ECE 312 (SP26):

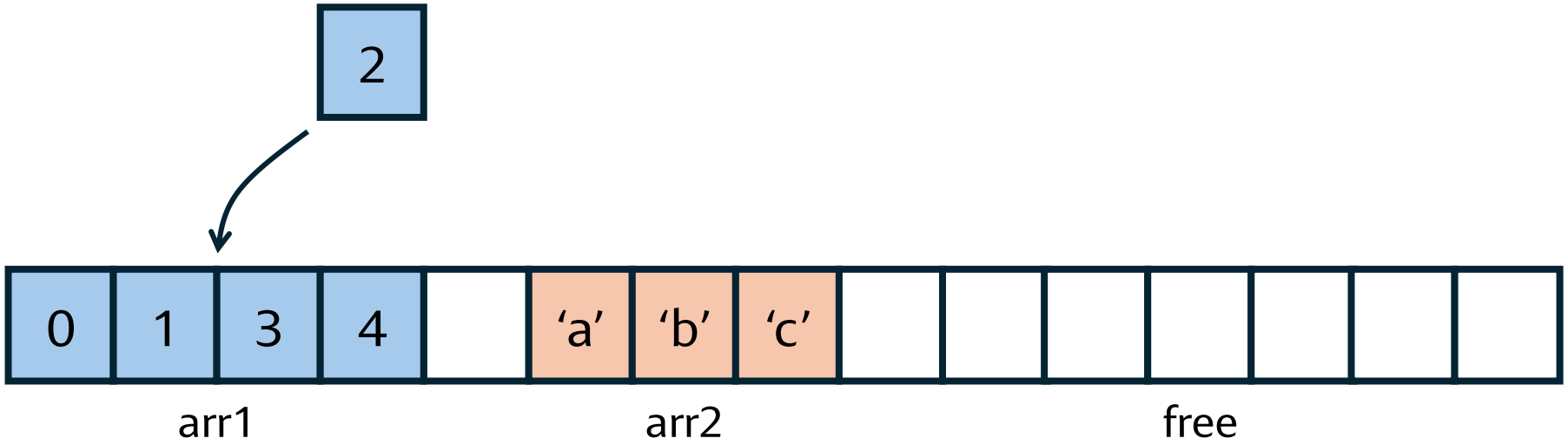
Linked List

Neil Zhao

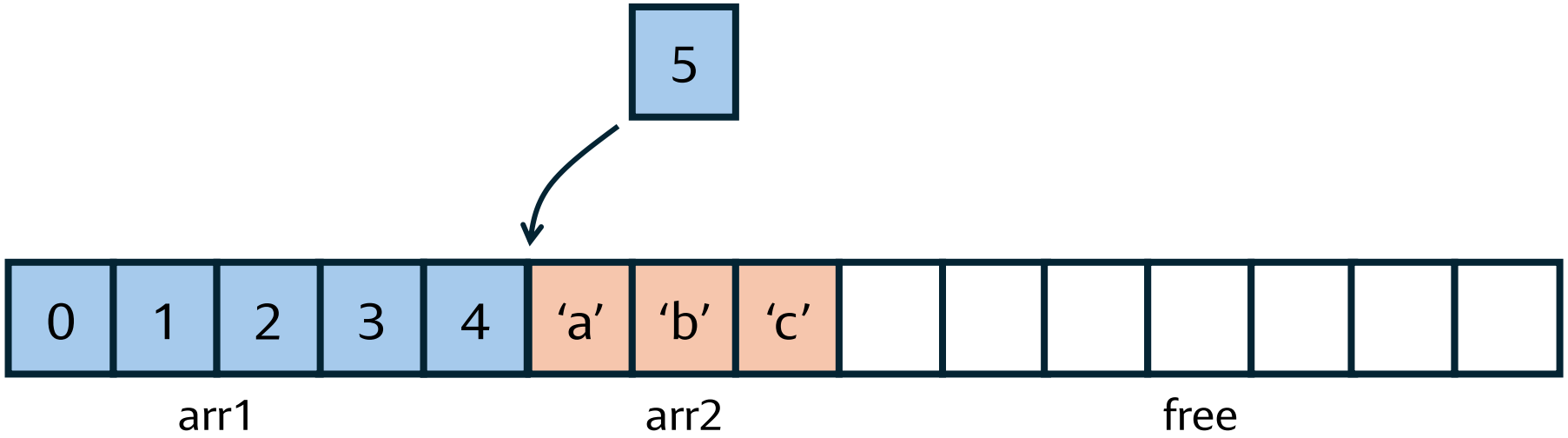
neil.zhao@utexas.edu

Notebook vs. Index Cards

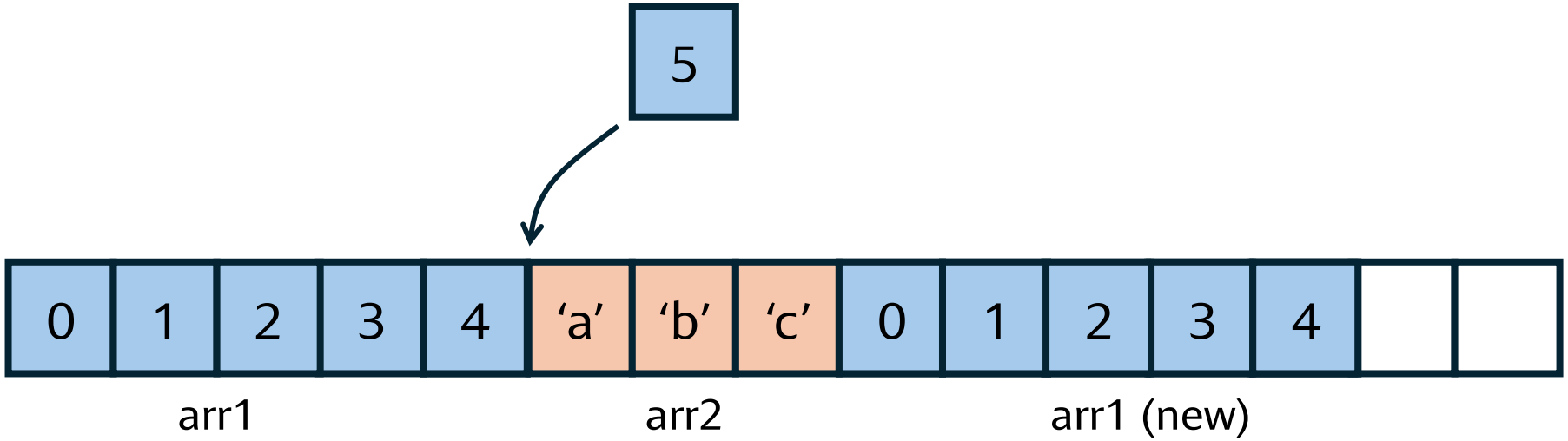
Dynamic Array



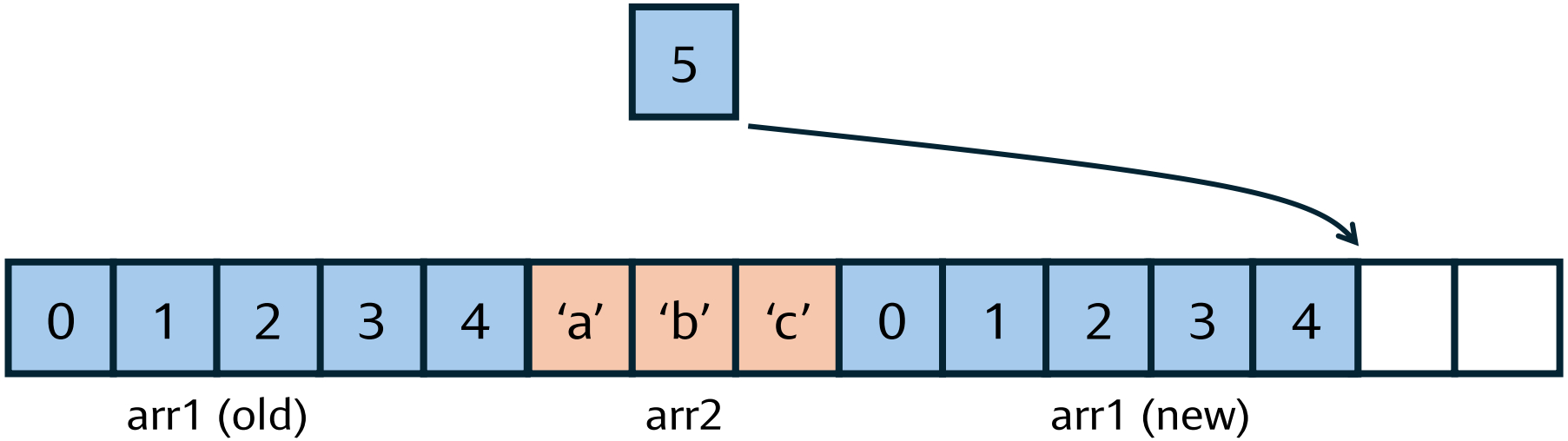
