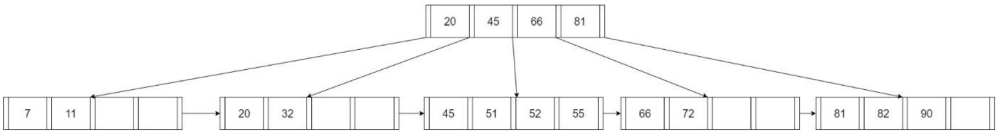
CS-353
HW-6

Ömer Oktay Gültekin (1)
2190143
Section 2

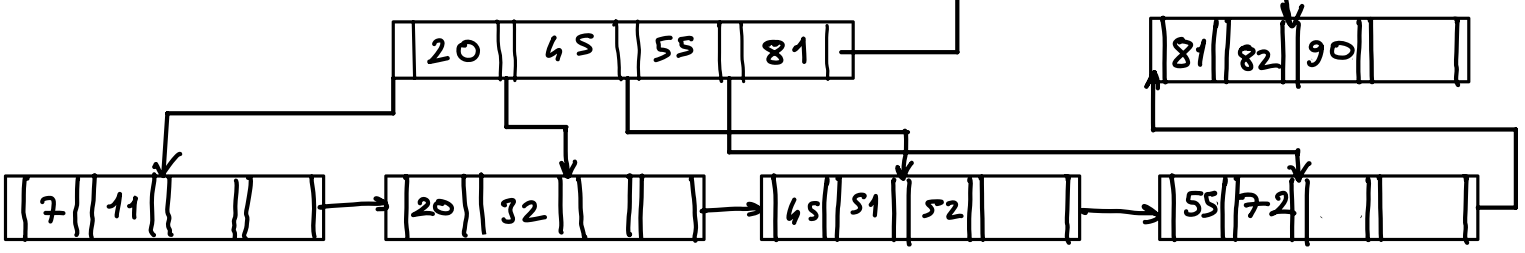
**Q.1 [48 pts, 12 pts each]** Given the following B+ tree. The maximum number of pointers for each node of the tree is 5.



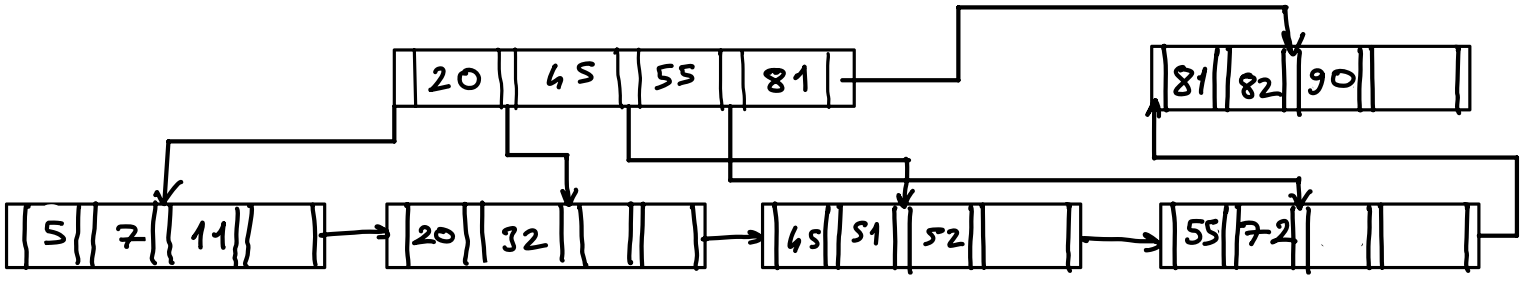

**For the following insertion/deletion operations you should use the algorithms discussed in the class.**

**(a)** Draw the resulting tree after deleting the entry with search key value $k = 66$ and then inserting entries with search key values $k = 5$, 9 and 12 successively.

