Data Structures

# Double Linked List – delete_at_first

Team Emertxe

# Double Linked List

Data Structure – Double Linked List -

# Analysis delete_at_first

Analysis: Logic / Cases

Flowchart

Algorithm

Code

# Analysis

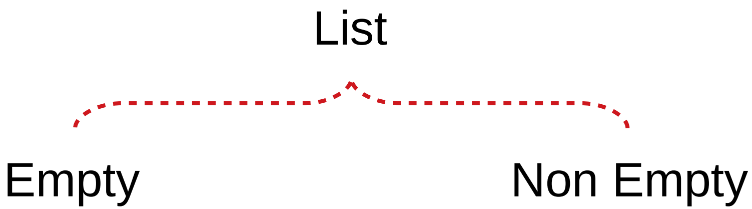
# Cases :

List

Analysis

delete_at_first

# Cases :



List

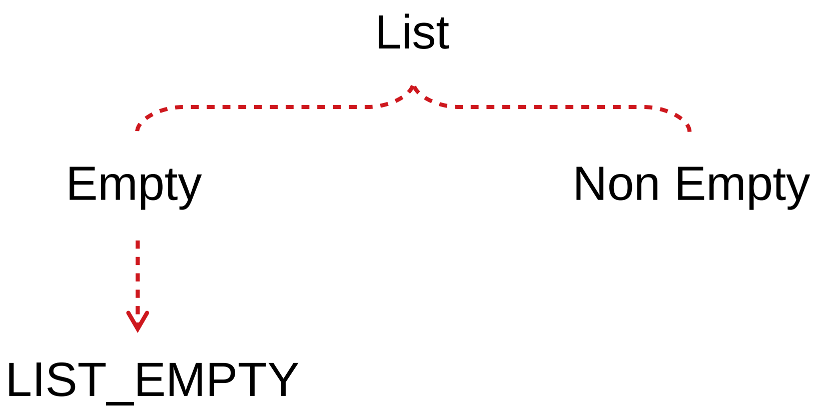
Empty

Non Empty

# Cases :

Analysis

delete_at_first



List

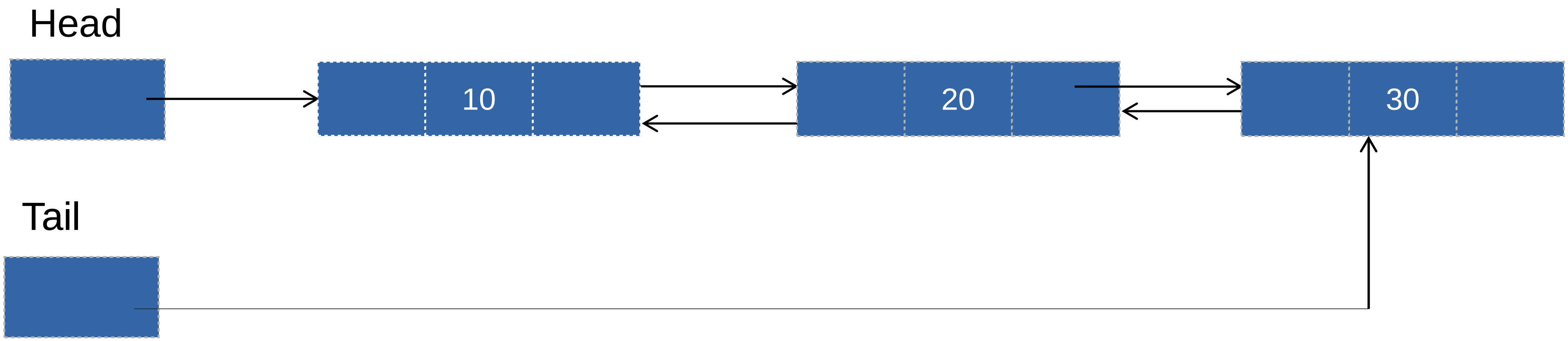
Empty

Non Empty

LIST_EMPTY

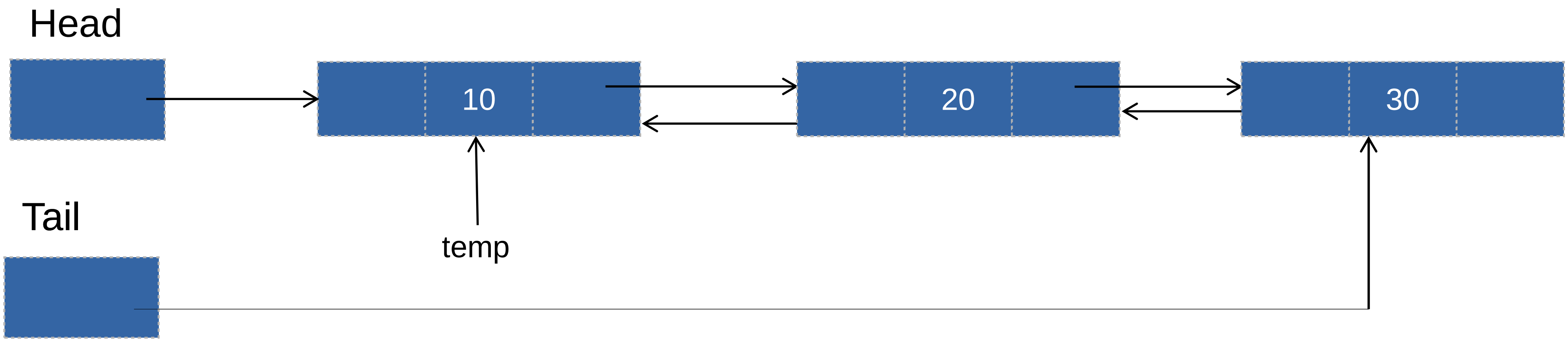
# Analysis

## delete_at_first



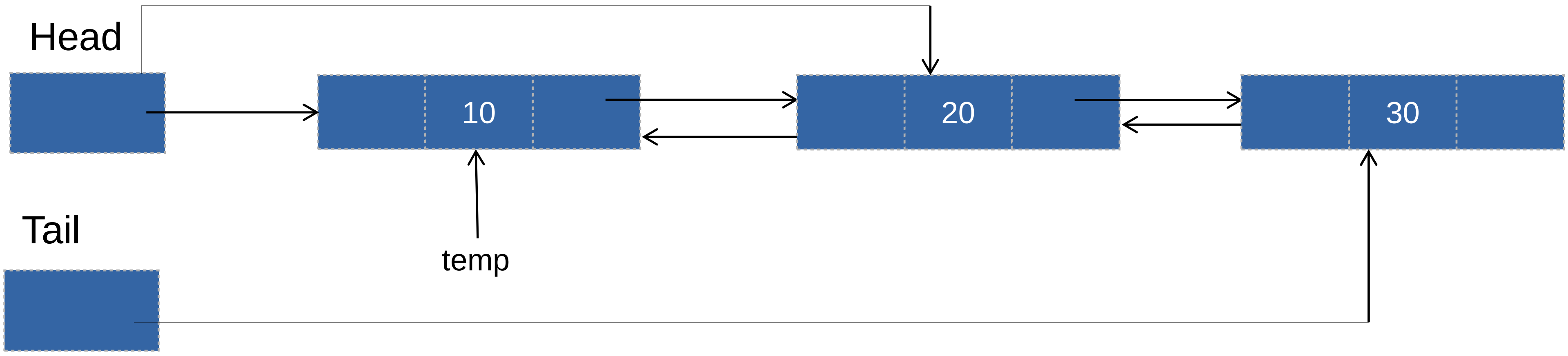