Dynamic Array



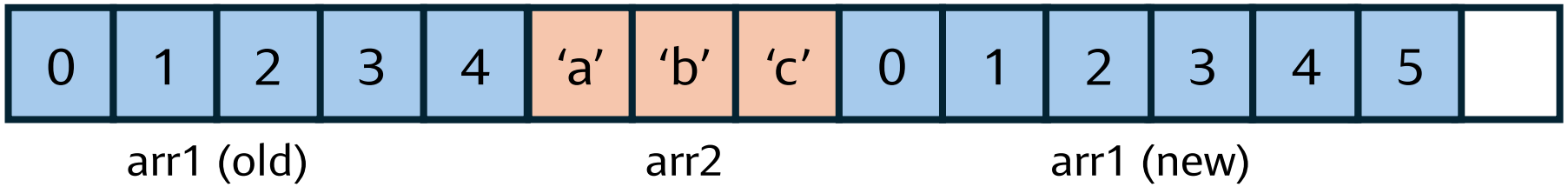
Dynamic Array



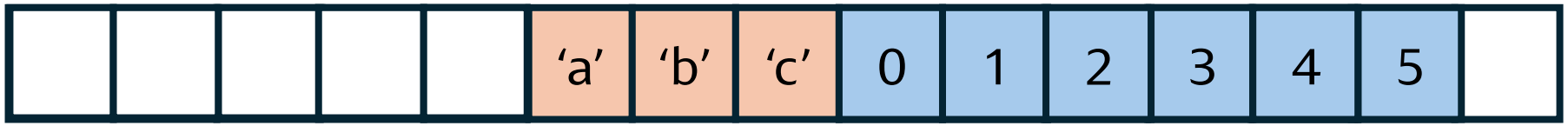
Dynamic Array



Dynamic Array

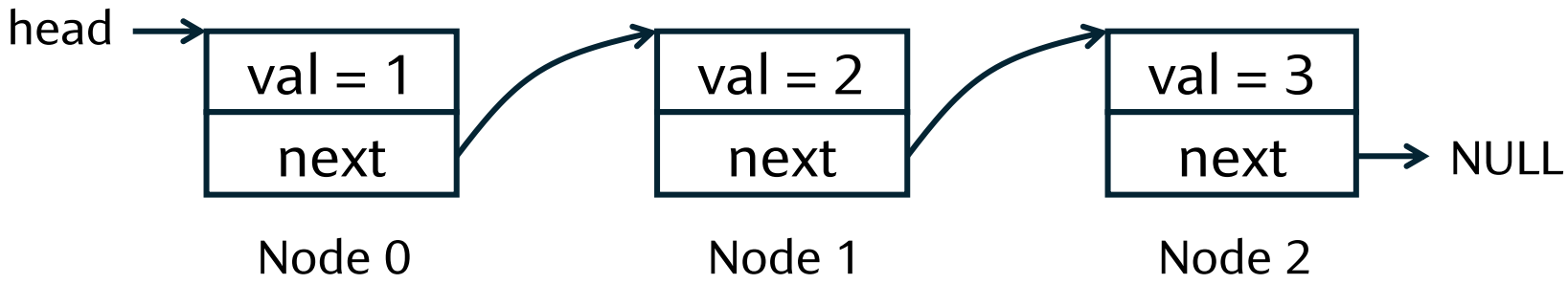


Dynamic Array



Linked List

```
