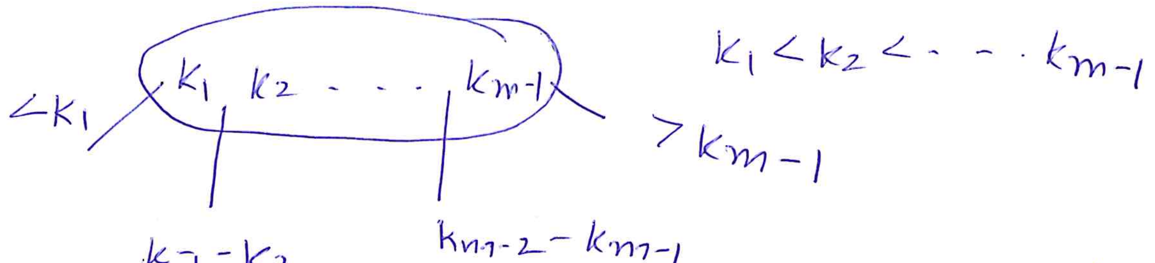
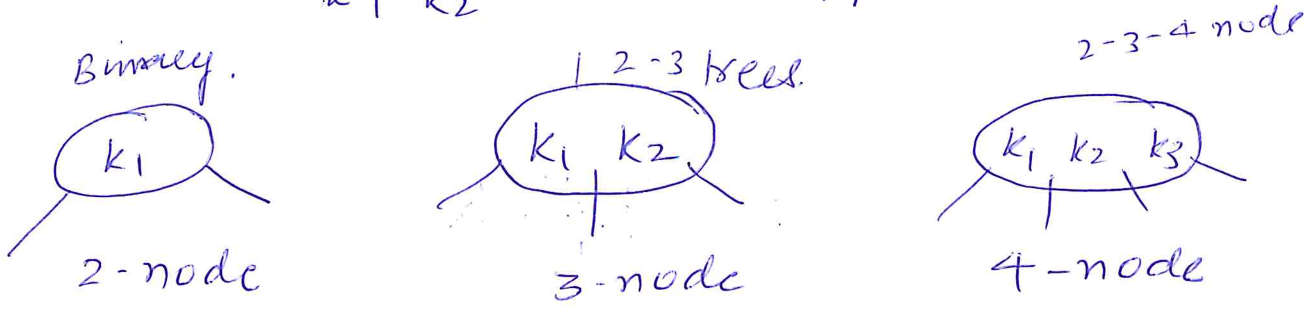
B Trees. – Storage System

m-ary – each node has upto (m-1) keys

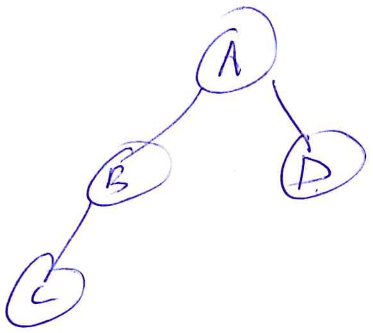




$\boxed{1}$ Deletion 2-3-4 trees.

