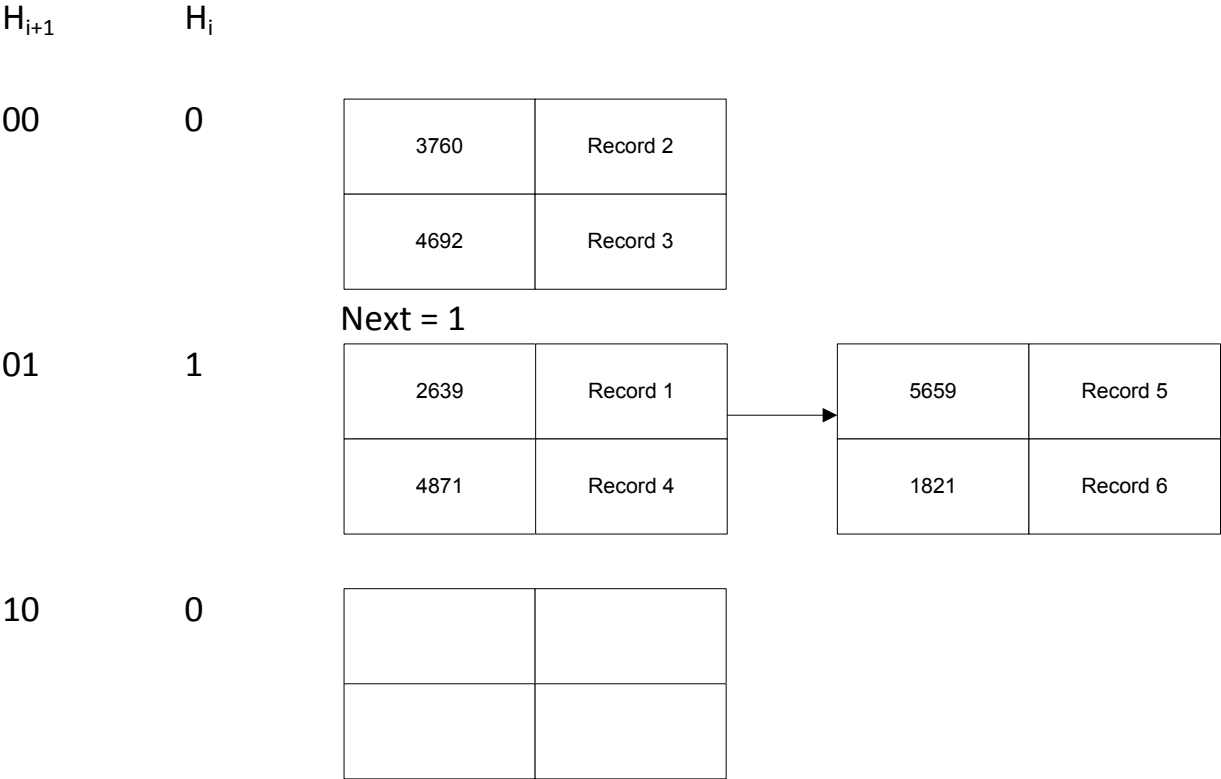
Inserting Record 5 causes an overflow page to be created:



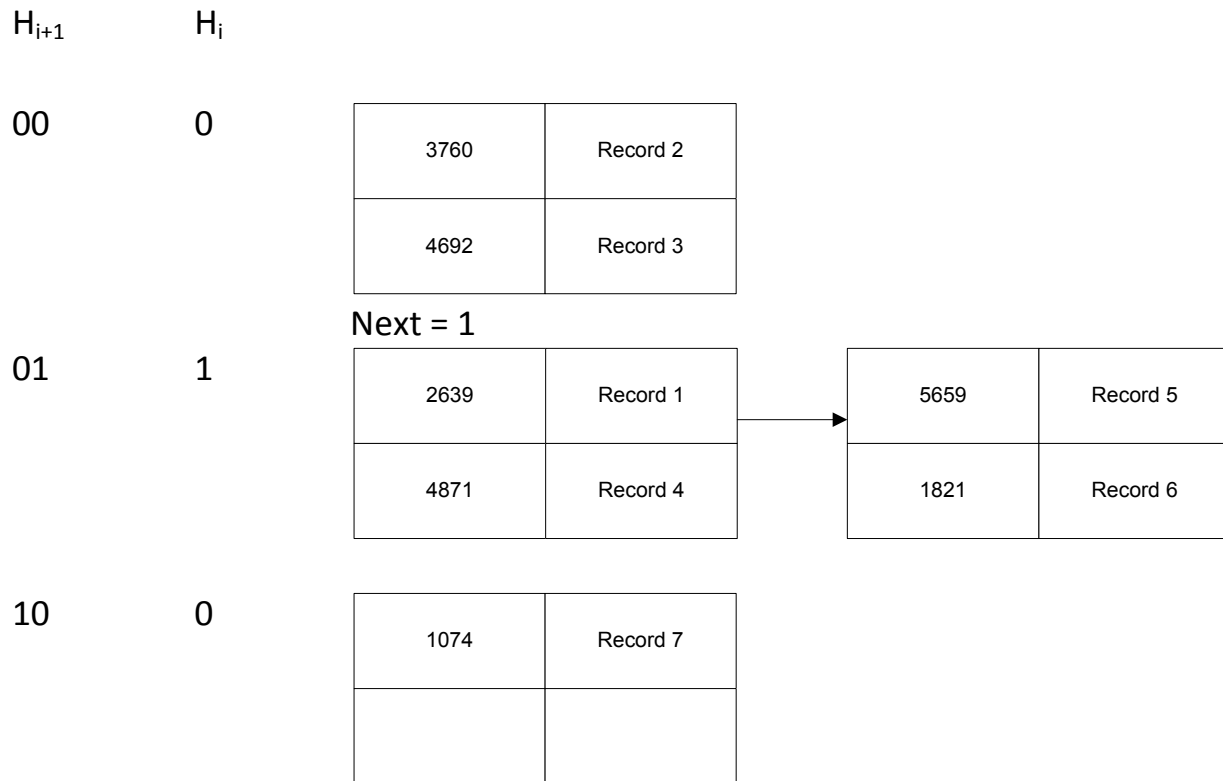
This is our trigger to split the bucket pointed to by Next. We split the bucket, and rehash the entries using  $H_{i+1}$ .