typedef struct ListNode {
    int val;
    struct ListNode *next;
} ListNode;
```

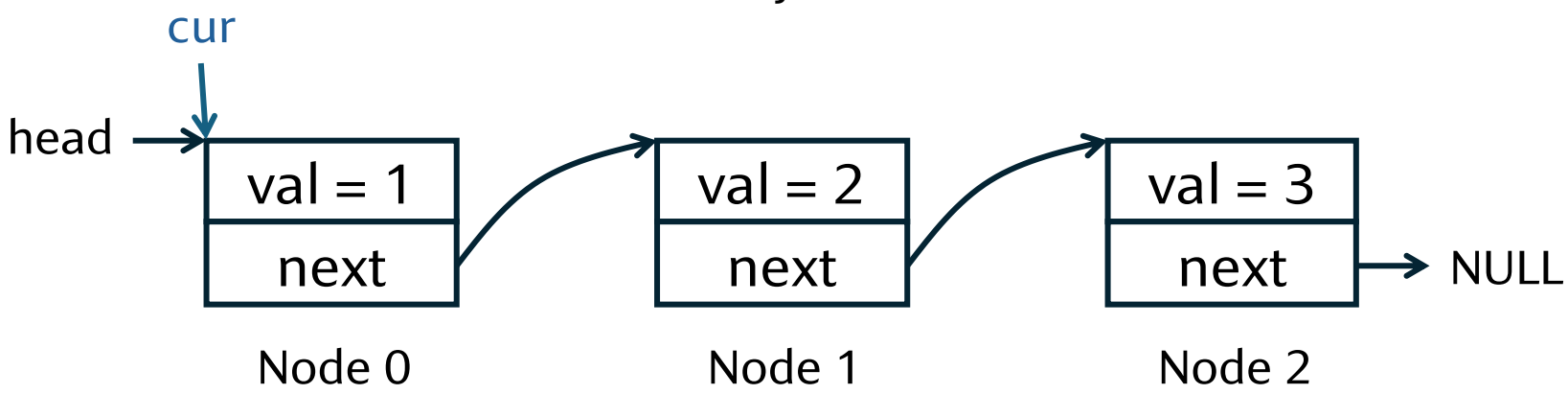


Empty list: head = NULL

Traverse Linked List

```
typedef struct ListNode {
    int val;
    struct ListNode *next;
} ListNode;
```