# Analysis

## delete_at_first



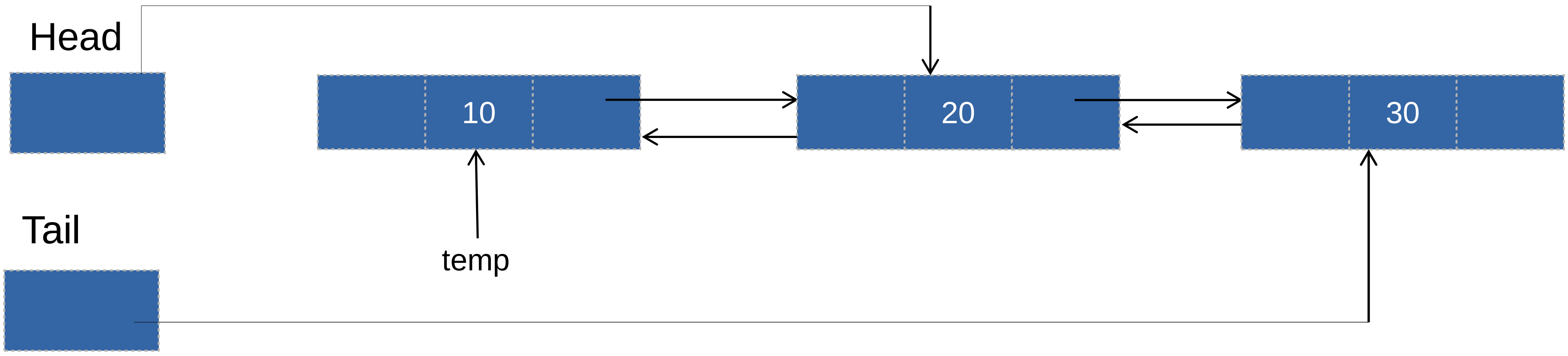
# Analysis

## delete_at_first



# Analysis

## delete_at_first



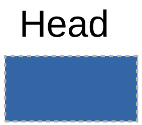
# Flowchart

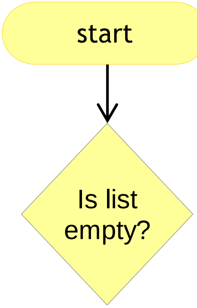
# Flowchart

## delete_at_first

# Flowchart

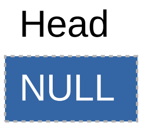
## delete_at_first

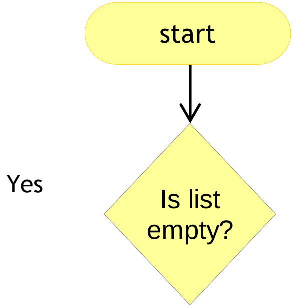




# Flowchart

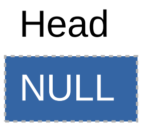
## delete_at_first

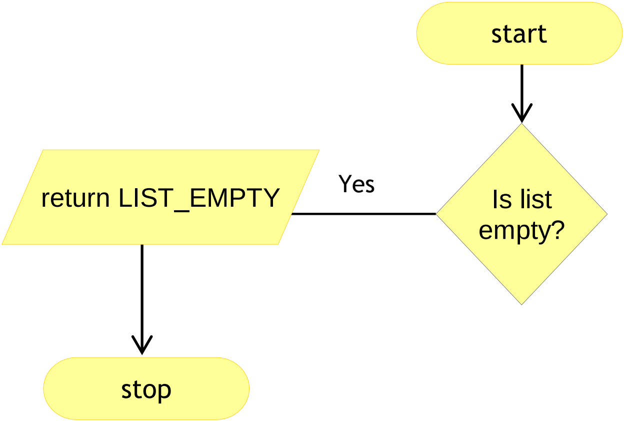




# Flowchart

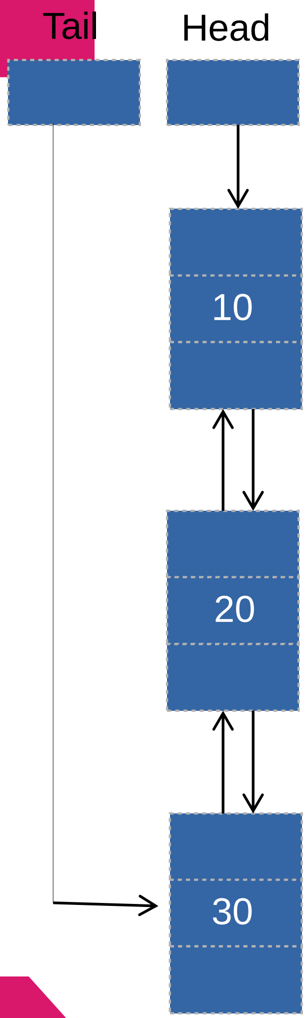
