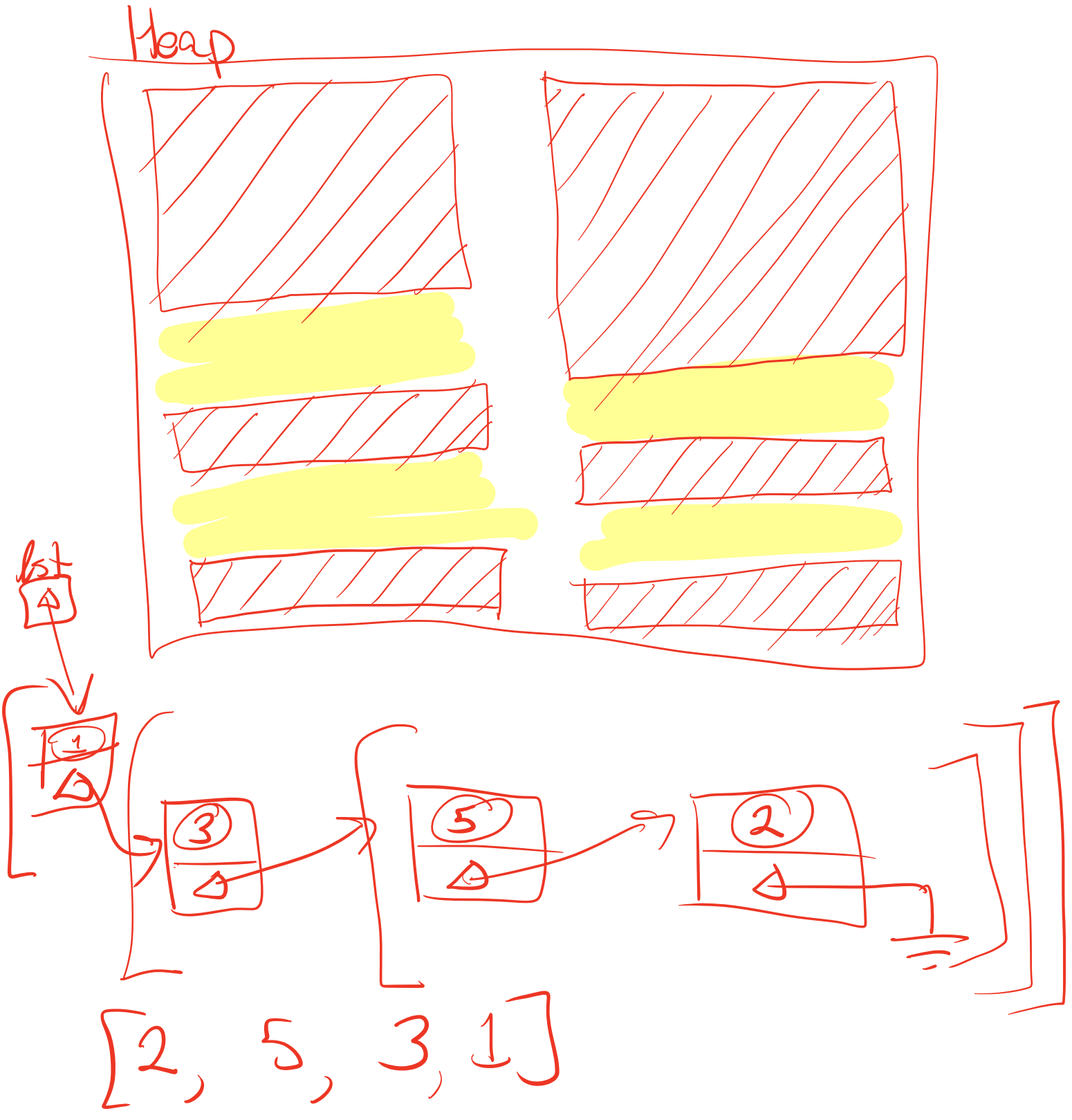
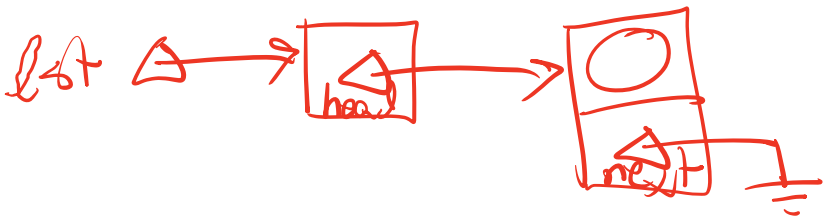