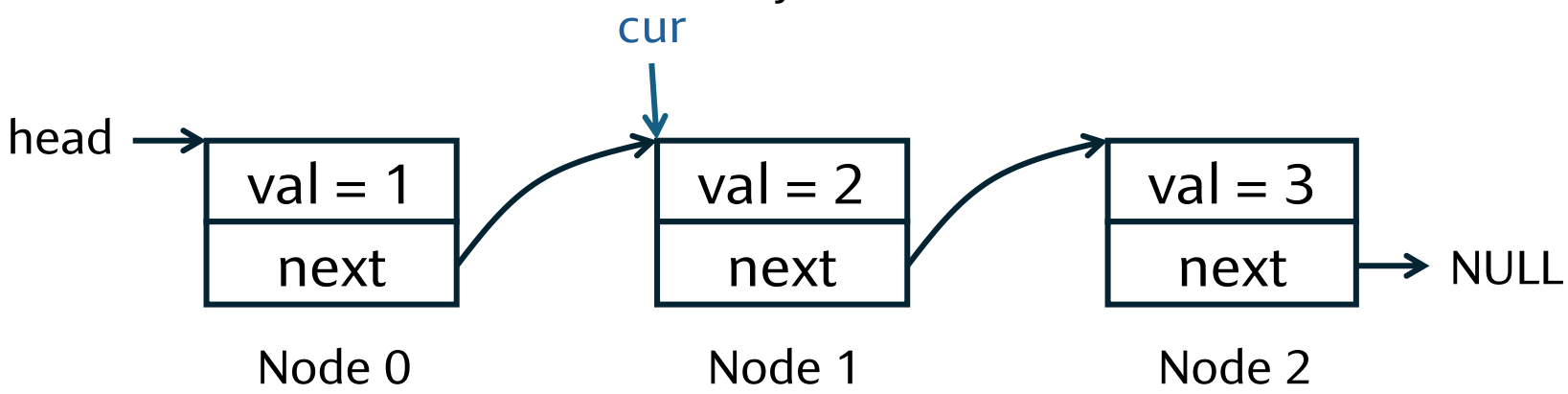
```
ListNode *cur = head;
while (cur) {
  // process *cur
  cur = cur->next;
}
```



Traverse Linked List

```
typedef struct ListNode {
    int val;
    struct ListNode *next;
} ListNode;
```

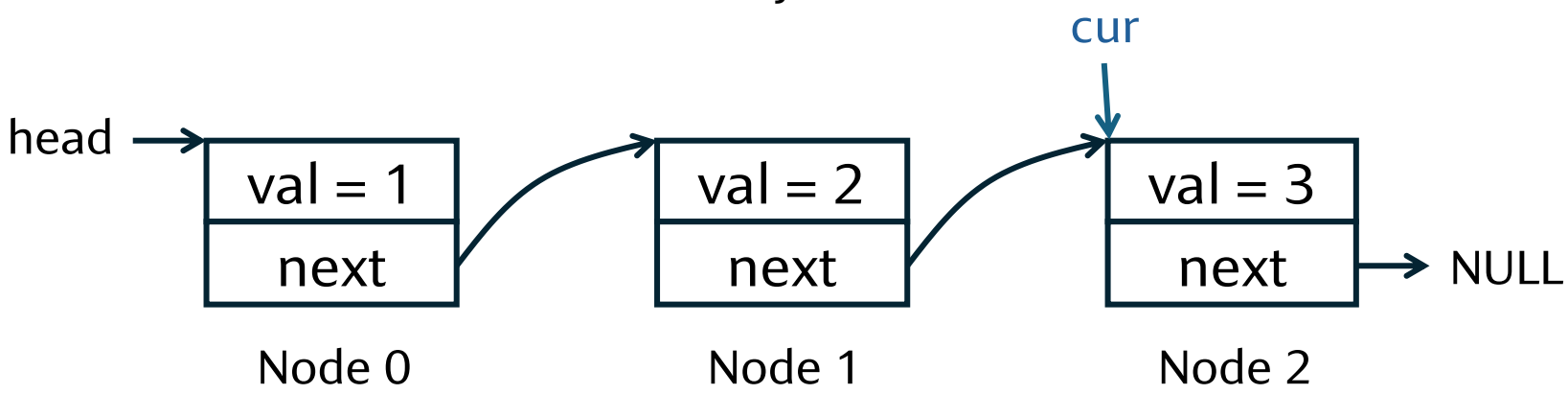
```
ListNode *cur = head;
while (cur) {
  // process *cur
  cur = cur->next;
}
```



Traverse Linked List

```
typedef struct ListNode {
    int val;
    struct ListNode *next;
} ListNode;
```

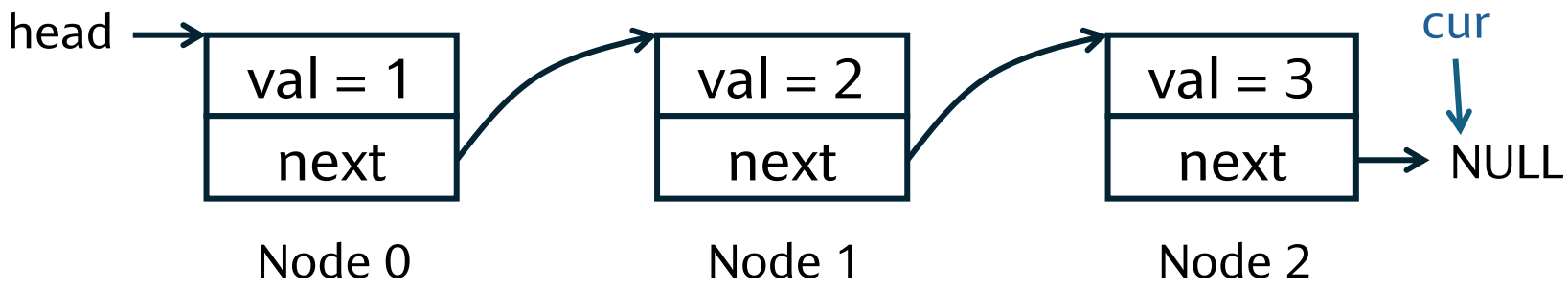
```
ListNode *cur = head;
while (cur) {
  // process *cur
  cur = cur->next;
}
```



Traverse Linked List

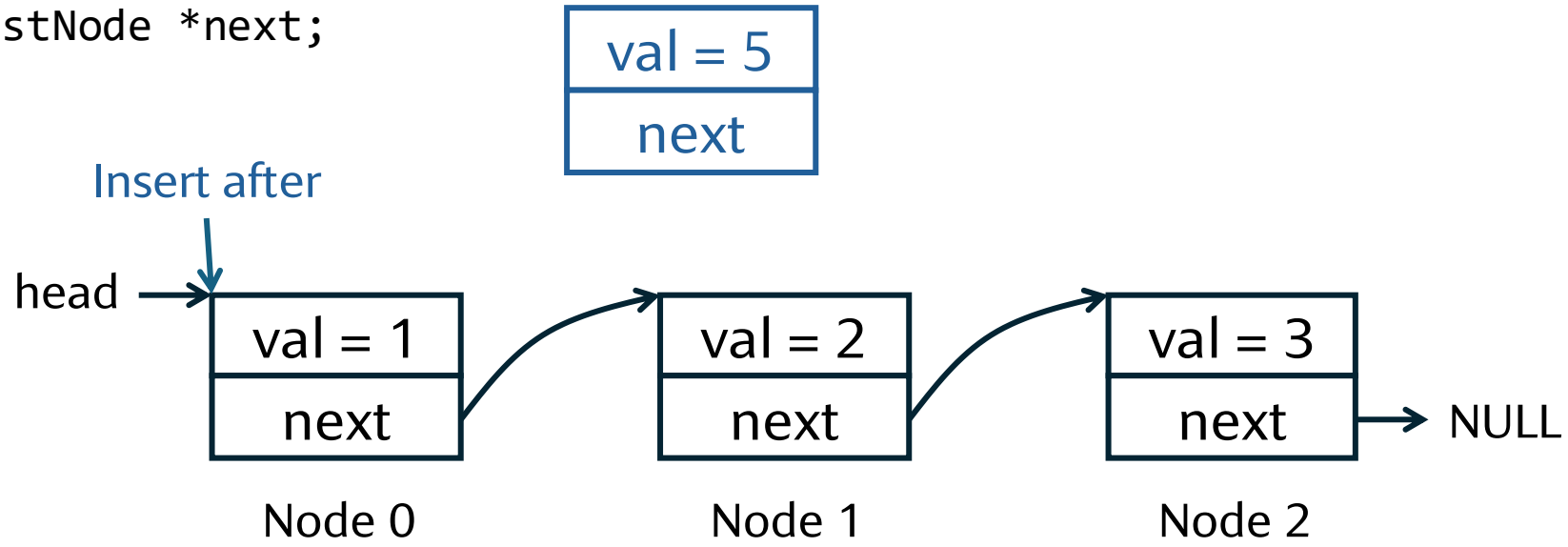
```
typedef struct ListNode {
    int val;
    struct ListNode *next;
} ListNode;
```

```
ListNode *cur = head;
while (cur) {
  // process *cur
  cur = cur->next;
}
```



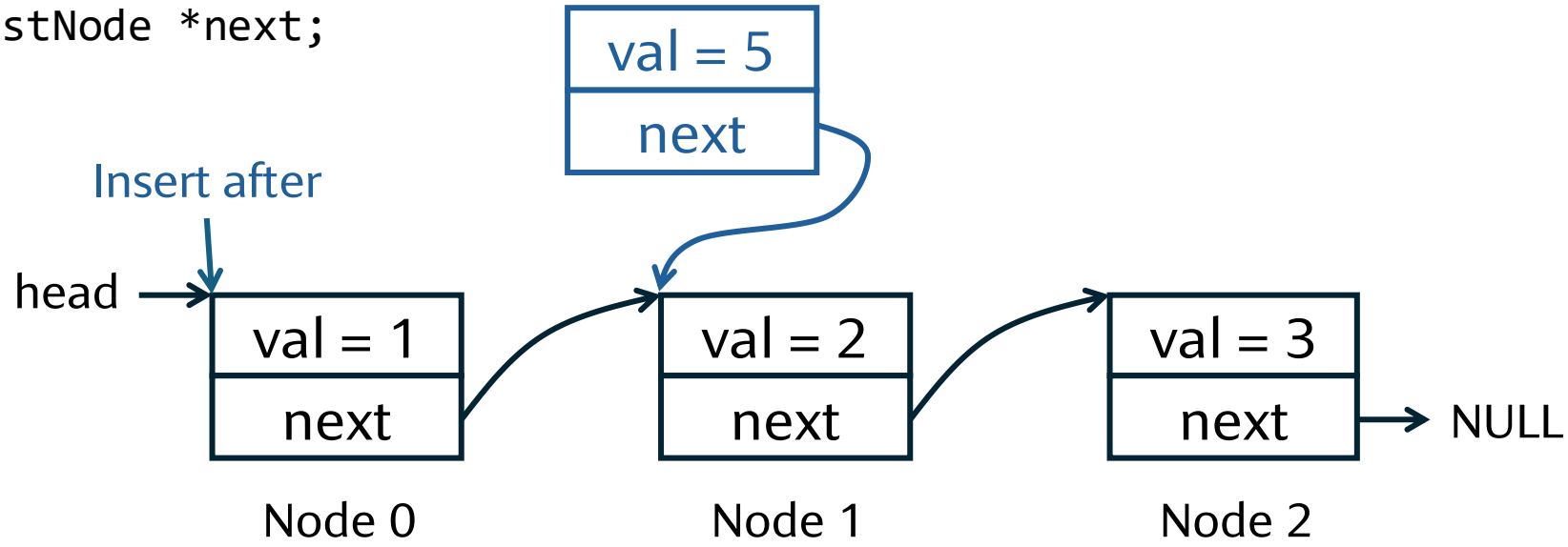
Insert After (Linked List)

```
typedef struct ListNode {
    int val;
    struct ListNode *next;
} ListNode;
```