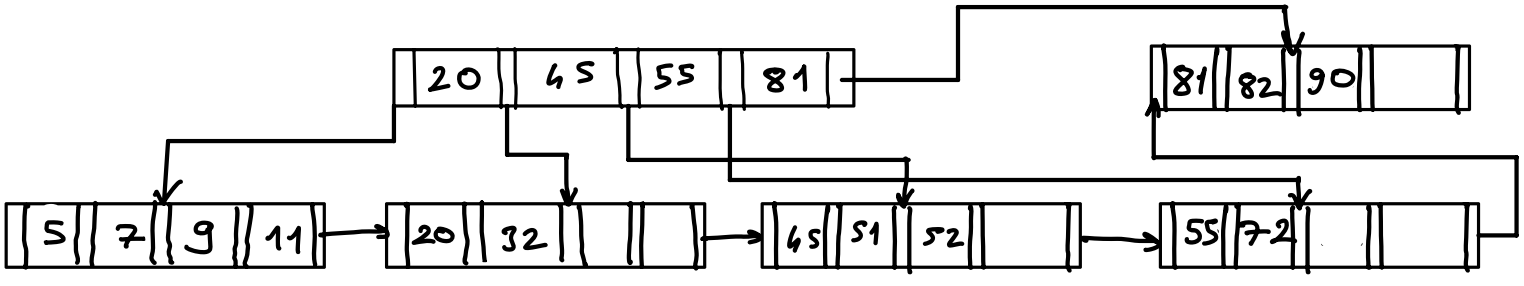
Delete k=66



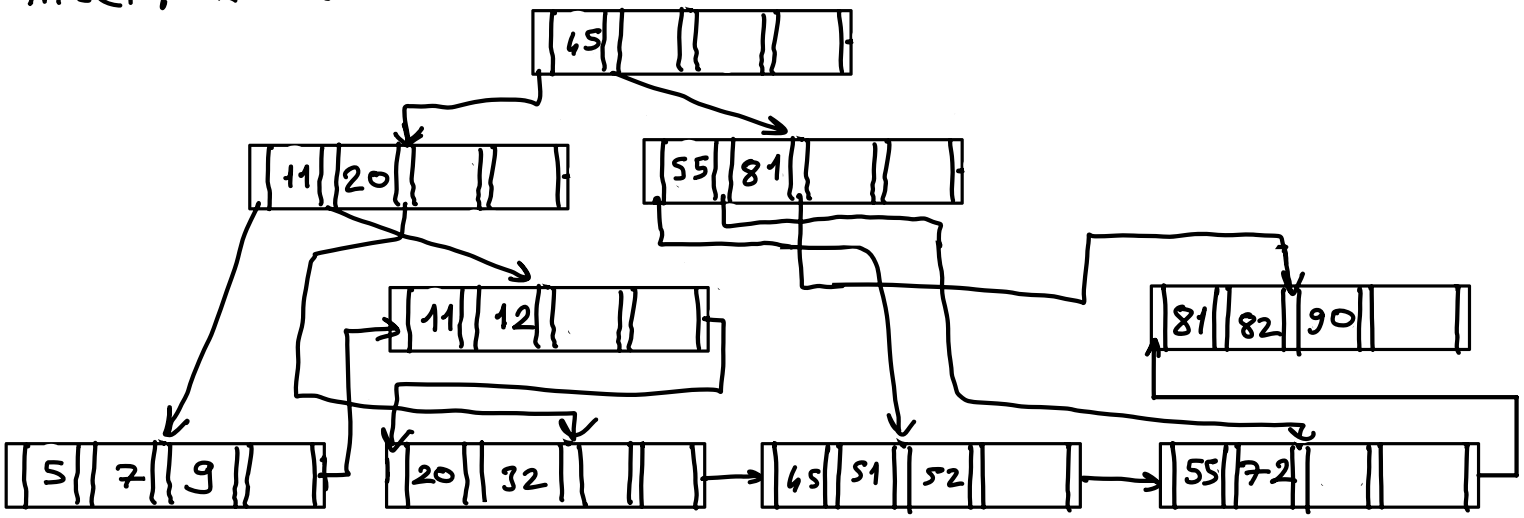

Insert k=5




Insert k=9

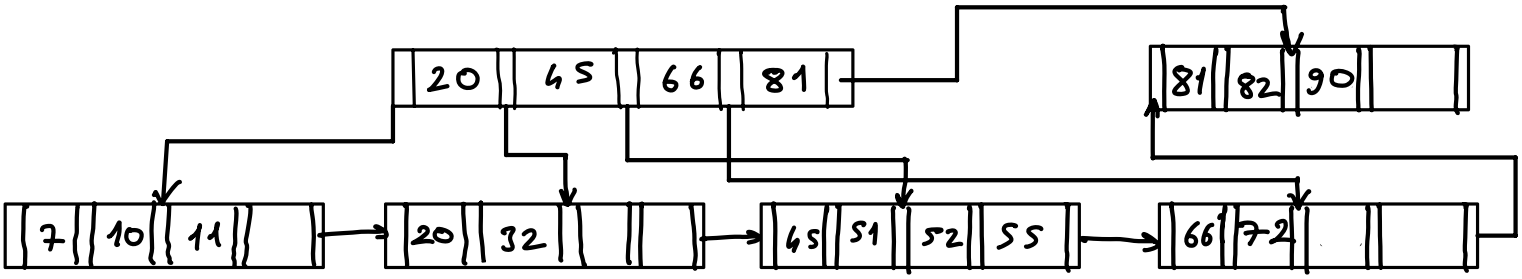


Insert k=12

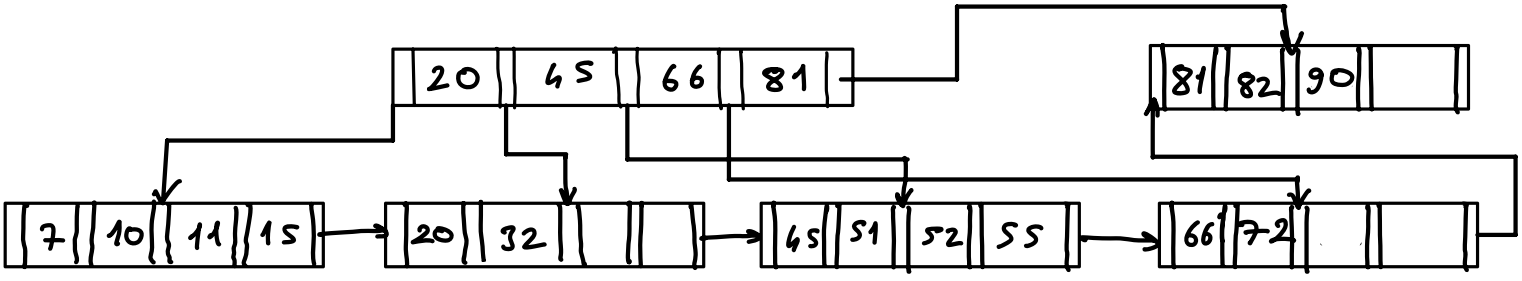



**(b)** Draw the resulting tree after inserting entries with search key values $k = 10$ and 15, and then deleting the entry with search key value $k = 20$ **from the original tree**.