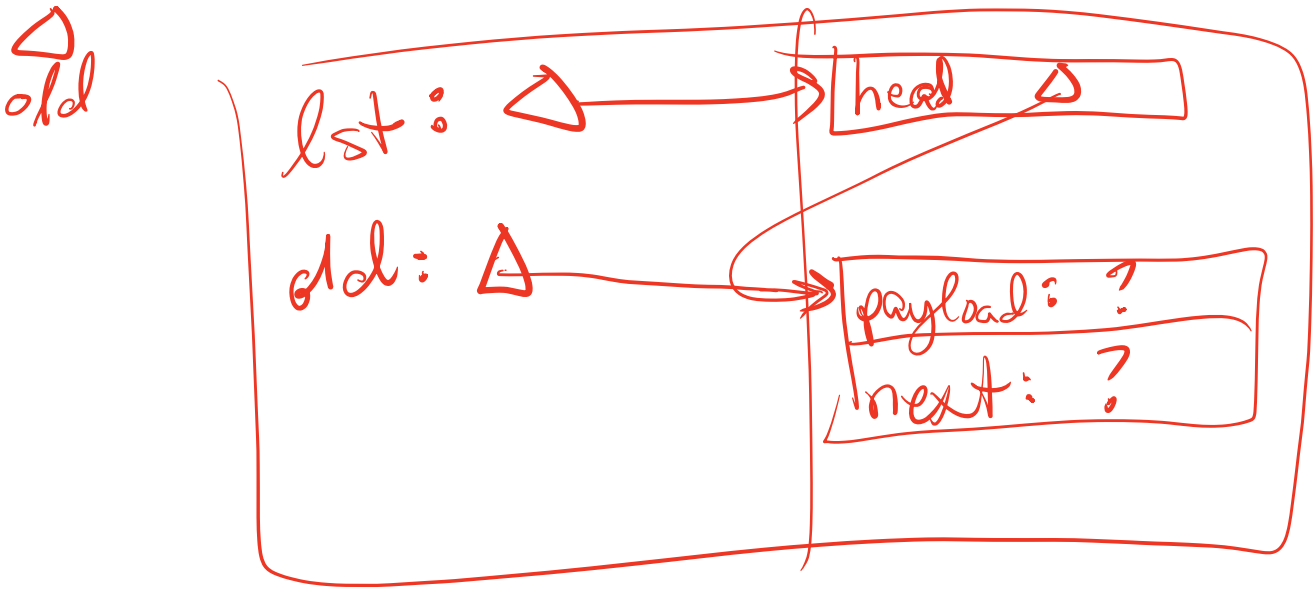
[2, 5, 3, 1]

List: Node*

Node: int | Node*

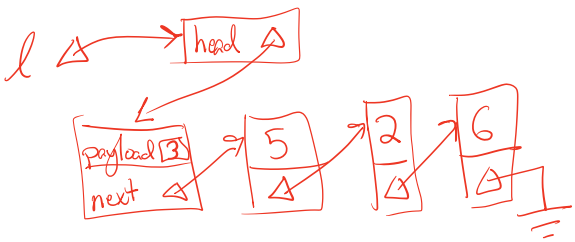
```
node* old = lst->head;
```

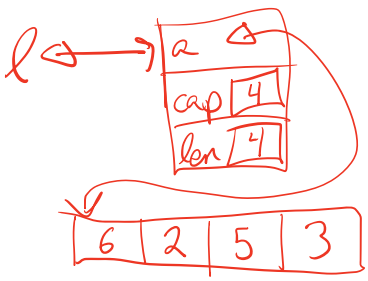




Sep 29

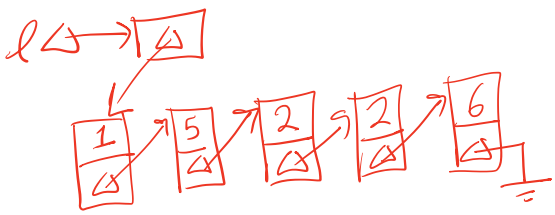
$l = [6\ 2\ 5\ 3]$

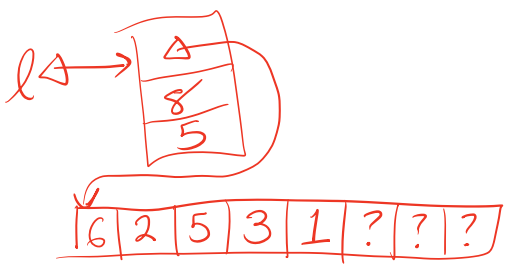




append(l, 1);

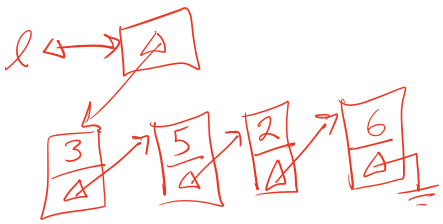
$l = [6\ 2\ 5\ 3\ 1]$

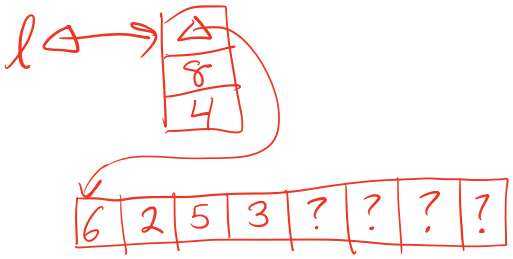


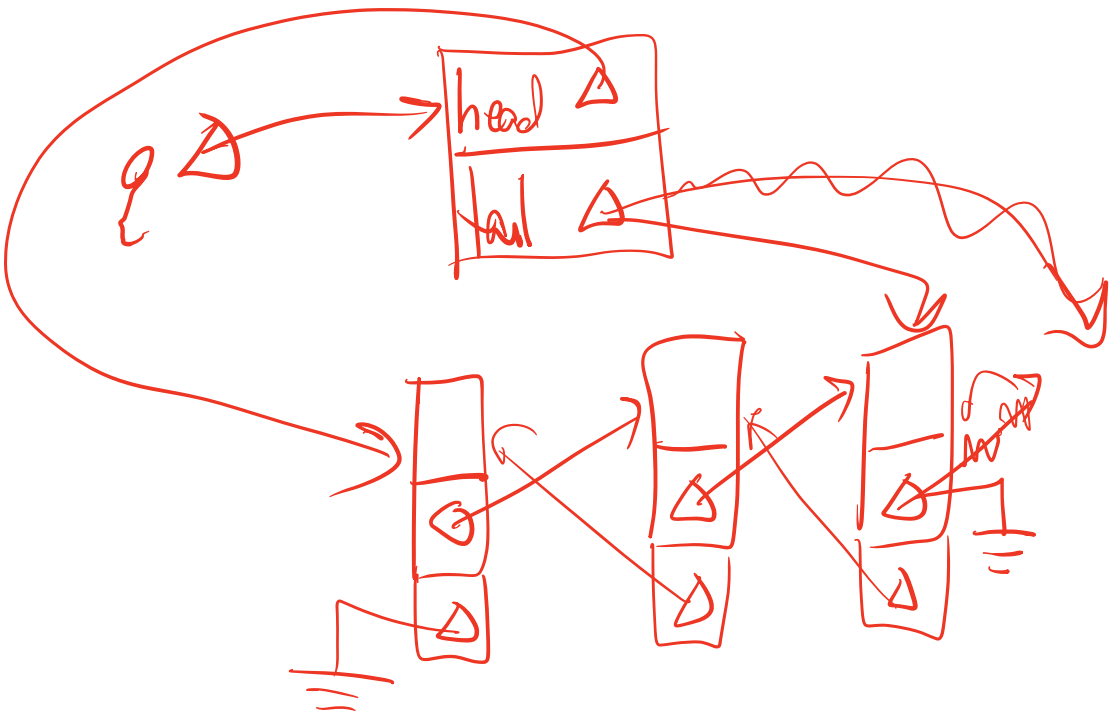


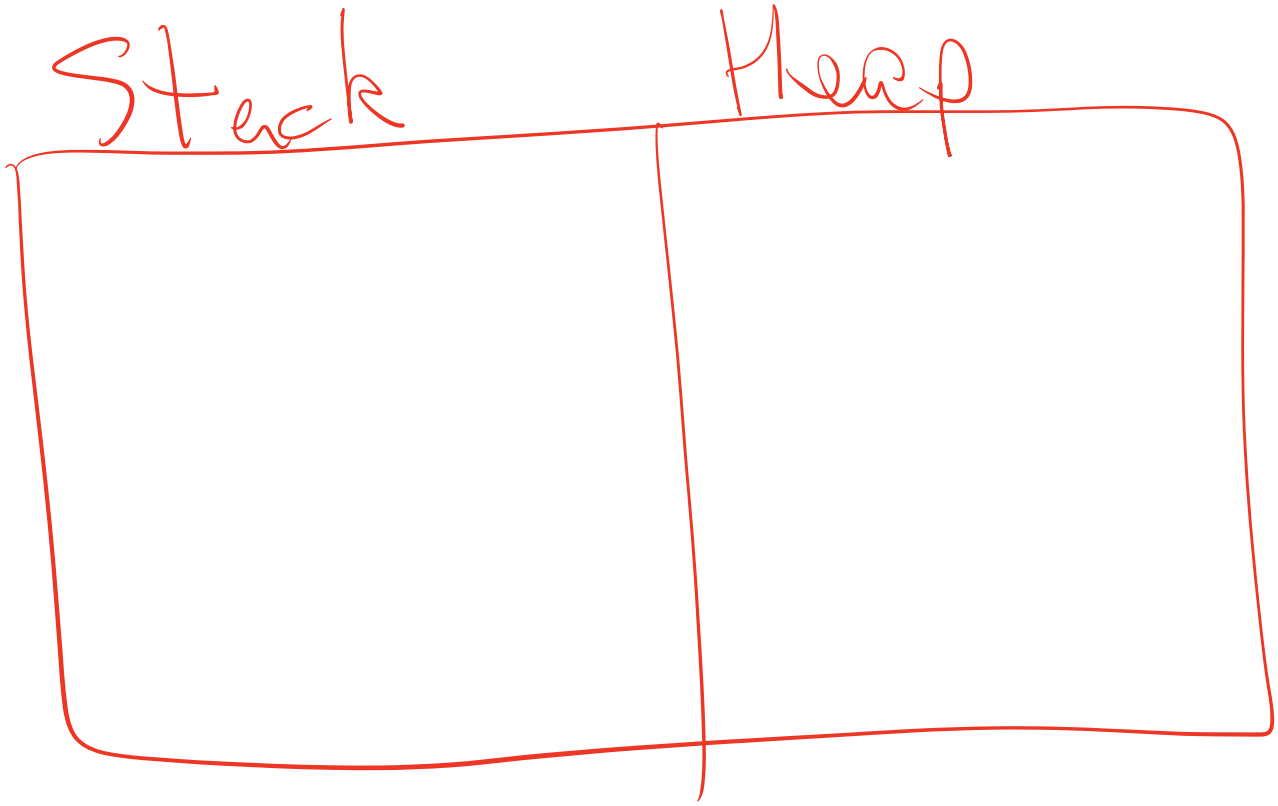
x = remove_last(l);

l = [6 2 5 3] x = 1



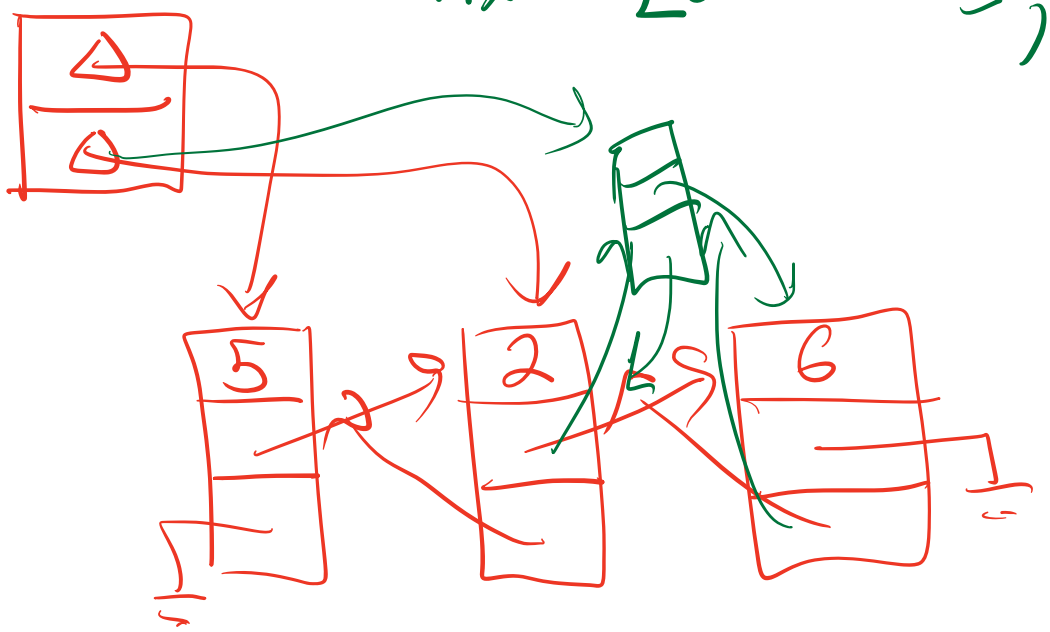




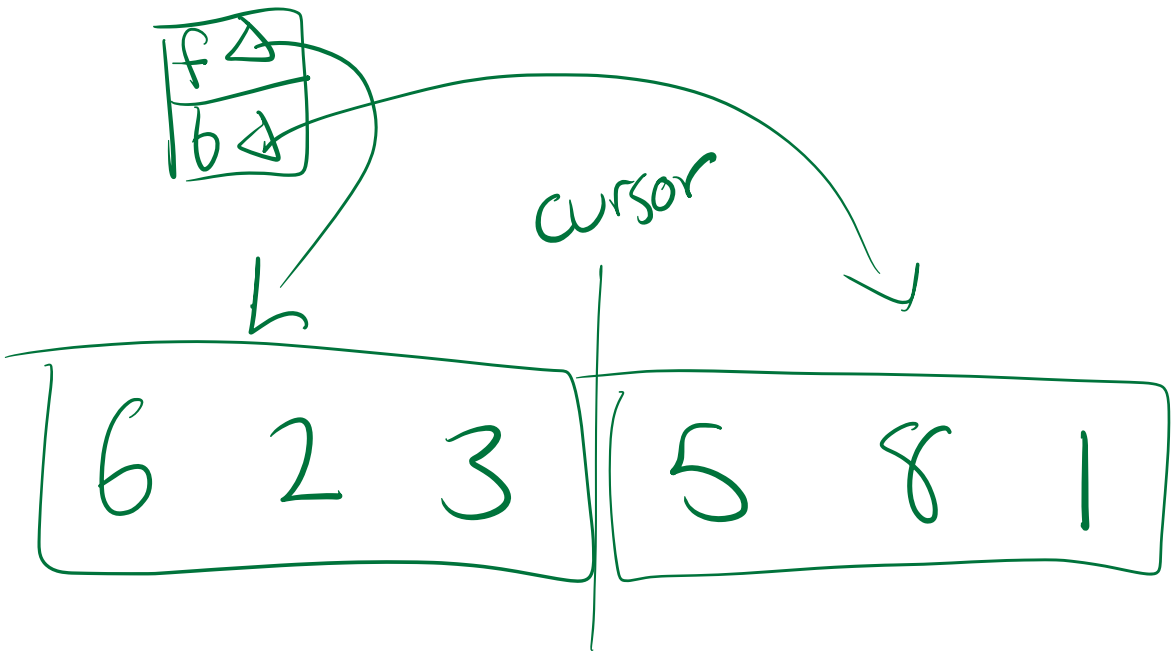


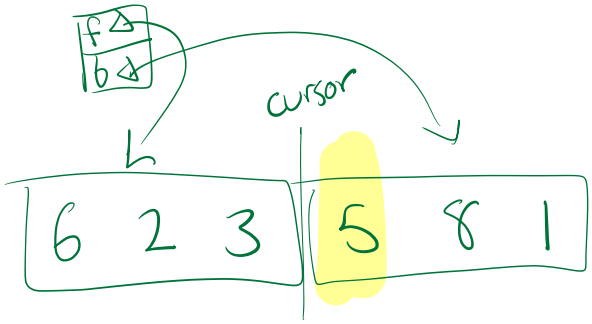
if ← while ← main

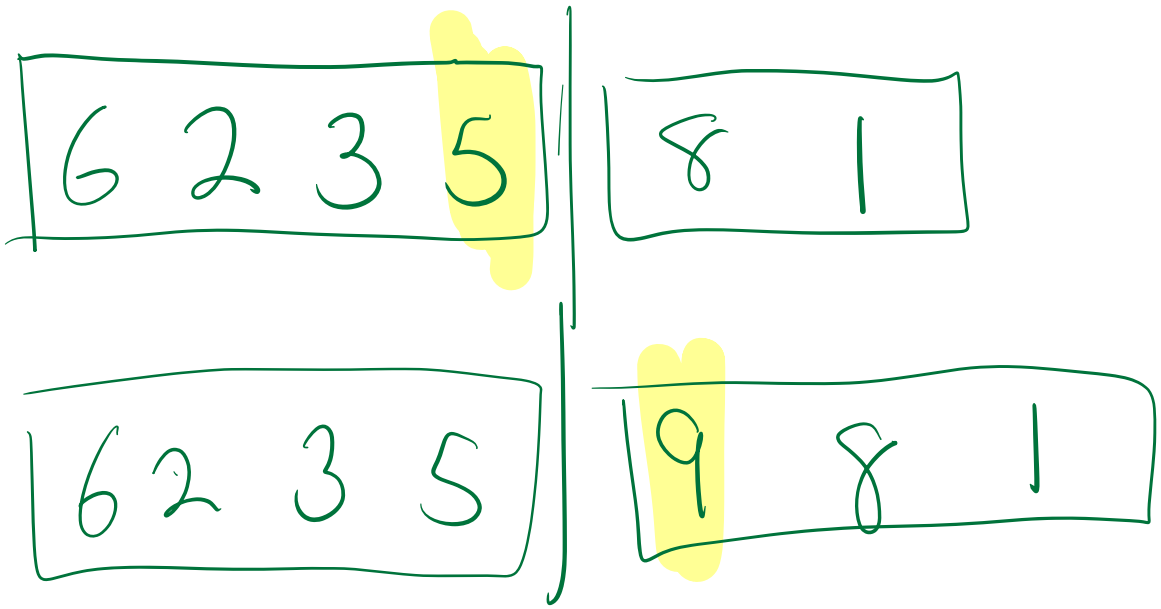
nl = [6 2 5], 1

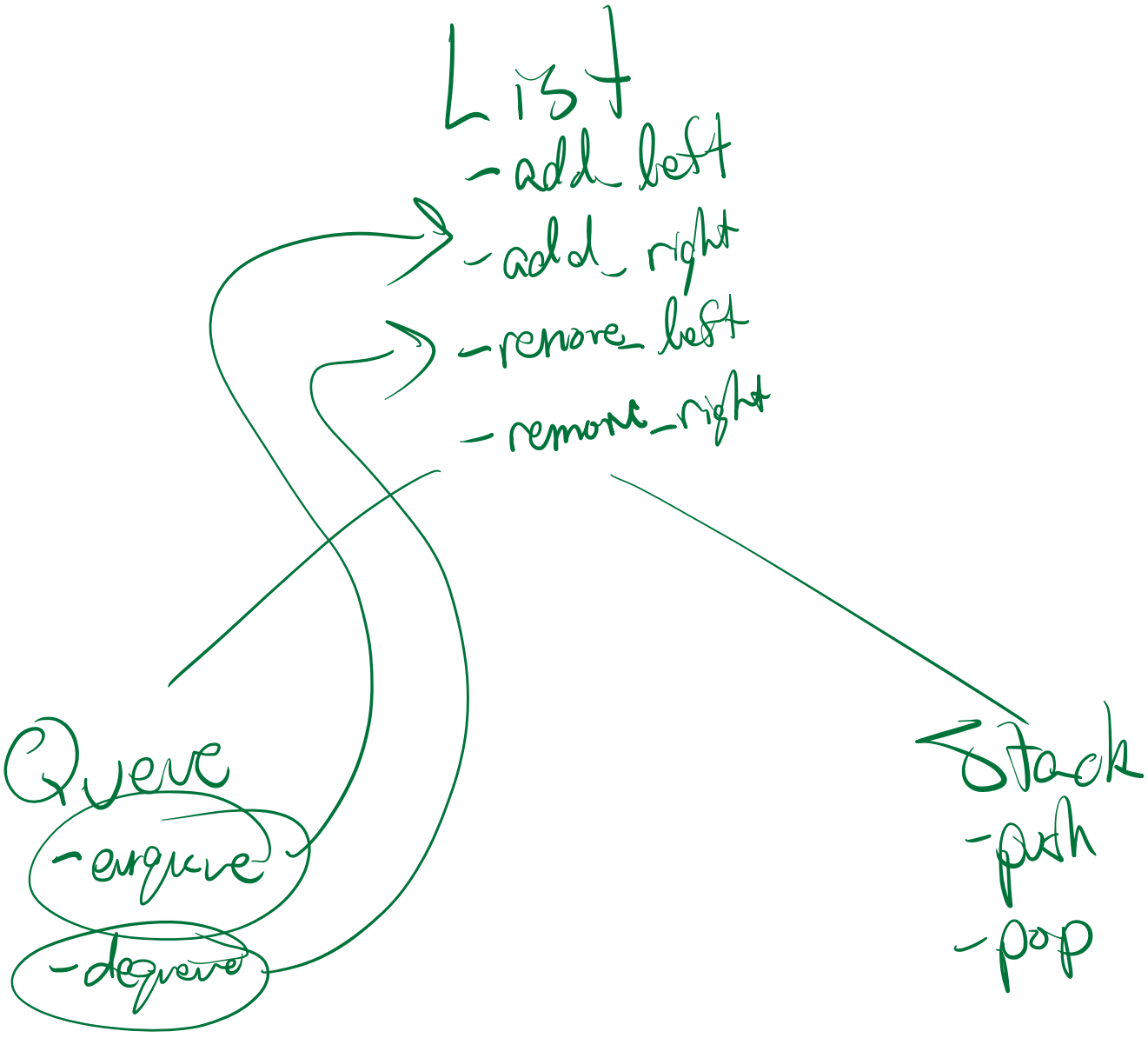


