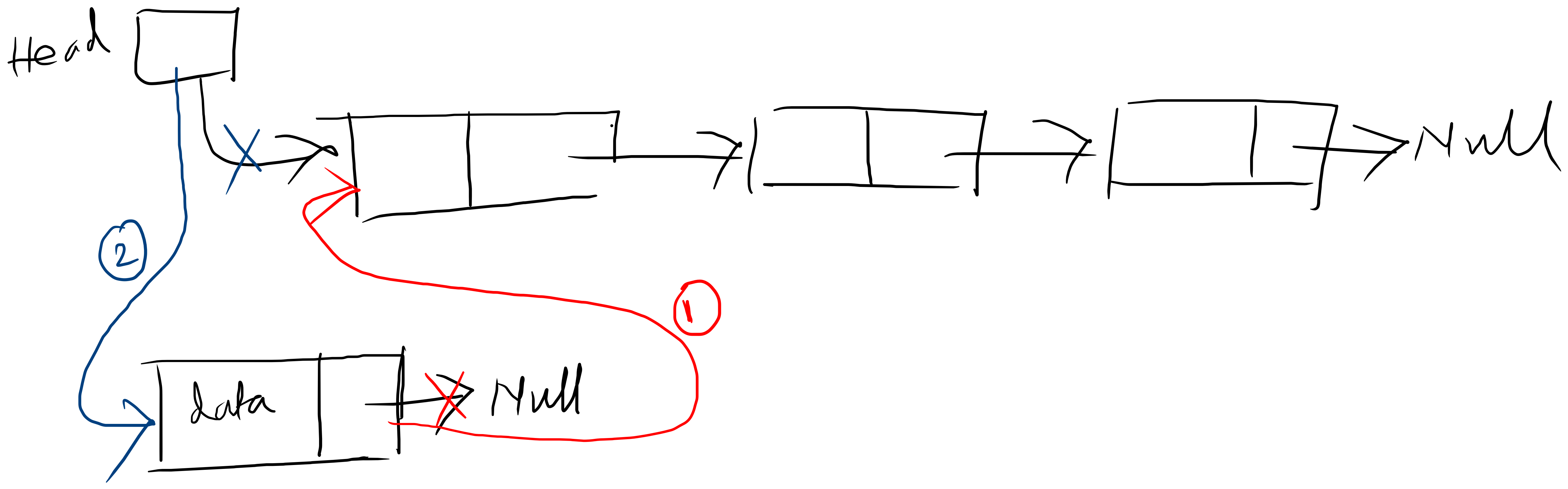
Insertion

- Insert-at-begining
- insert-at-end
- insert_at_a-position

Insert_at begining

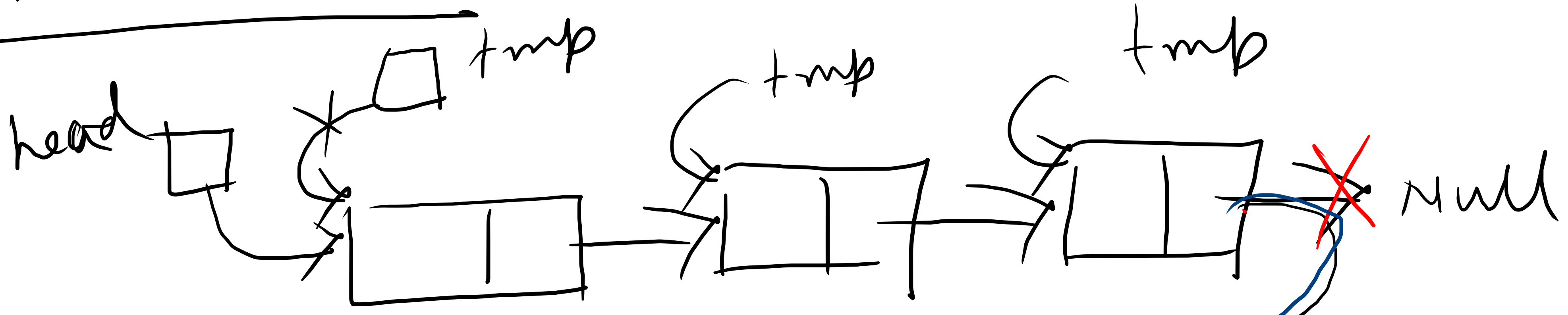


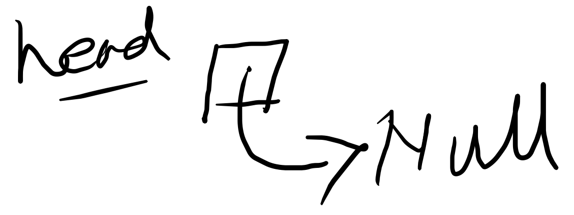
```
Insert_at_beg (L, K)
{ create a node newnode
  newnode . data = K
  newnode . next = Null

  newnode.next = L. head
  L. head = newnode
```

