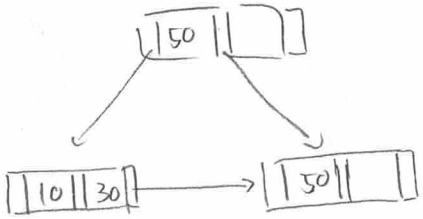
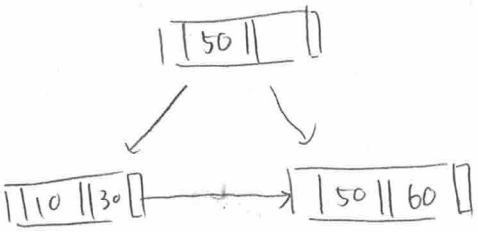
CS143 HW#4 Junhong Wang (5049411113)

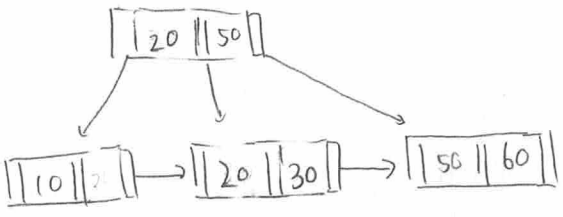
1. (a)



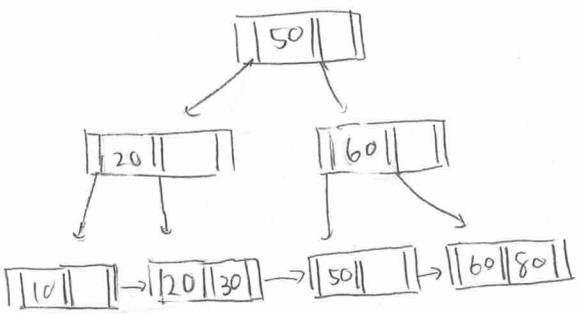
insert 60



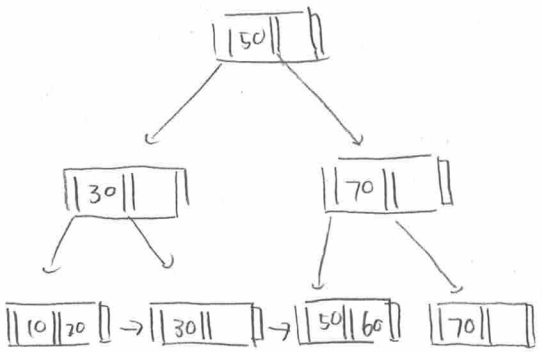
insert 20



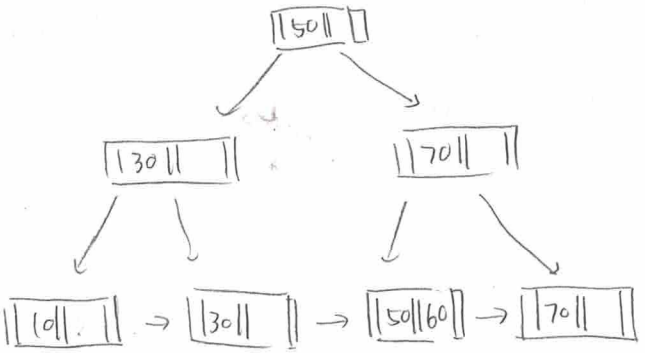
insert 80



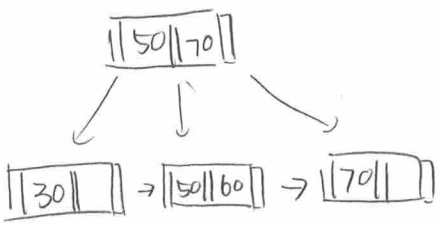
(b)



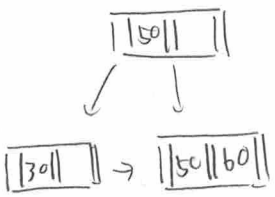
delete 20



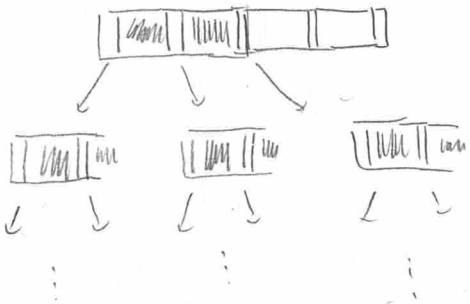
delete 10



delete 70



2.