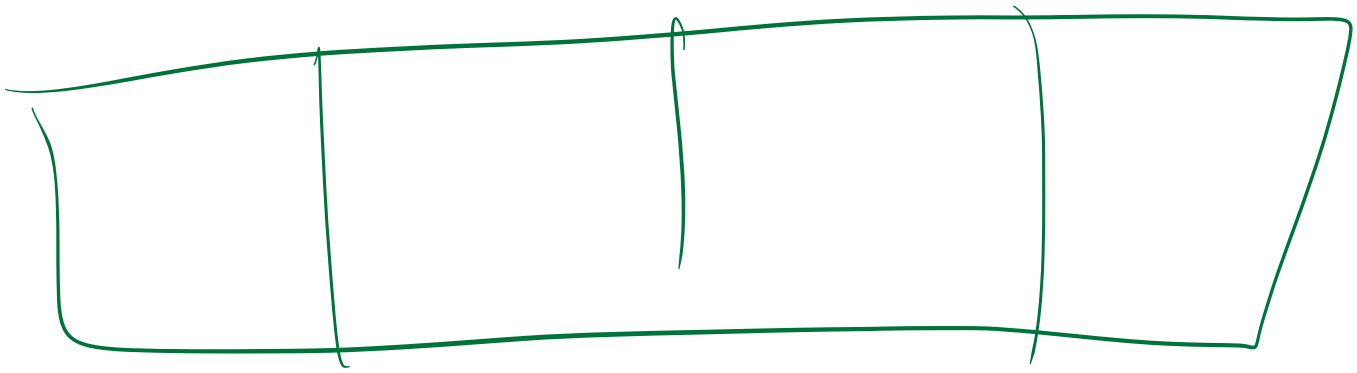
l = [2




f
b
cursor
6 2 3 5 8 1


6 2 3 5
8 1
6 2 3 5
9 8 1


List
- add_left
- add_right
- remove_left
- remove_right
Queue
- enqueue
- dequeue
Stack
- push
- pop



```
sort(List l, (bool)(*thing, *thing)
                    comes_before
     .
     .
     .
     if(comes_before(a, b)) {
          .
          .
     }
     .
     .
     .
```

```
int compare(int a, int b) {
    if(a < b) return -1;
    if(a > b) return 1;
    if(a == b) return 0;
```