

```
/*
Sorting+Binary search (Memory leaks)
Use dummy node to make a unified pattern to all nodes

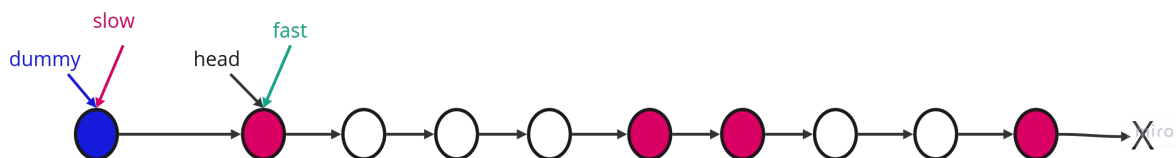
```

Time complexity:  $O(n \log n)$

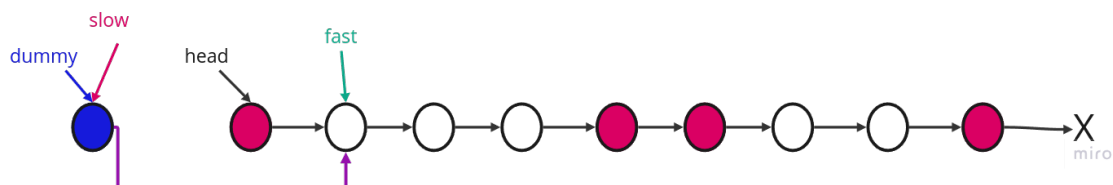
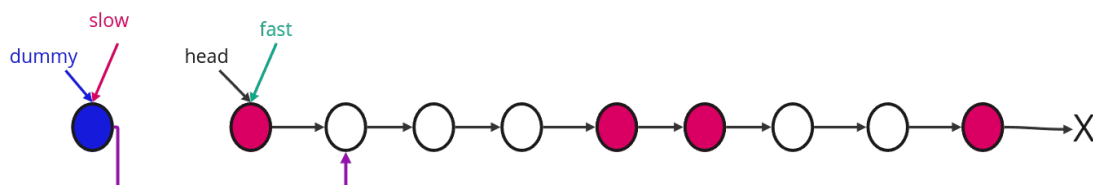
Space complexity:  $O(\log n)$

```
*/
typedef std::vector<int> vi;
class Solution {
public:
    ListNode* modifiedList(vi& nums, ListNode* head){
        std::sort(nums.begin(),nums.end());

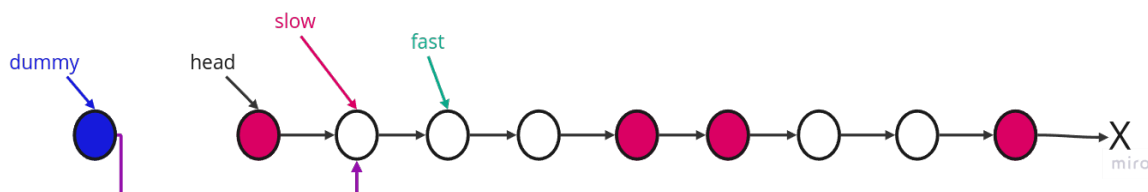
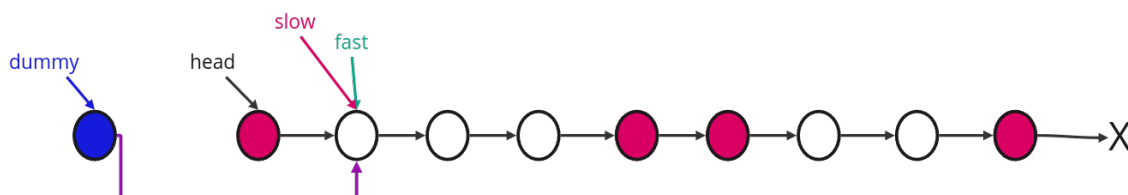
        ListNode* dummy=new ListNode(0,head);
        ListNode* fast=head;
        ListNode* slow=dummy;
        while(fast){
            if(std::binary_search(nums.begin(),nums.end(),fast->val)) slow->next=fast->next;
            else slow=fast;
            fast=fast->next;
        }
        return dummy->next;
    }
};
```