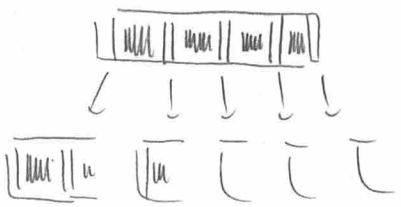
$2 \cdot 3^0$

$2 \cdot 3^1$

$2 \cdot 3^4 = 162$

$2 \cdot 3^5 = 486$

max height : 6



$4 \cdot 5^0$

$4 \cdot 5^1$

$4 \cdot 5^2 = 100$

$4 \cdot 5^3 = 500$

min height : 4

3. hash

106 → 01101010

115 → 01110011

916 → 10010100

0 → 00000000

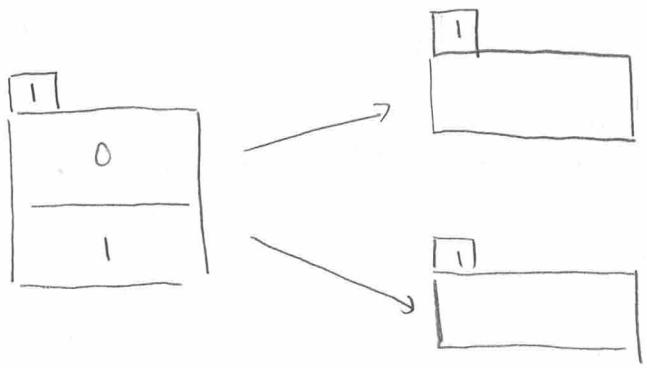
96 → 01100000

126 → 01111110

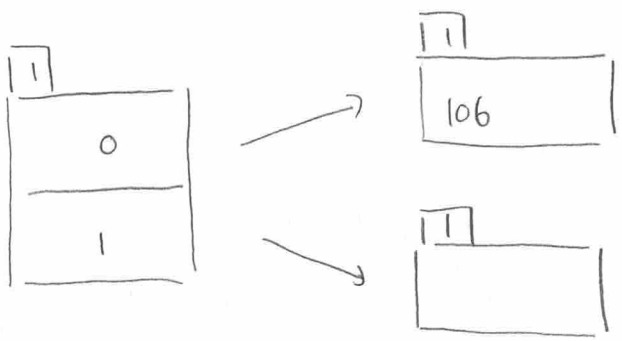
16 → 00010000

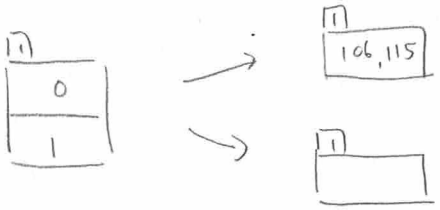
15 → 00001111

31 → 00011111



