

SORTING A LINKED LIST CONTAINING 0's 1's & 2's

★ In this problem, we are given a linked list containing only 0's, 1's & 2's and our job is to return the sorted version.

Bruteforce solution is to create 3 counters, each keeping track of how many 0's, 1's and 2's occur in one pass. Then in the next pass we just replace all values. Time Complexity is $O(2N)$ while Space Complexity is constant.

Optimal solution is to modify in the links such that 0's only point to 0's, 1's only point to 1's and 2's only point to 2's.

C++ :

```
Node * sortLL012(Node * head) {  
    Node * zero = nullptr;  
    Node * one = nullptr;  
    Node * two = nullptr;  
    Node * mover = head;  
    Node * zero_h = nullptr;  
    Node * one_h = nullptr;  
    Node * two_h = nullptr;  
}
```

```

while (mover != nullptr) {
    if (mover->data == 0) {
        if (zero == nullptr) {
            zero = mover;
            zeroh = zero;
        } else {
            zero->next = mover;
            zero = zero->next;
        }
    } else if (mover->data == 1) {
        if (one == nullptr) {
            one = mover;
            oneh = one;
        } else {
            one->next = mover;
            one = one->next;
        }
    } else {
        if (two == nullptr) {
            two = mover;
            twoh = two;
        } else {
            two->next = mover;
            two = two->next;
        }
    }
    mover = mover->next;
}
zero->next = oneh;
one->next = twoh;
two->next = nullptr;
return zeroh;
}

```