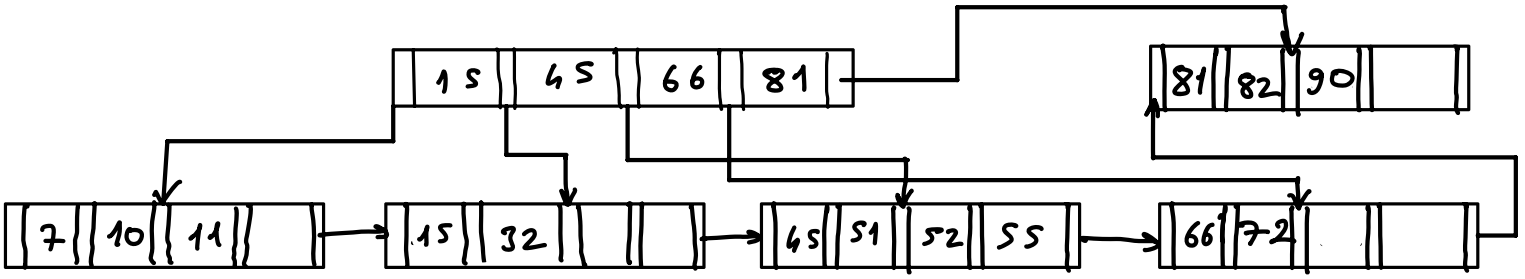
Insert k=10




Insert k=15

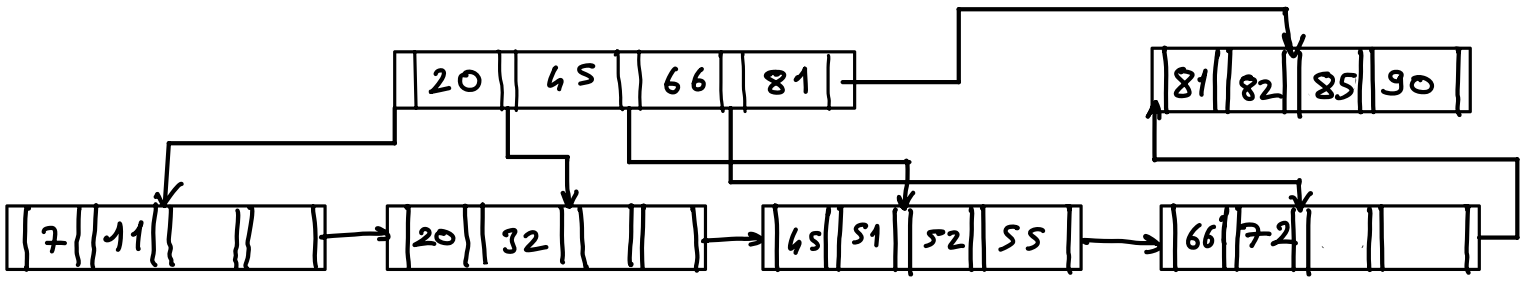



Delete k=20

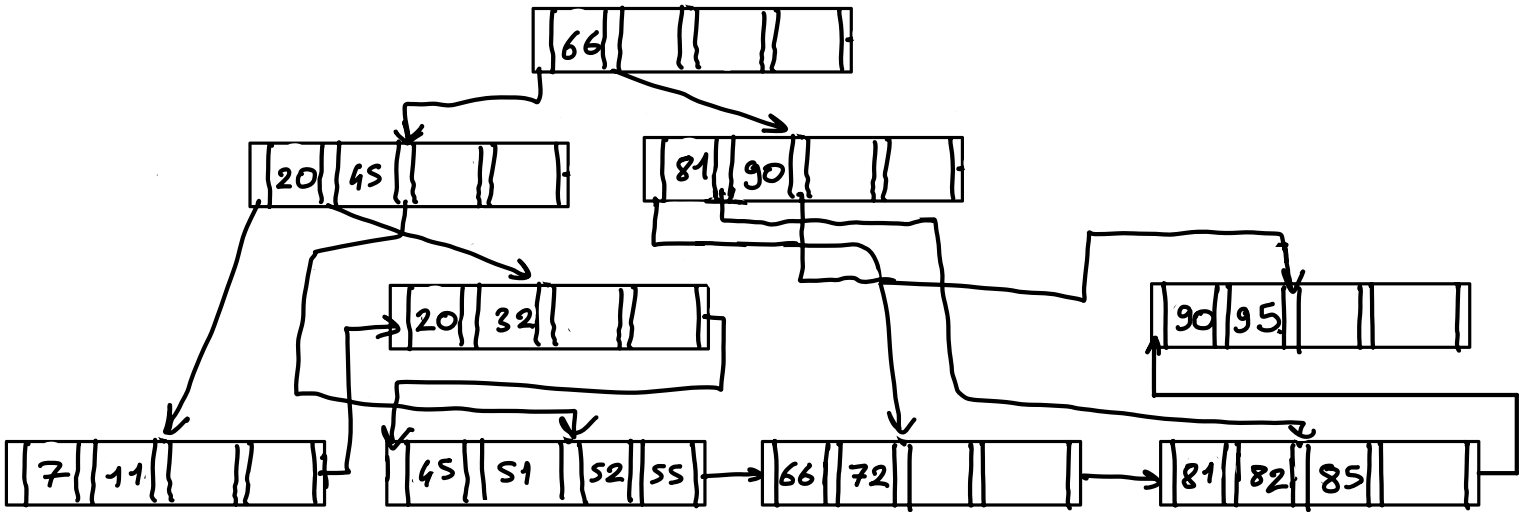


**(c)** Draw the resulting tree after inserting entries with search key values $k = 85$ and 95 **into the original tree**.

Insert k = 85