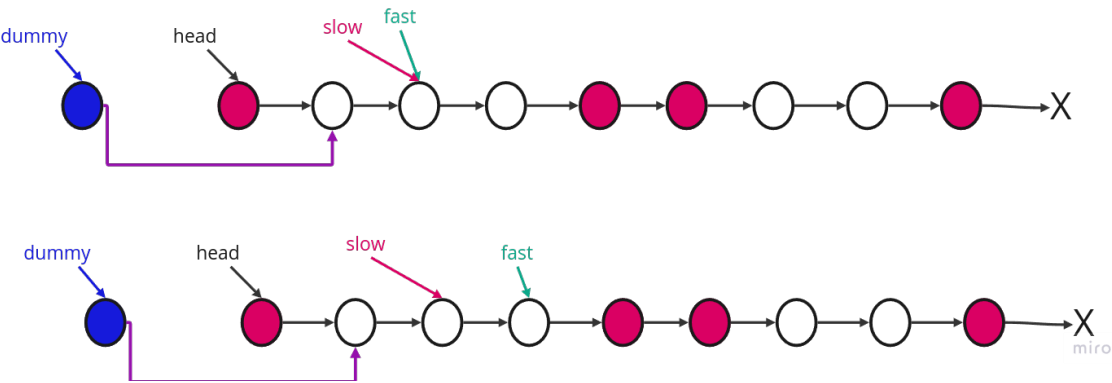
fast points to the node to delete:



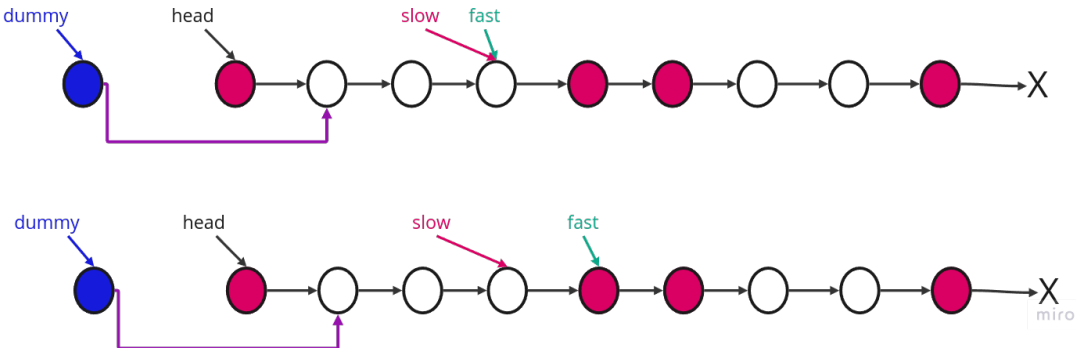
fast does not point to the node to delete:



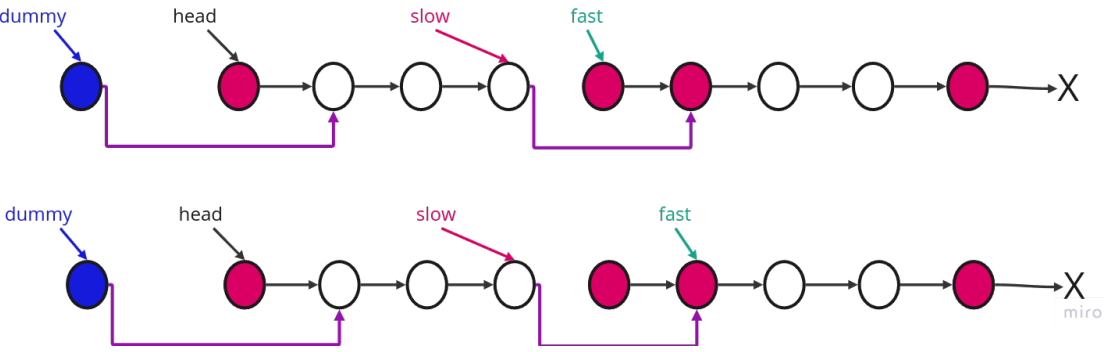
fast does not point to the node to delete:



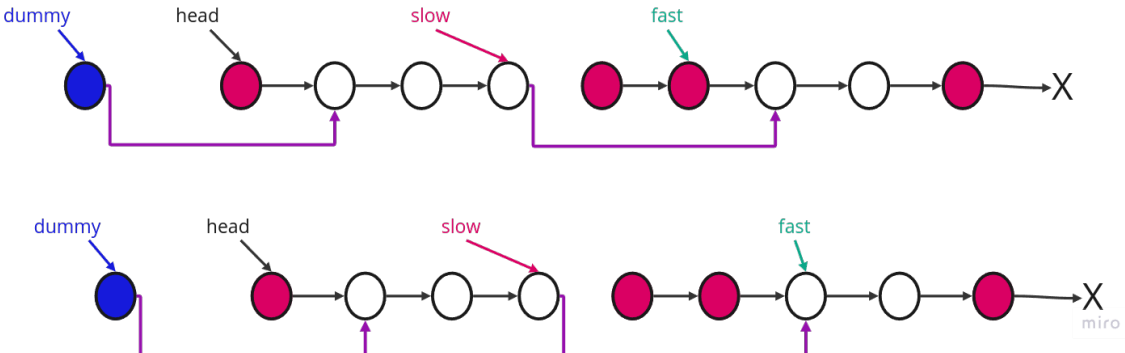
fast does not point to the node to delete:



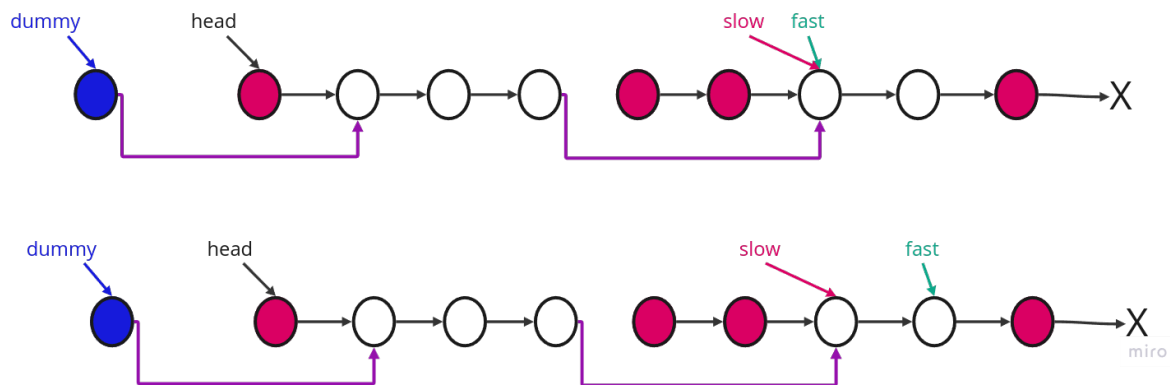
fast points to the node to delete:



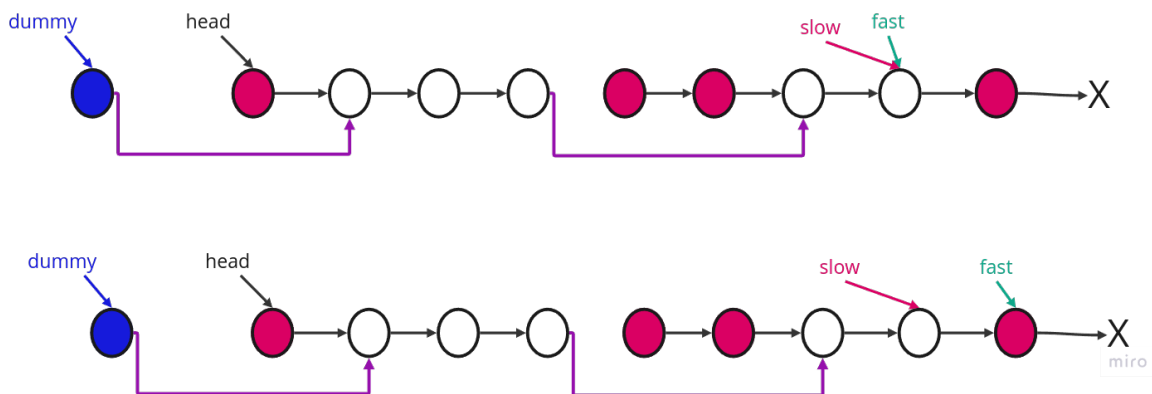
fast points to the node to delete:



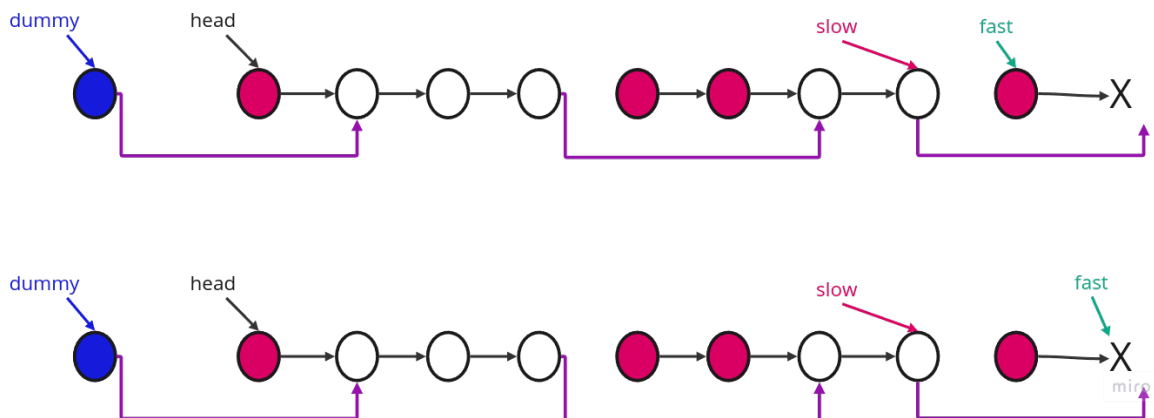
fast does not point to the node to delete:



fast does not point to the node to delete:



fast points to the node to delete:



At end the we get:

