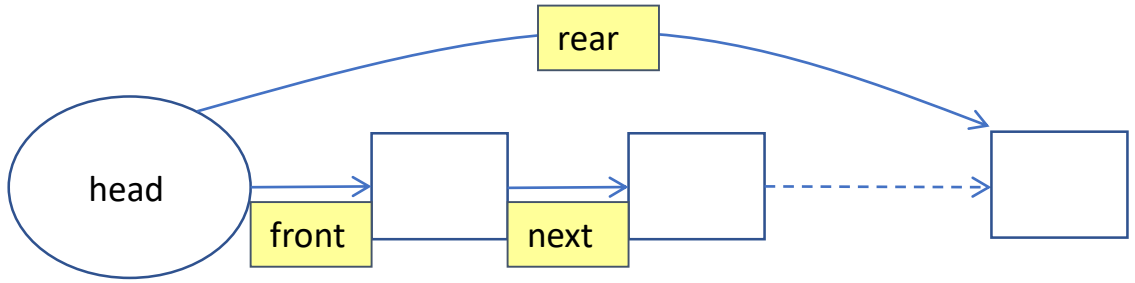
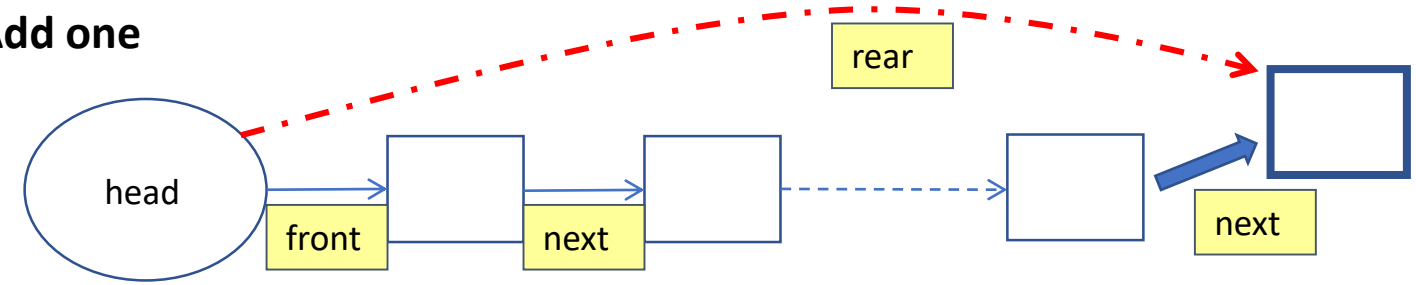
Topic 4: Linked list (remove)

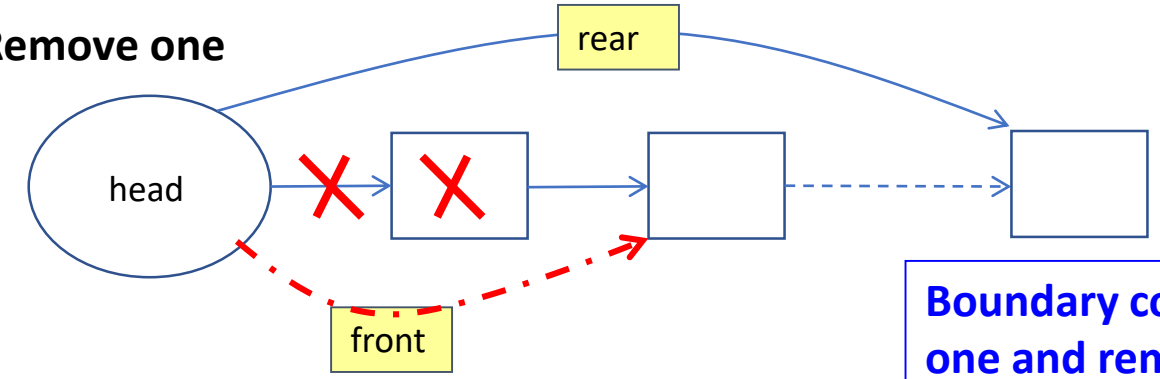
Linked list (Add to last and remove from first → FIFO)



Add one



Remove one



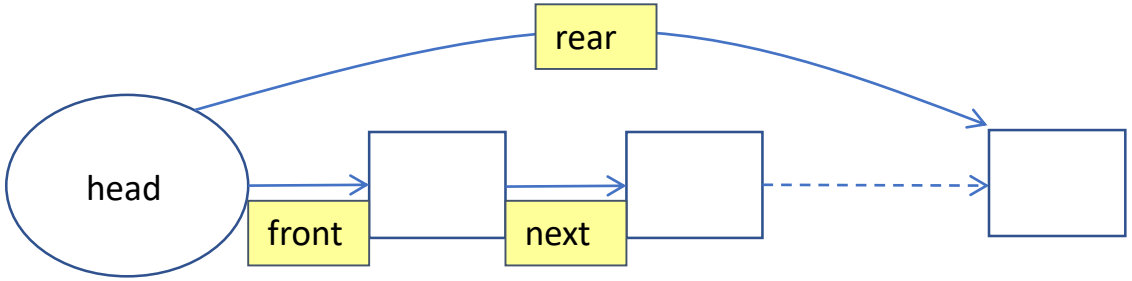
Boundary condition (Add the first one and remove the last one)