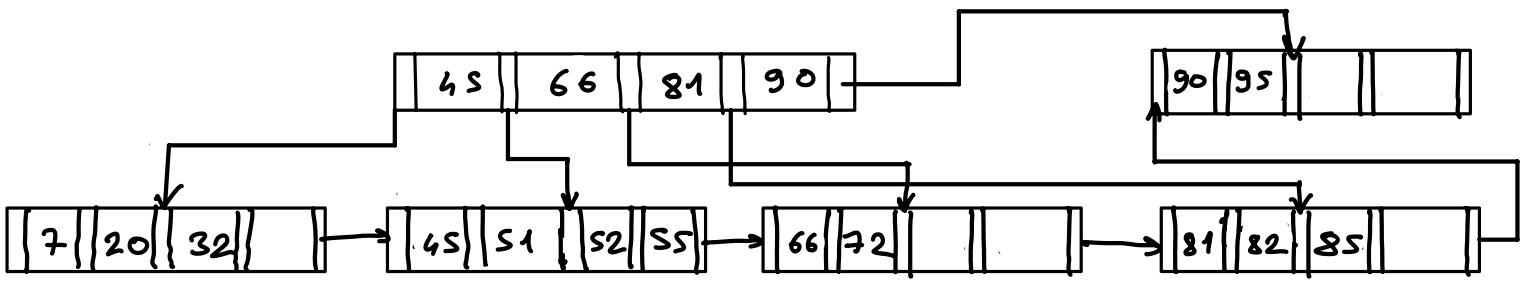

Insert k = 95




**original tree.**

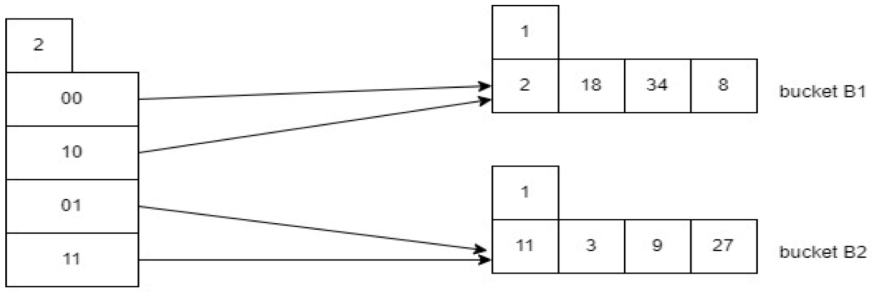
**(d)** Draw the resulting tree after deleting the entry with search key value $k = 11$ **from the resulting tree in part (c)**.

Delete k = 11 (causing merge because of too few args)