## delete_at_first

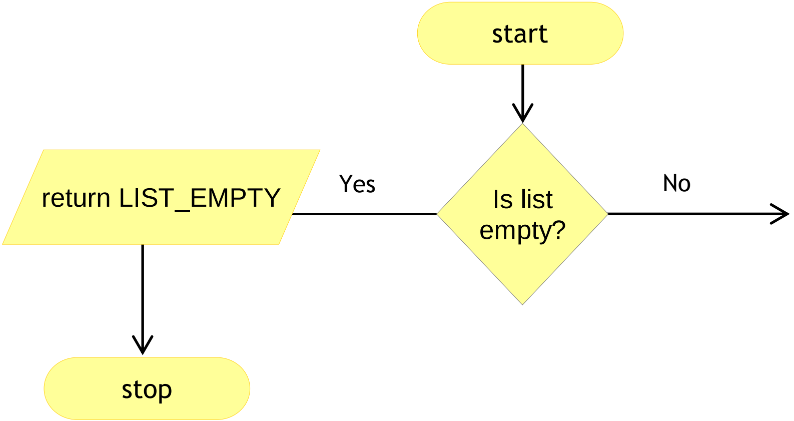




# Flowchart

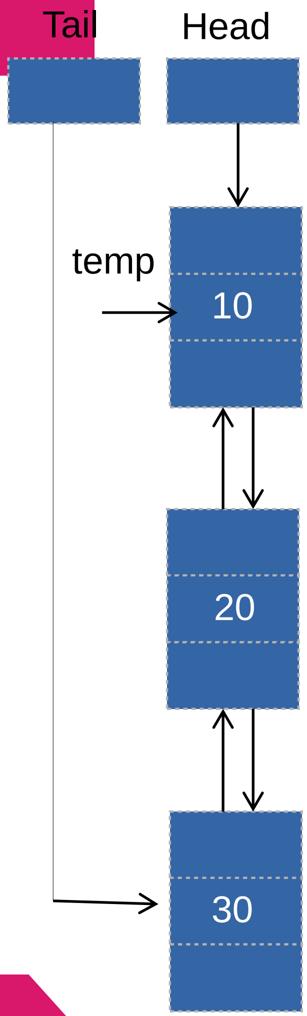
## delete_at_first

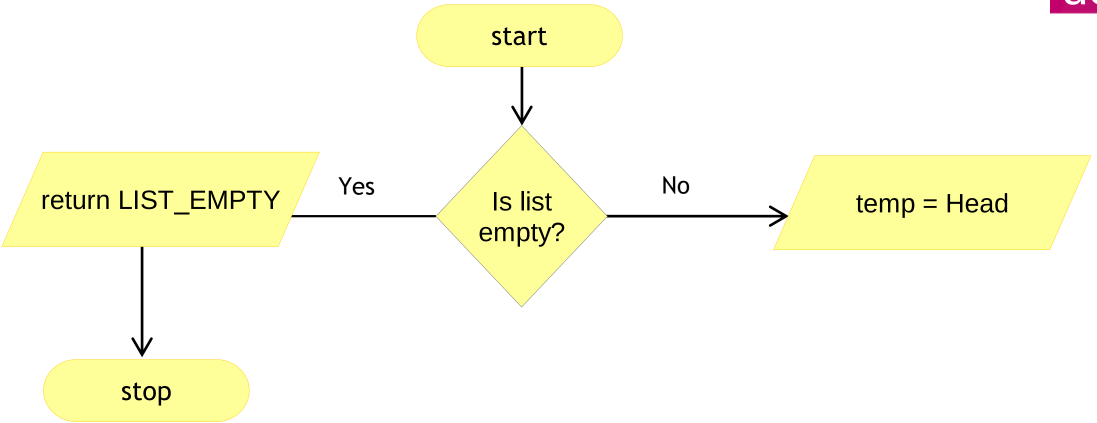




# Flowchart

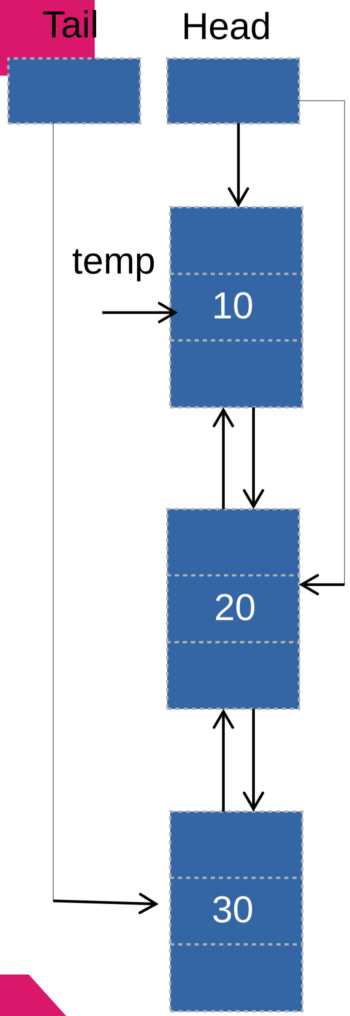
## delete_at_first

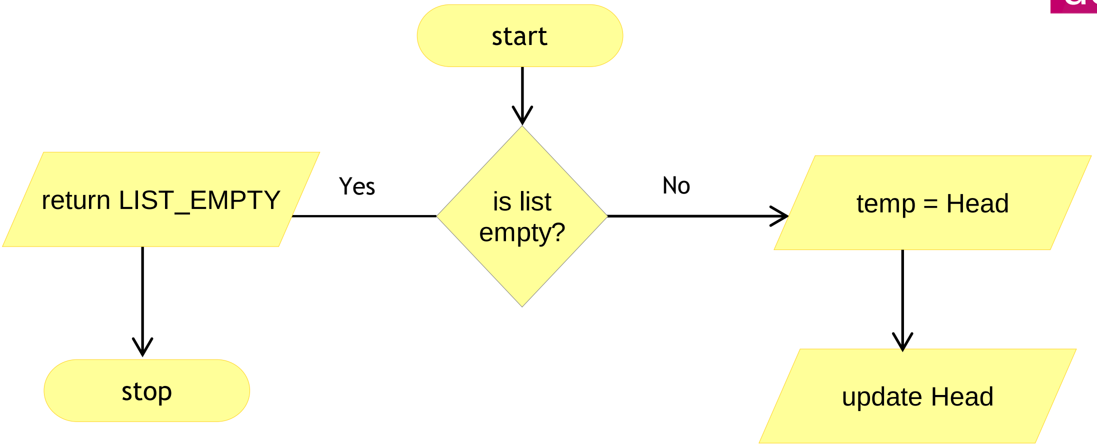