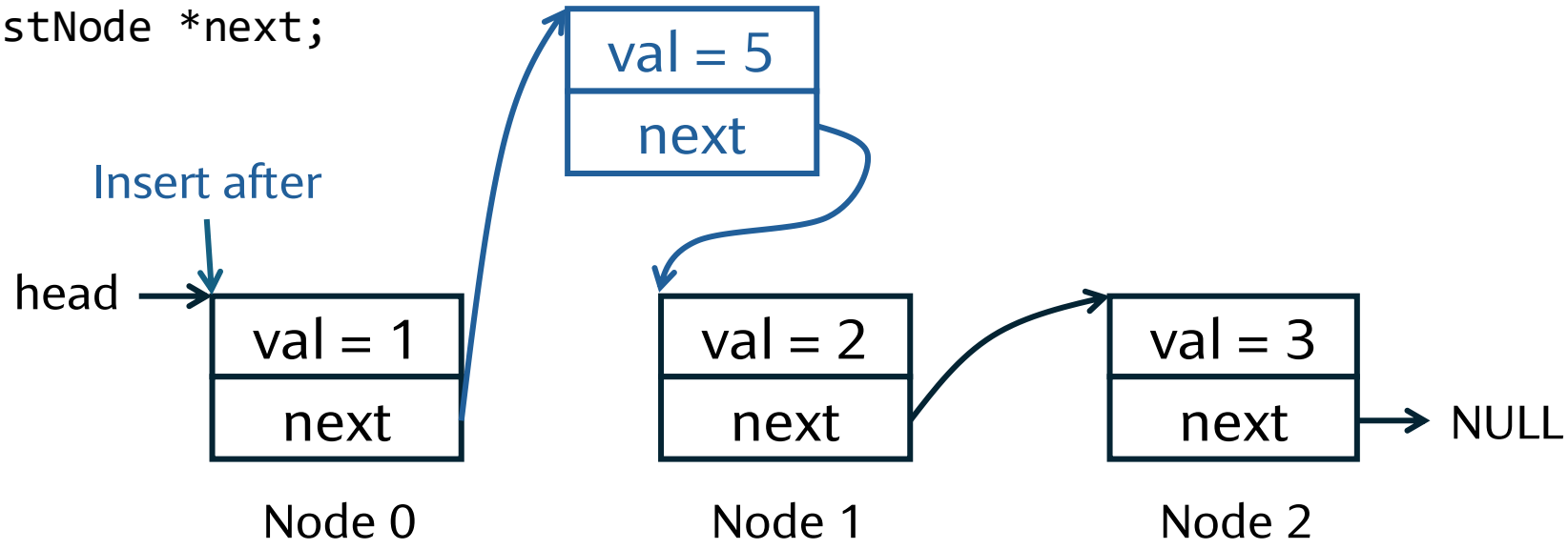
Insert After (Linked List)

```
typedef struct ListNode {
    int val;
    struct ListNode *next;
} ListNode;
```



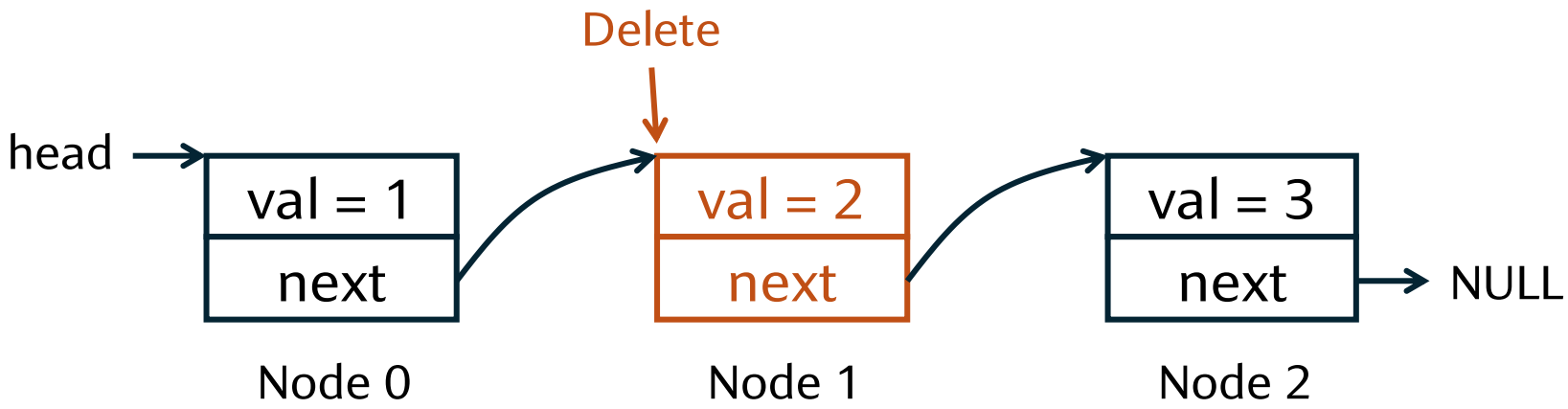
Insert After (Linked List)

```
typedef struct ListNode {
    int val;
    struct ListNode *next;
} ListNode;
```