– 3 cases

k - element to be deleted

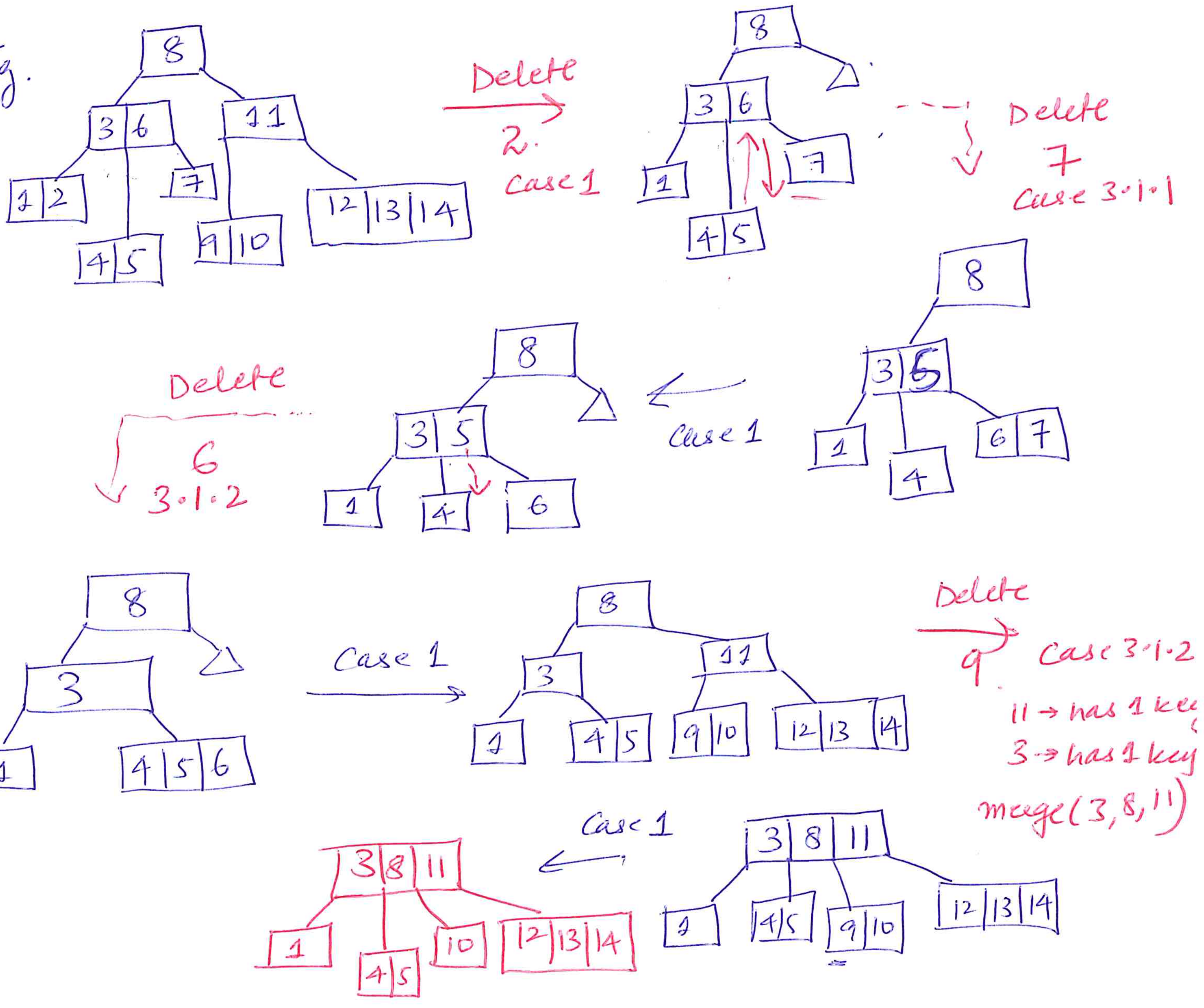


- If the element (k) is.