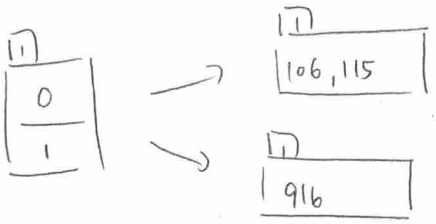
insert 106



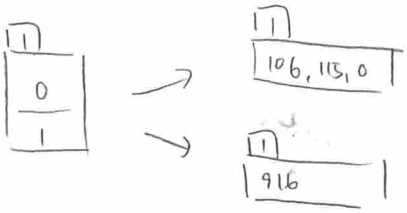


insert 115

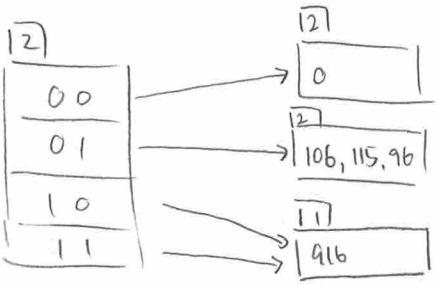
insert 916



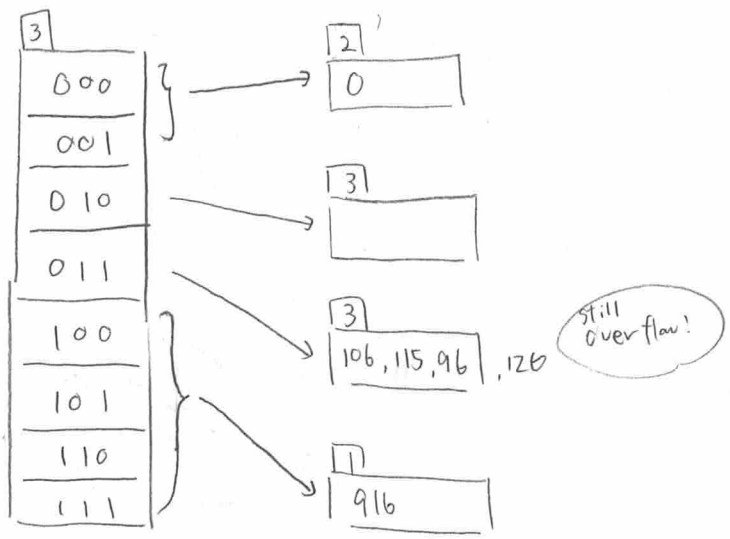
insert 0

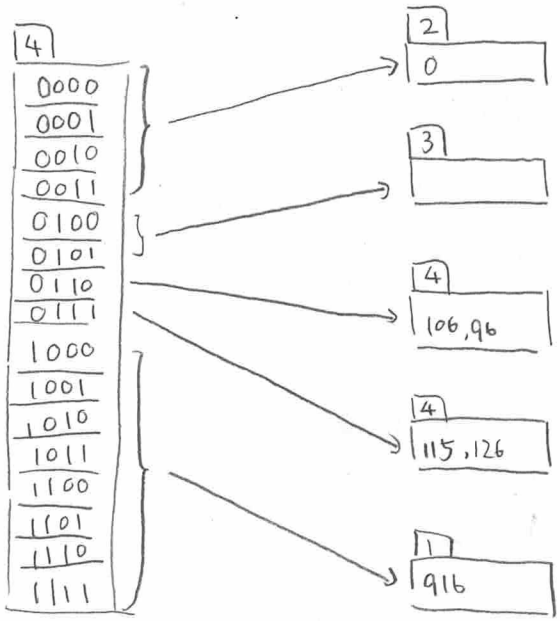


insert 96

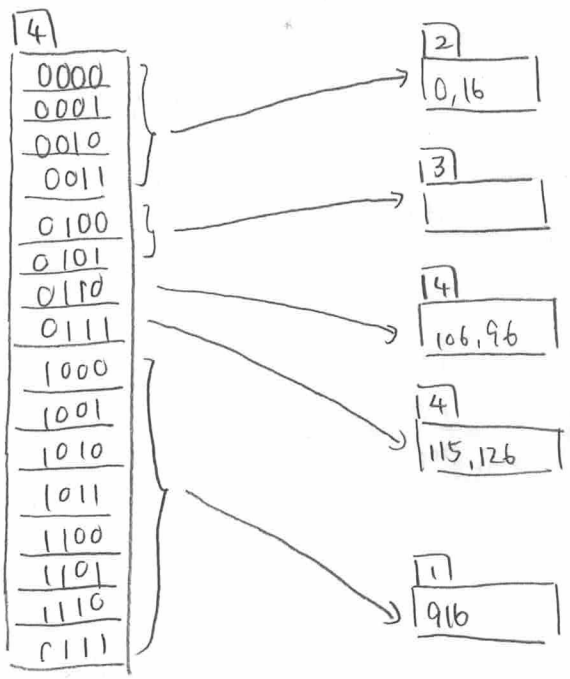


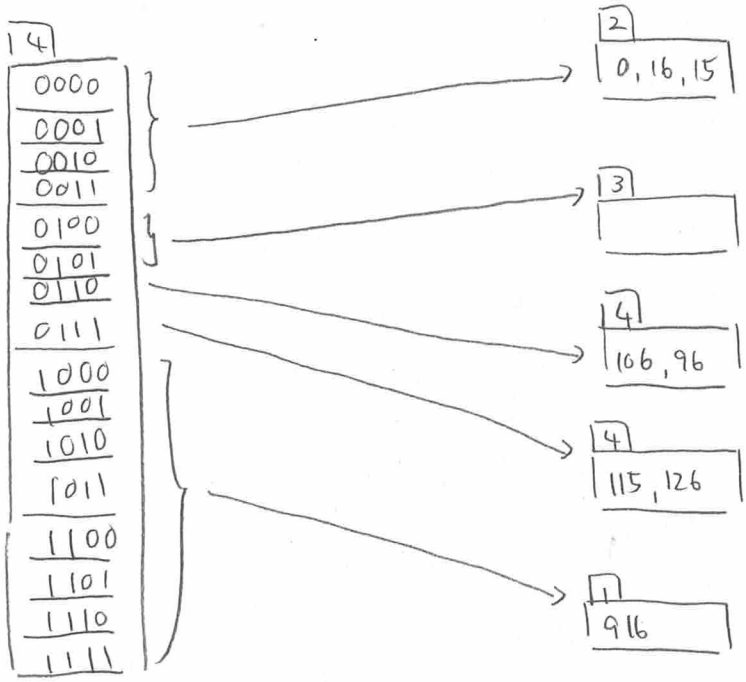
insert 126




4
0000
0001
0010
0011
0100
0101
0110
0111
1000
1001
1010
1011
1100
1101
1110
1111
2
0
3
4
106,96
4
115,126
1
916