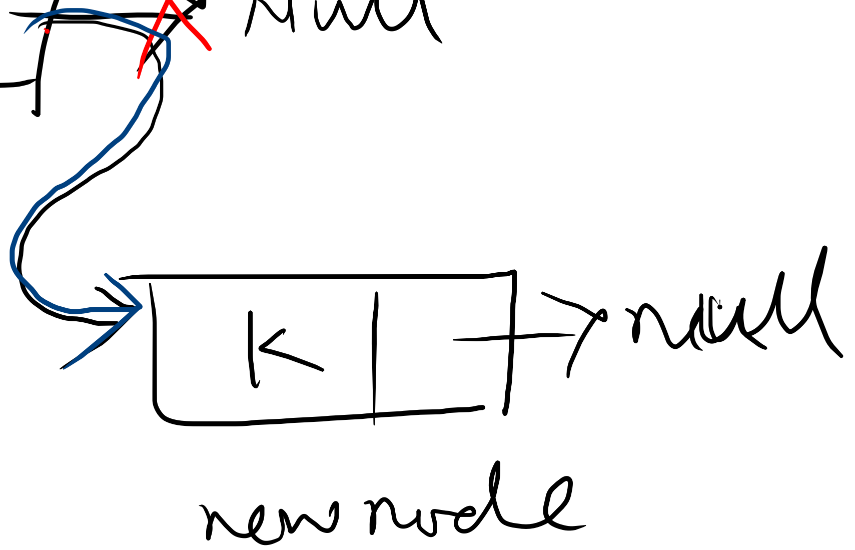
Time : $\theta(1)$

space : $\theta(1)$

Insert_at_end







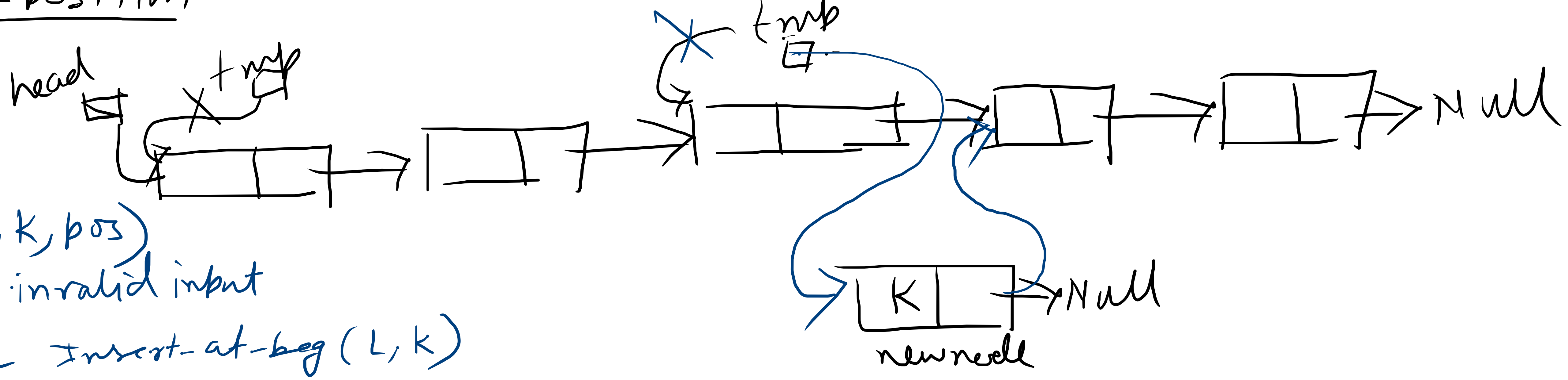
```
Insert_at_end (L, K)
 create a node newnode
   newnode.data = K
   newnode.next = Null
   tmp is a node pointer
   tmp = L.head
     if tmp = Null
       L.head = newnode
     else
        while tmp.next ≠ Null
              tmp = tmp.next
        tmp.next = newnode.
```