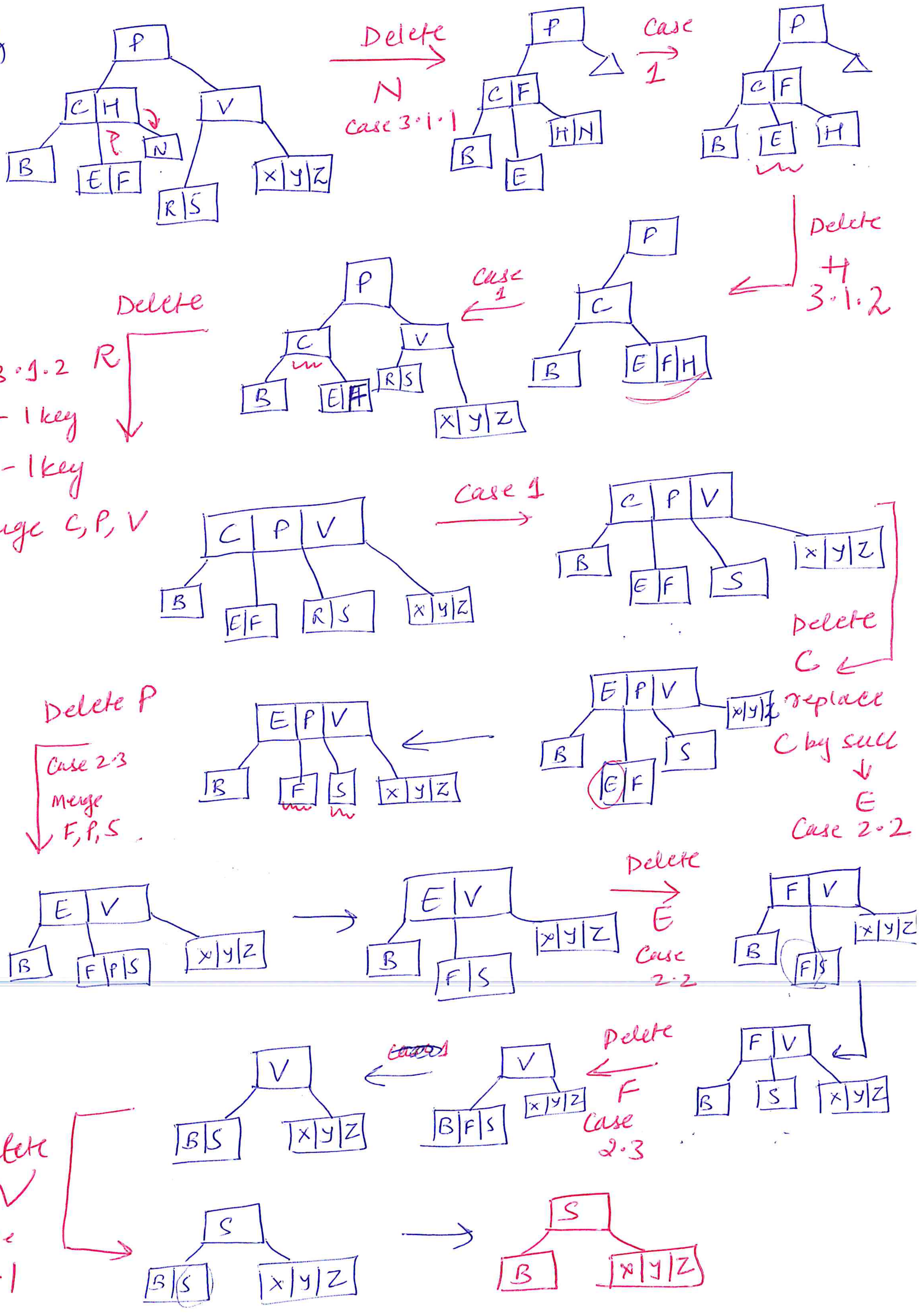
```
1  if (k is in the leaf) → atleast
                           2 keys
       remove 'k'

2. else if (k is an internal node)
   2.1 if (left-child ≥ 2 keys)
       {
         replace (k, pred)
         remove (pred)
       }
   2.2 else if (right-child ≥ 2 keys)
       {
         replace (k, succ)
         remove (succ)
       }
   2.3 else // both children have 1 key.
       {
         merge (left-ch, k, right-ch)
         remove (k)
       }
```

```
3. else // k is not in internal node.
        ch → to-be-visited so far. [not on root]
   3.1 if (ch has only 1 key)
      3.1.1 if (ch's sibling ≥ 2 keys)
               rotate a key into
                     ch
      3.1.2 else if (ch's sibling has 1 key)
               merge (ch, parent, sibling)
```

#g.



#3


Delete N
Case 3.1.1
Case 1
Delete H
3.1.2
Case 1
Delete R
3.1.2
V - 1 key
C - 1 key
merge C, P, V
Case 1
Delete C
replace C by succ
E
Case 2.2
Delete P
Case 2.3
merge F, P, S
Delete E
Case 2.2
Delete F
Case 2.3
~~Case 1~~
Delete V
Case 2.1