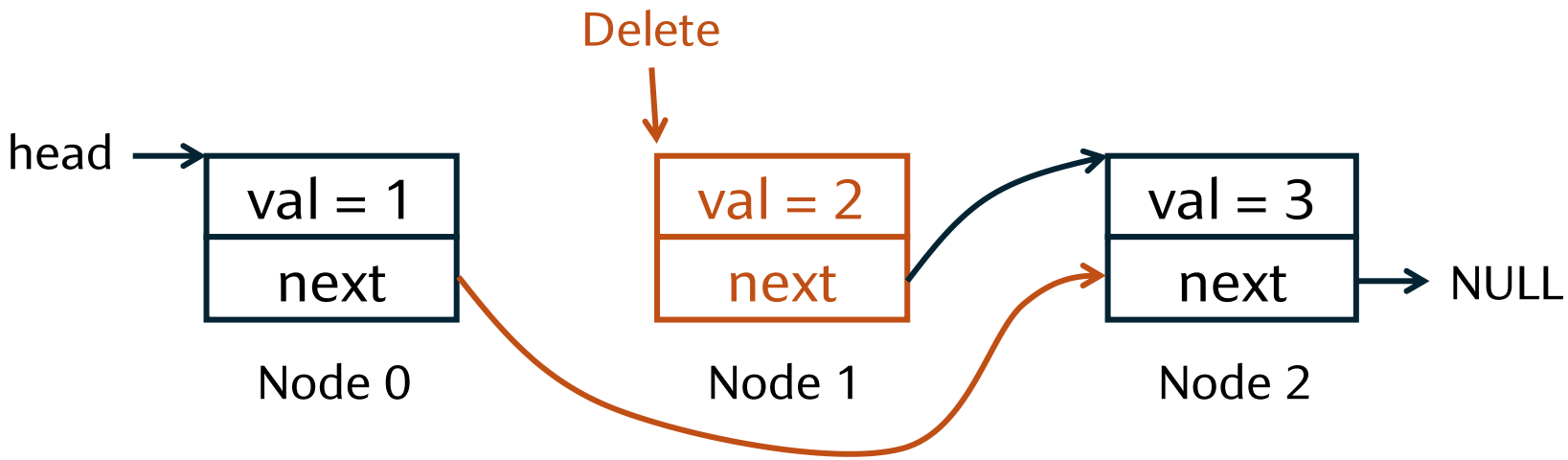
Delete Node (Linked List)

```
typedef struct ListNode {
    int val;
    struct ListNode *next;
} ListNode;
```



Delete Node (Linked List)

```
typedef struct ListNode {
    int val;
    struct ListNode *next;
} ListNode;
```



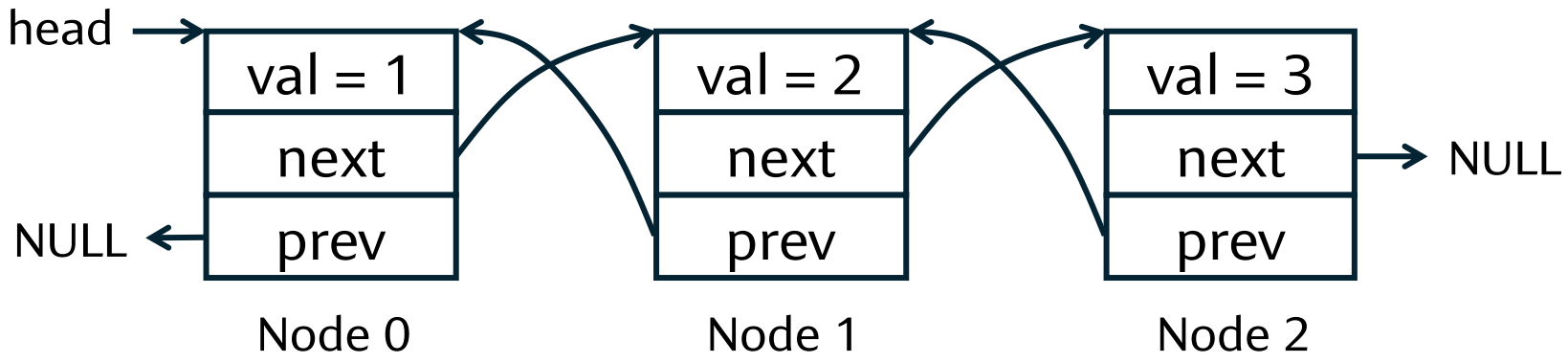
Delete Node (Linked List)

```
typedef struct ListNode {
    int val;
    struct ListNode *next;
} ListNode;
```

Note: We may need to update the head pointer if we are deleting the first element