Pointer & structure & linked list remove

- Remove the first one

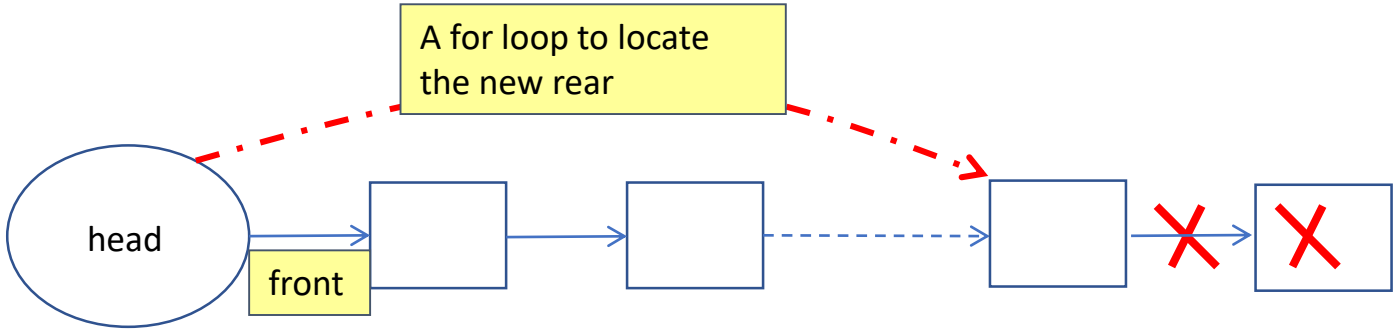
```
void RemStudent (tRegHead *p)
{
    tReg *stu_ptr;

    stu_ptr = p->front;
    p->front = stu_ptr->next;
    p->count --;

    printf ("Remove student ID: %d with score: %d \n",
                stu_ptr->ID, stu_ptr->score);

    free (stu_ptr);
}
```

Linked list (Add to last and remove from last → Stack)



Add one (same as FIFO)

Remove the last one



W6-assignment

- Based on your program last week!
- Implement a delete_last function

```
ryanpan@RyanPanPC /Volumes/MyWorks/D_Data/teaching/1
Input a number (-1 to exit, -2 to delete last): 5
  list->counts: 1
  The sorted list: 5

Input a number (-1 to exit, -2 to delete last): 10
  list->counts: 2
  The sorted list: 5 10

Input a number (-1 to exit, -2 to delete last): 20
  list->counts: 3
  The sorted list: 5 10 20

Input a number (-1 to exit, -2 to delete last): 15
  list->counts: 4
  The sorted list: 5 10 15 20

Input a number (-1 to exit, -2 to delete last): -2
  list->counts: 3
  The sorted list: 5 10 15

Input a number (-1 to exit, -2 to delete last): -2
  list->counts: 2
  The sorted list: 5 10

Input a number (-1 to exit, -2 to delete last): 15
  list->counts: 3
  The sorted list: 5 10 15

Input a number (-1 to exit, -2 to delete last): 13
  list->counts: 4
  The sorted list: 5 10 13 15

Input a number (-1 to exit, -2 to delete last): 20
  list->counts: 5
  The sorted list: 5 10 13 15 20

Input a number (-1 to exit, -2 to delete last): -2
  list->counts: 4
  The sorted list: 5 10 13 15

Input a number (-1 to exit, -2 to delete last): -2
  list->counts: 3
  The sorted list: 5 10 13

Input a number (-1 to exit, -2 to delete last): 1
  list->counts: 4
  The sorted list: 1 5 10 13

Input a number (-1 to exit, -2 to delete last): 0
  list->counts: 5
  The sorted list: 0 1 5 10 13

Input a number (-1 to exit, -2 to delete last): -2
  list->counts: 4
  The sorted list: 0 1 5 10

Input a number (-1 to exit, -2 to delete last): -2
  list->counts: 3
  The sorted list: 0 1 5

Input a number (-1 to exit, -2 to delete last): -2
  list->counts: 2
  The sorted list: 0 1

Input a number (-1 to exit, -2 to delete last): -4
  list->counts: 3
  The sorted list: -4 0 1

Input a number (-1 to exit, -2 to delete last): -2
  list->counts: 2
  The sorted list: -4 0
```

```
Input a number (-1 to exit, -2 to delete last): 1
  list->counts: 3
  The sorted list: -4 0 1

Input a number (-1 to exit, -2 to delete last): -2
  list->counts: 2
  The sorted list: -4 0

Input a number (-1 to exit, -2 to delete last): -2
  list->counts: 1
  The sorted list: -4

Input a number (-1 to exit, -2 to delete last): -2
  list->counts: 0
  The sorted list:

Input a number (-1 to exit, -2 to delete last): 9
  list->counts: 1
  The sorted list: 9

Input a number (-1 to exit, -2 to delete last): -2
  list->counts: 0
  The sorted list:

Input a number (-1 to exit, -2 to delete last): 4
  list->counts: 1
  The sorted list: 4

Input a number (-1 to exit, -2 to delete last): -2
  list->counts: 0
  The sorted list:

Input a number (-1 to exit, -2 to delete last): -2
    There is nothing to delete

Input a number (-1 to exit, -2 to delete last): -1
```

- Add a new function delete_last

```
void delete_last(tNumStorHead *list);
```

- No while loop or for loop in the delete_last function is allowed !!!
  - Think necessary modifications by yourself

```
void get_input(tNumStorHead *list)
{
    int input = 0, result;

    while (input != -1)
    {
        printf("Input a number (-1 to exit, -2 to delete last): ");
        scanf("%d", &input);
        if (input == -2)
        {
            delete_last(list);
        }
        else if (input != -1)
        {
            sort_list (list, input);
        }
    }
}
```