# Flowchart

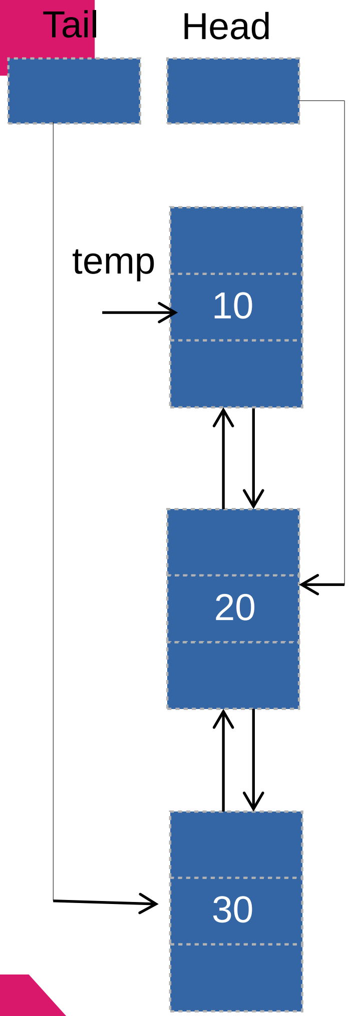
## delete_at_first

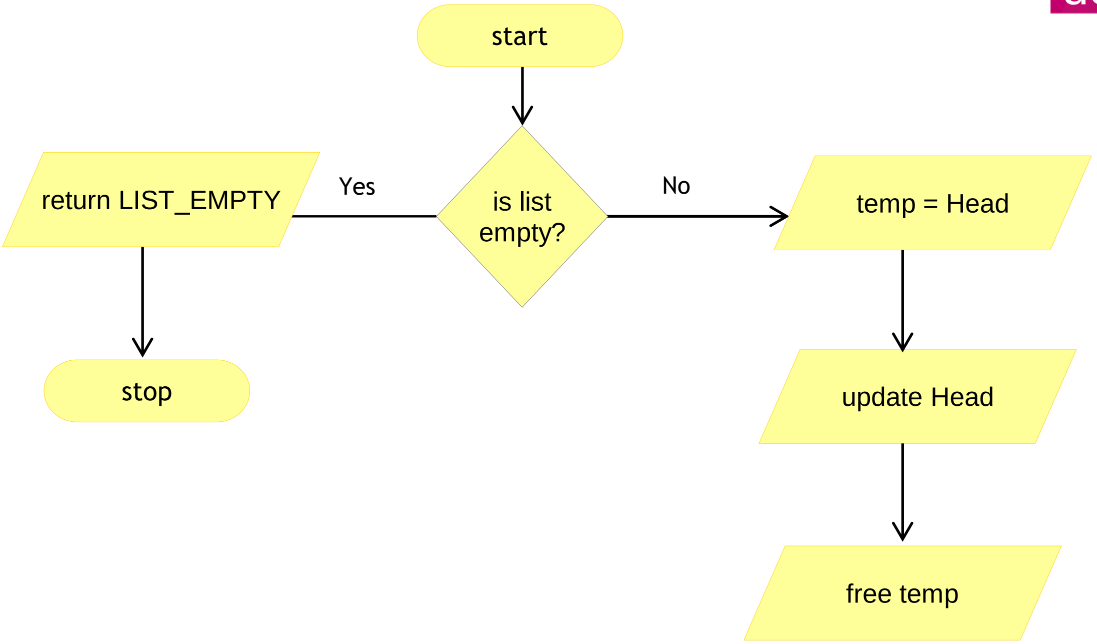




# Flowchart

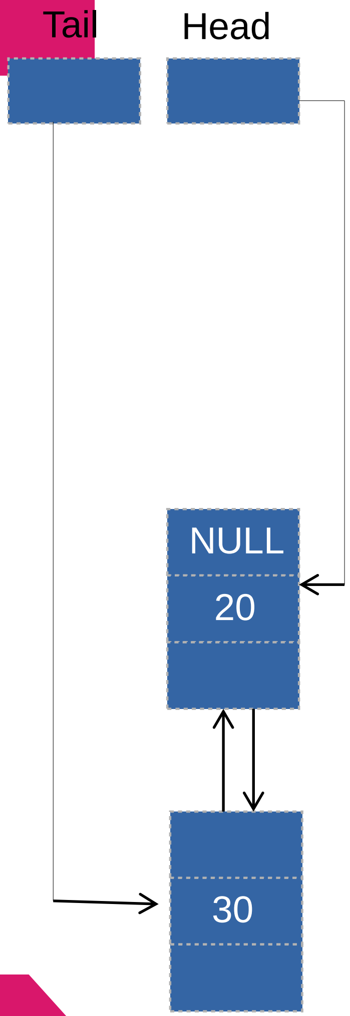
## delete_at_first

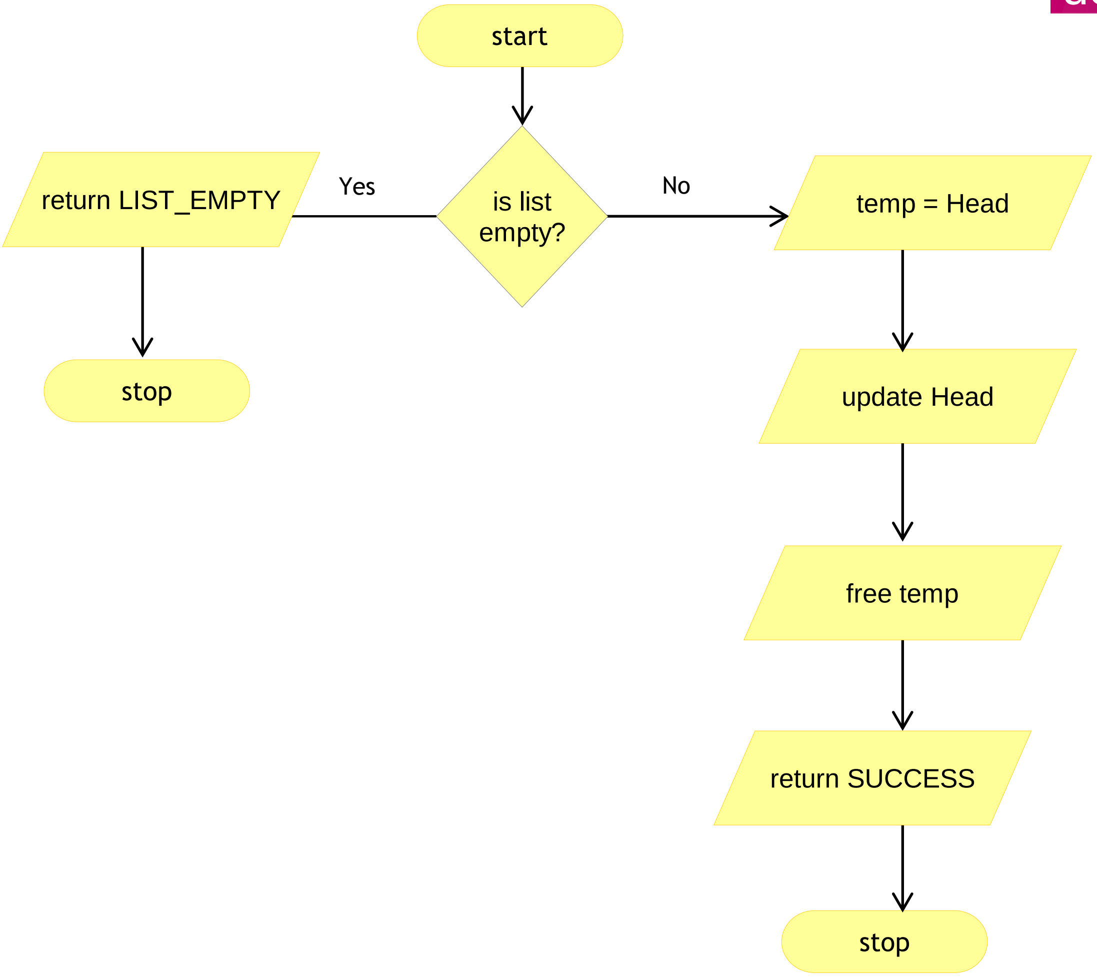




# Flowchart

## delete_at_first





start

is list empty?

Yes