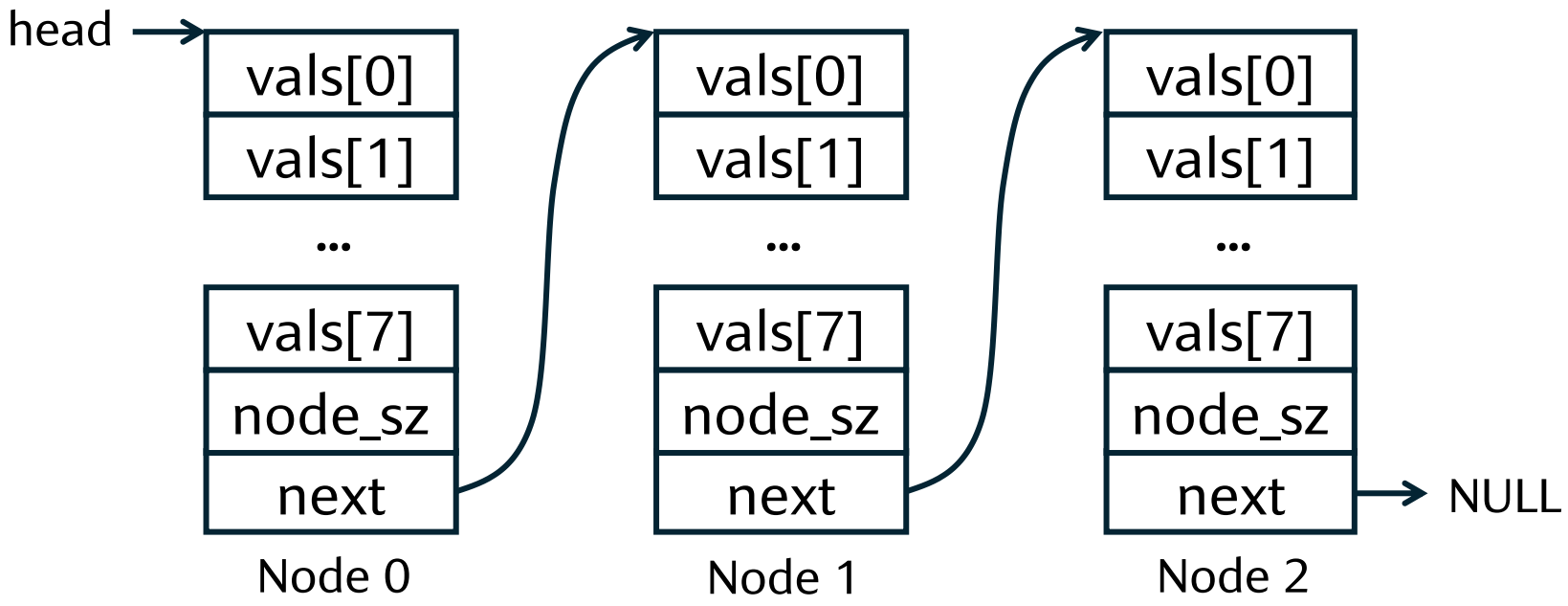
Doubly Linked List

```
typedef struct ListNode {
    int val;
    struct ListNode *next, *prev;
} ListNode;
```

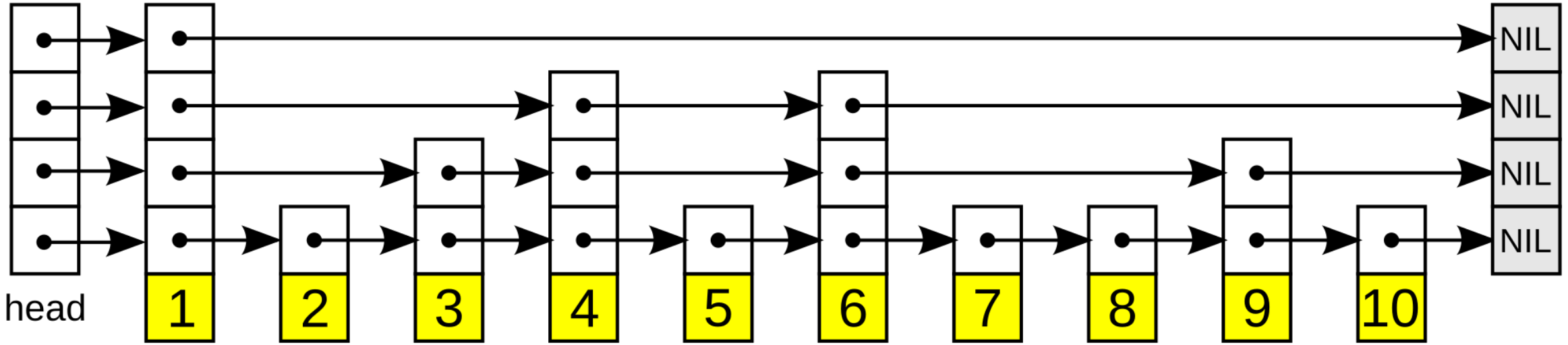


Unrolled Linked List

```
#define NODE_CAP 8
typedef struct ListNode {
    int vals[NODE_CAP];
    size_t node_sz;
    struct ListNode *next;
} ListNode;
```



Skip List



Source: https://en.wikipedia.org/wiki/Skip_list#/media/File:Skip_list.svg