Time: $O(n)$

Space: $\theta(1)$

Insert-at-a-position

Pos = 4



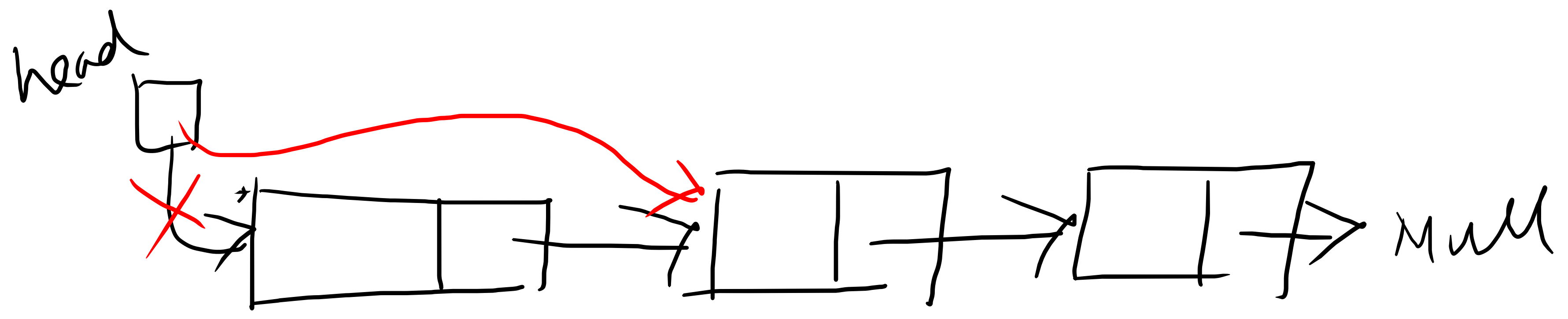
```
Insert-at-a-pos(L, K, pos)
If pos<1 then invalid input
If pos = 1 then Insert-at-beg(L, K)
else
    count = 1
    tmp is the pointer to node
    tmp = L.head
    flag = 0
    while tmp ≠ Null
        if count = pos-1
            create a new node
            newnode.data = K
            newnode.next = Null
            newnode.next = tmp.next
            tmp.next = newnode
            flag = 1
            break
        tmp = tmp.next
        count = count+1
If flag = 0 invalid input
```

Time: $O(n)$

Space: $O(1)$

Deletion

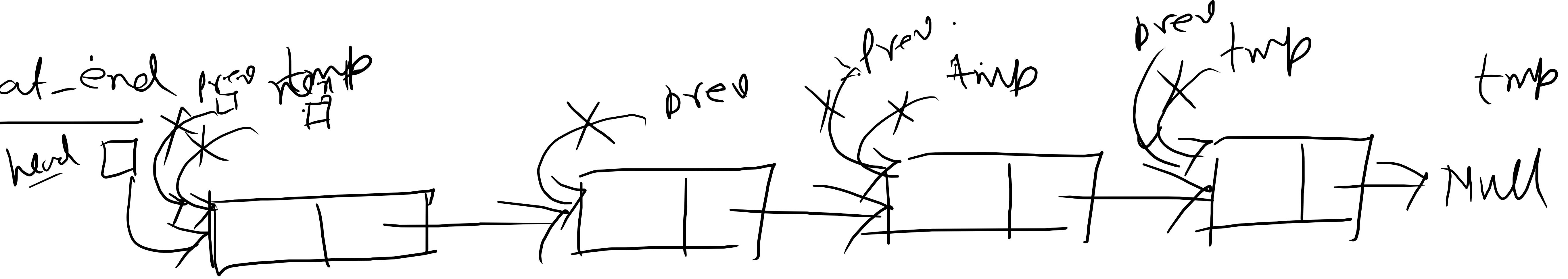
```
Deletion_at_beg (L)
  If L.head = Null
        Nothing to delete
  else
        L.head = L.head.next .
```



Time : $\Theta(1)$

Space : $\Theta(1)$

Deletion_at_end



```
Delete_atend ( L )
IF L.head = Null
        Nothing to delete
else
    prev and tmp are two node pointers
    prev = l.head
    tmp = L.head
    While tmp ≠ Null
        Prev = tmp
        tmp = tmp.next
    prev = Null
```

Time = $O(n)$

space = $\theta(1)$

H/w

Delete at a position