**Q.2 [52 pts, 26 pts each]**

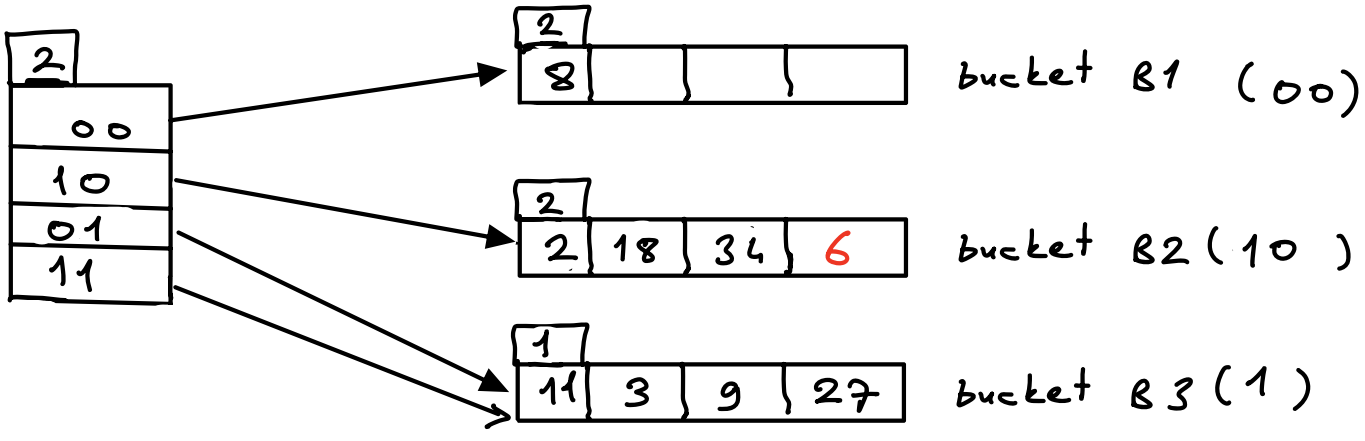
Given an extendable hash index with the following contents. Each bucket can contain at most 4 key values and the hash value of a search key x is the binary value of x. The least significant bit of a hash value is used during insertion.



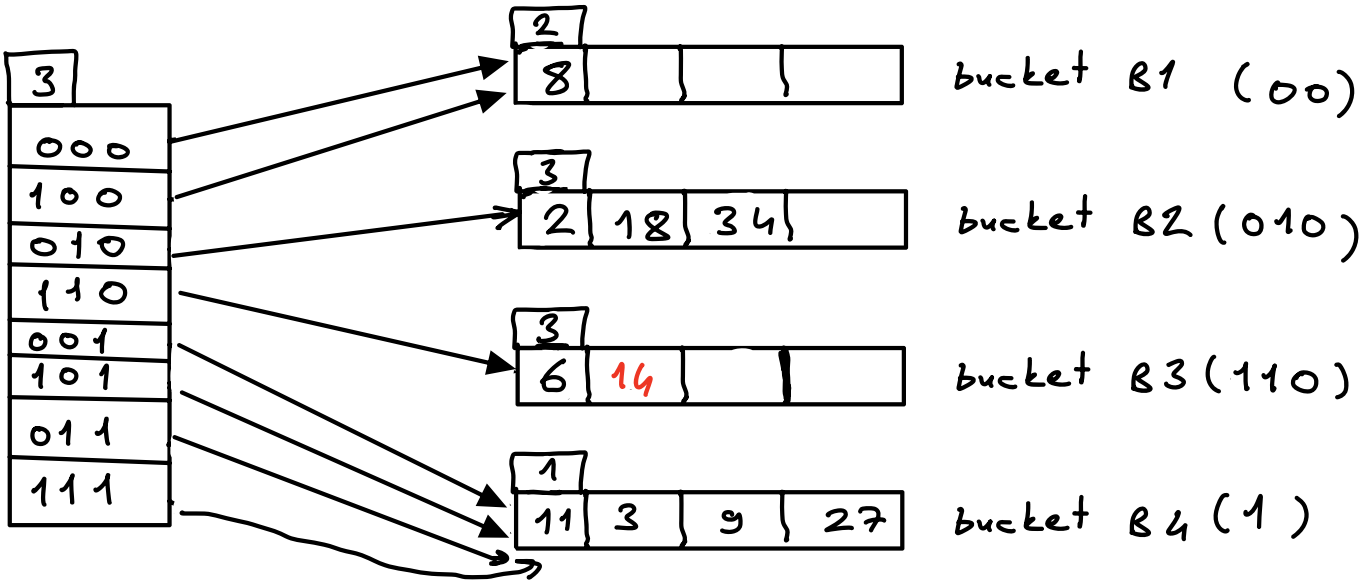

2 → 00(10)
18 → 01 00(10)
34 → 10 00(10)
8 → 10(00)
11 → 10(11)
3 → 0 0(11)
9 → 10(01)
27 → 110(11)

**(a)** What is the minimum number of key value insertions that will cause the bucket address table to double? Provide one such sequence of insertions and give the final contents of the index structure.