We hash Record 6 using  $H_i$ , and it returns a 1. Comparing that value to the Next pointer (i.e. if hash value < Next, use  $H_{i+1}$  rather than  $H_i$ ), we see that value is still using the  $H_i$  function, and we go to place it in the appropriate hash bucket. There is room in the overflow bucket for this value, giving us the following:



When we look at Record 7, we see that  $H_i = 0$ . Since this value is less than Next, we know we must use  $H_{i+1}$ . That hash value is 10, and we place Record 7 appropriately.



When we hash Record 8 with  $H_i$ , we get 1, which tells us to use  $H_i$ . When we look at the buckets for entry 1, we see this is going to create another overflow page. Since Next is pointing to this entry, we might as well split the bucket first before placing the record. In addition, this is the last entry in the original has table, so we need to reset Next,  $H_i$ , and  $H_{i+1}$  to handle the new level. First we split the bucket:

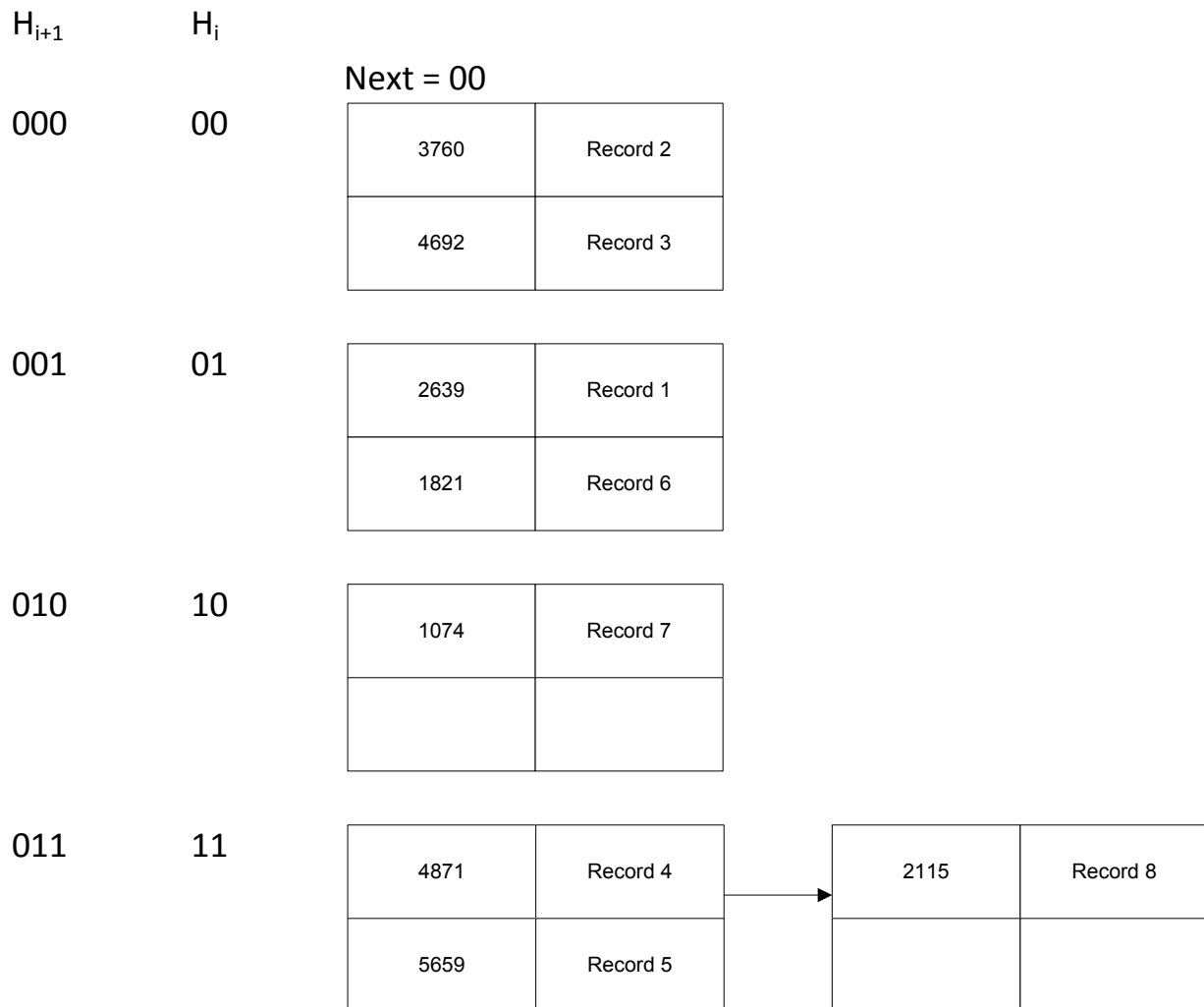
$H_{i+1}$	$H_i$					
00	0	<table border="1"> <tr> <td>3760</td> <td>Record 2</td> </tr> <tr> <td>4692</td> <td>Record 3</td> </tr> </table>	3760	Record 2	4692	Record 3
3760	Record 2					
4692	Record 3					
		Next = 1				
01	1	<table border="1"> <tr> <td>2639</td> <td>Record 1</td> </tr> <tr> <td>1821</td> <td>Record 6</td> </tr> </table>	2639	Record 1	1821	Record 6
2639	Record 1					
1821	Record 6					
10	0	<table border="1"> <tr> <td>1074</td> <td>Record 7</td> </tr> <tr> <td></td> <td></td> </tr> </table>	1074	Record 7		
1074	Record 7					
11	1	<table border="1"> <tr> <td>4871</td> <td>Record 4</td> </tr> <tr> <td>5659</td> <td>Record 5</td> </tr> </table>	4871	Record 4	5659	Record 5
4871	Record 4					
5659	Record 5					

And now we update Next and the hash algorithms to reflect hitting the bottom of the table.