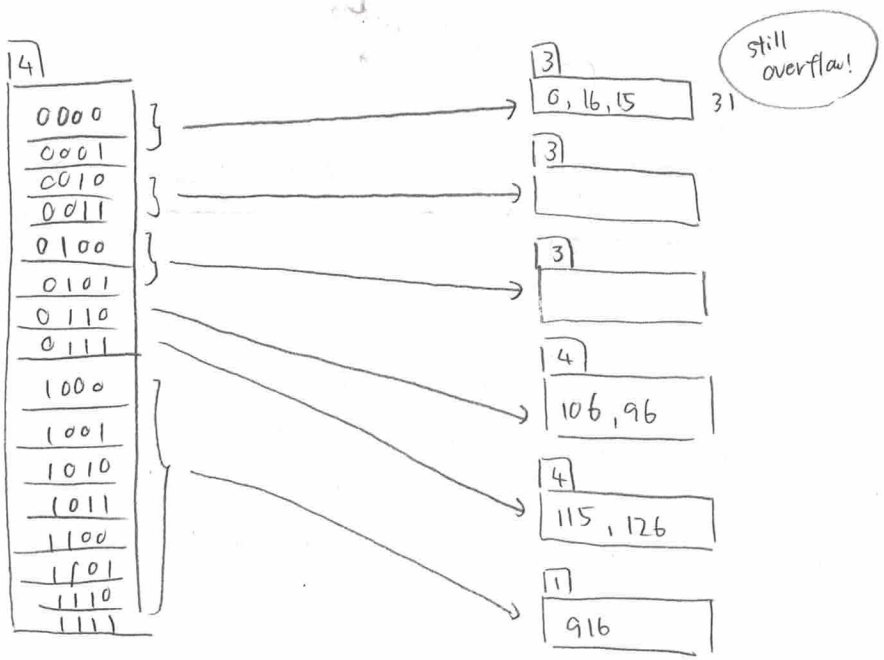
insert 16

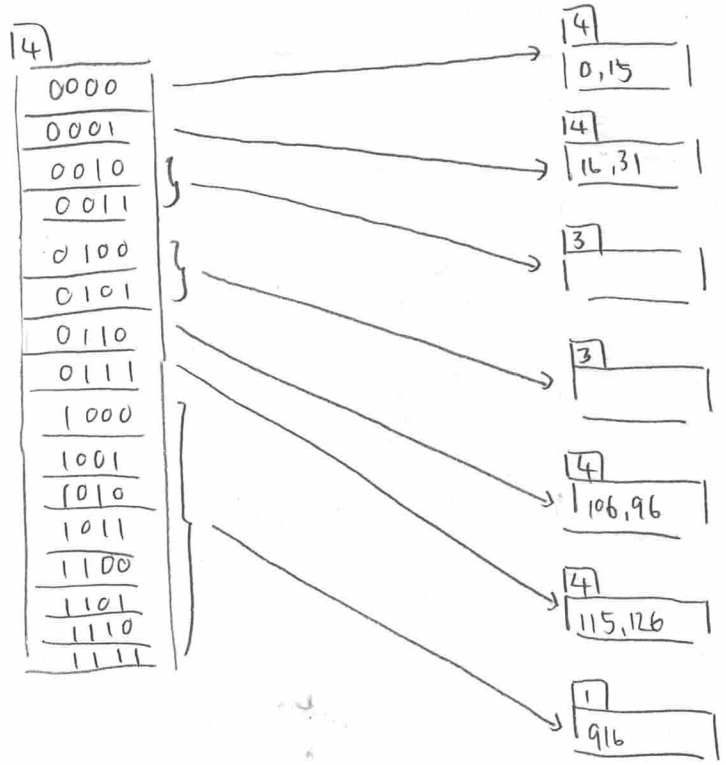

4
0000
0001
0010
0011
0100
0101
0110
0111
1000
1001
1010
1011
1100
1101
1110
1111
2
0,16
3
4
106,96
4
115,126
1
916



insert 15

insert 31




4
0000
0001
0010
0011
0100
0101
0110
0111
1000
1001
1010
1011
1100
1101
1110
1111
4
0,15
4
16,31
3
3
4
106,96
4
115,126
1
916

4. 1.

$R(A, B) \rightarrow$ 5000 tuples and 20 bytes / tuple

$S(B, C) \rightarrow$ 500 tuples and 190 bytes / tuple

R: 20 bytes / tuple = 1000 bytes / 50 tuples = 100 disk blocks / 5000 tuples

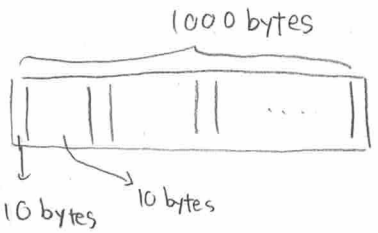
S: 190 bytes / tuple = 950 bytes / 5 tuples $\Rightarrow$ 5 tuples / disk block

$\Rightarrow$ 500 tuples / 100 disk blocks

So 100 + 100 = **200 [disk blocks]**