In the current state we have 2 full buckets. If we add 1 new key, then added node will split up according to Least Significant bits (LSB) of the keys. In order to minimize insertions, we need to fill the fuller node when we do splitting. Bucket 1 has 3 "10" LSB and 1 "00" LSB. Bucket 2 has 3 "11" LSB and 1 "01" LSB. We can add 1 "10" LSB to split the nodes. I will add 6 (110) whose LSB is "10".




If we add one more key whose LSB is "10". Index table must be doubled. I will add 14 (1110) whose LSB is "10". Note that B2 will be splitted according to their 3 LSB's.

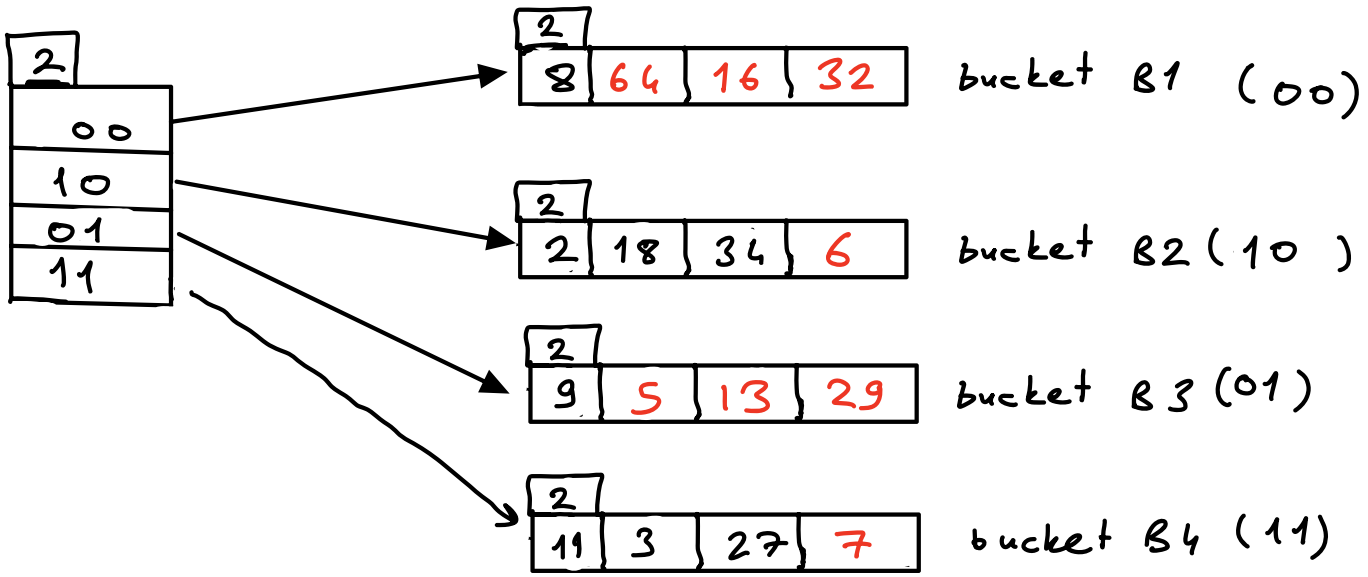



Minimum number of insertions is 2 and sequence is 6, 14.