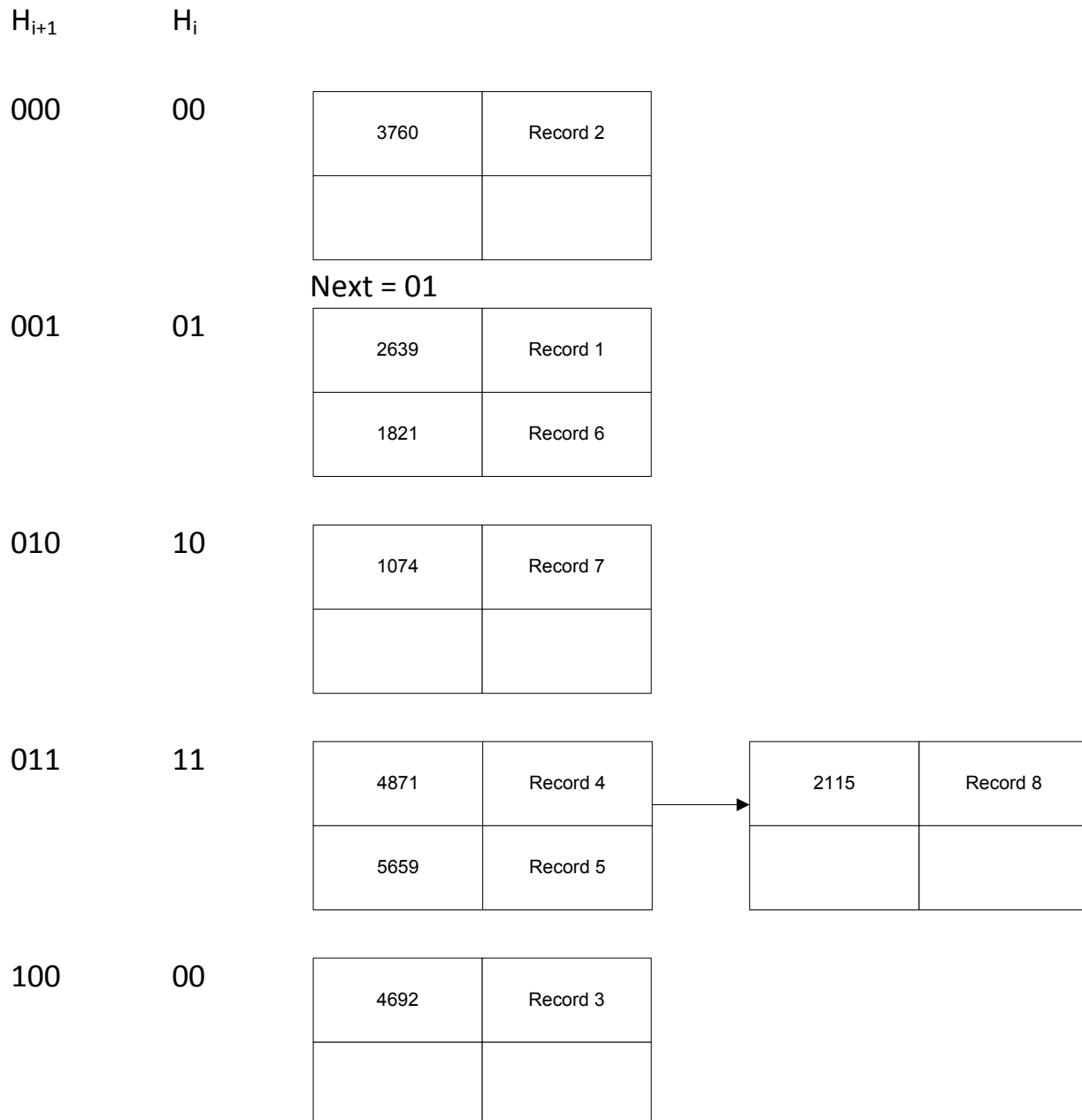
$H_{i+1}$	$H_i$					
		Next = 00				
000	00	<table border="1"><tr><td>3760</td><td>Record 2</td></tr><tr><td>4692</td><td>Record 3</td></tr></table>	3760	Record 2	4692	Record 3
3760	Record 2					
4692	Record 3					
001	01	<table border="1"><tr><td>2639</td><td>Record 1</td></tr><tr><td>1821</td><td>Record 6</td></tr></table>	2639	Record 1	1821	Record 6
2639	Record 1					
1821	Record 6					
010	10	<table border="1"><tr><td>1074</td><td>Record 7</td></tr><tr><td></td><td></td></tr></table>	1074	Record 7		
1074	Record 7					
011	11	<table border="1"><tr><td>4871</td><td>Record 4</td></tr><tr><td>5659</td><td>Record 5</td></tr></table>	4871	Record 4	5659	Record 5
4871	Record 4					
5659	Record 5					



We've finished our doubling activity, now it's back to placing Record 8. We hash it to  $H_i$ , which is 11 (note we are now looking at 2 bits, not 1). Next is 00, which tells us to use  $H_{i+1}$ . Again we find this bucket is full, which means we would add an overflow page. First we add Record 8 to an overflow page:



Then we process the next split (note, since we have already processed an split this insert, we could just move to the next insert here – the decision to process another split is strictly arbitrary) :