4.2.

1 node



n pointers and n-1 keys

$$10n + 10(n-1) \leq 1000$$

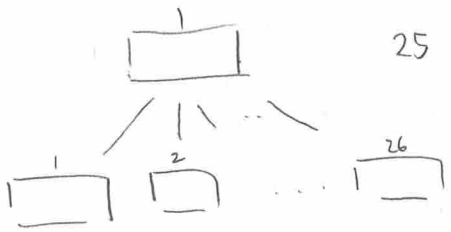
$$20n \leq 1010$$

$$n \leq 50.5$$

$$\boxed{50}$$

4.3.

(a) $n = 50$



$25\ (26)^0 = 25$ keys

$25 \cdot (26)^1 = 650$ keys > 500

500 tuples

We need depth = 2

Note $25 \cdot 20 = 500$

So we need 20 nodes in level 2

Thus 1 (root) + 20 = **21 nodes**

(b) Read one block at a time.

Reading the tuples from S: 100

Writing the constructed B+ tree : 21

Thus total of **121 IOs**.

4. Cost = $\lceil b_r / (M-2) \rceil \times b_s + b_r$

$\lceil 100 / 28 \rceil \times 100 + 100 =$ **500 (disk IOs)**

It's worthwhile because it would have taken only 121 + 21 + 100 = 242 disk IOs for optimized algorithm. even considering the index construction overhead.