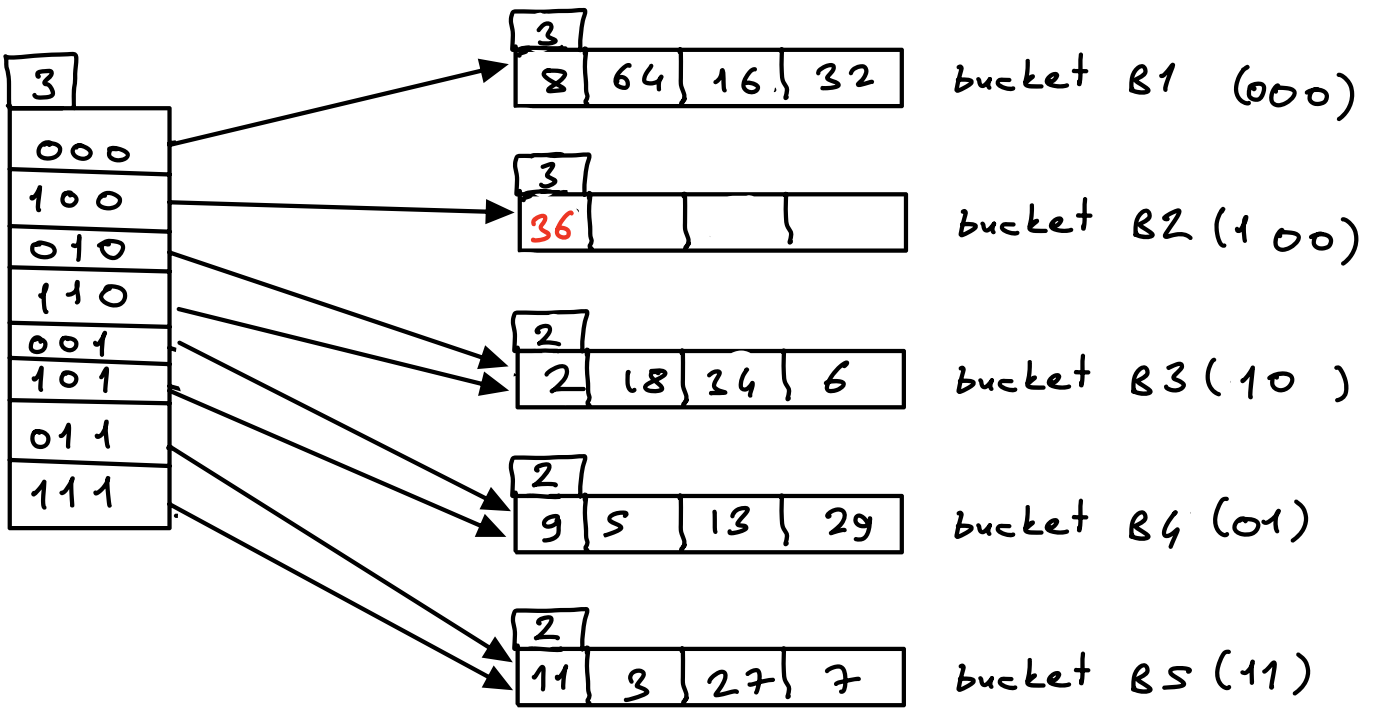
index structure.

**(b)** What is the maximum number of key value insertions that will cause the bucket address table to double? Provide one such sequence of insertions and give the final contents of the index structure.

There is 4 node potential in the current state and we have 2 full nodes currently. Therefore we can insert 8 key maximum without doubling bucket address table. The nineth value will certainly double table. At the beginning we have 3 "10" LSB, 1 "00" LSB, 3 "11" LSB and 1 "01" LSB. Therefore we need 1 "10", 3 "00", 1 "11" 3 "01" LSBs keys. We can add following values where their LSBs are circled: 6 (1(10)), 64 (1000(00)), 16 (100(00)), 32 (1000(00)), 7 (1(11)), 5 (1(01)), 13 (1 1(01)), 29 (1 1 1(01))




Then we can add 1 new value whose LSB can be anything. Let choose 36 (100100). Then final structure after adding 9 values:




Maximum number of insertions is 9 and sequence is 6, 64, 16, 32, 7, 5, 13, 29, 36.