Using  $H_i$ , Record 9 hashes to 00. Since this value is less than Next (01), we know we must use  $H_{i+1}$ . We rehash to 100, and place Record 9 in the appropriate place.

$H_{i+1}$	$H_i$										
000	00	<table border="1"> <tr> <td>3760</td> <td>Record 2</td> </tr> <tr> <td></td> <td></td> </tr> </table>	3760	Record 2							
3760	Record 2										
		Next = 01									
001	01	<table border="1"> <tr> <td>2639</td> <td>Record 1</td> </tr> <tr> <td>1821</td> <td>Record 6</td> </tr> </table>	2639	Record 1	1821	Record 6					
2639	Record 1										
1821	Record 6										
010	10	<table border="1"> <tr> <td>1074</td> <td>Record 7</td> </tr> <tr> <td></td> <td></td> </tr> </table>	1074	Record 7							
1074	Record 7										
011	11	<table border="1"> <tr> <td>4871</td> <td>Record 4</td> <td rowspan="2">→</td> <td>2115</td> <td>Record 8</td> </tr> <tr> <td>5659</td> <td>Record 5</td> <td></td> <td></td> </tr> </table>	4871	Record 4	→	2115	Record 8	5659	Record 5		
4871	Record 4	→	2115	Record 8							
5659	Record 5										
100	00	<table border="1"> <tr> <td>4692</td> <td>Record 3</td> </tr> <tr> <td>1620</td> <td>Record 9</td> </tr> </table>	4692	Record 3	1620	Record 9					
4692	Record 3										
1620	Record 9										