return LIST_EMPTY

stop

No

temp = Head

update Head

free temp

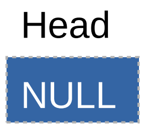
return SUCCESS

stop

# Algorithm

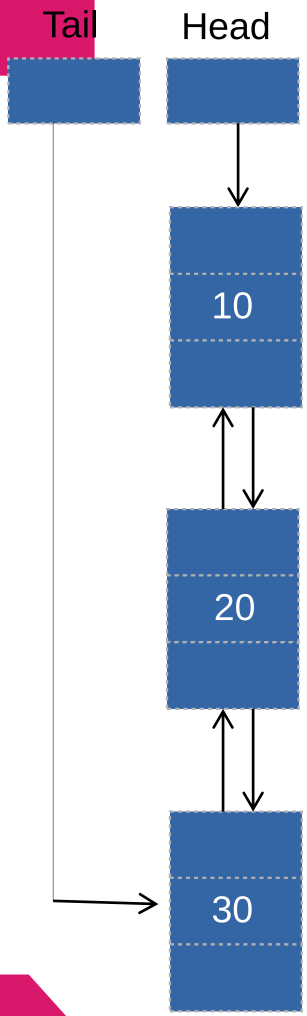
## Algorithm : delete_at_first(Head)

```
if (Head = NULL)
```



## Algorithm : delete_at_first(Head)

```
if (Head = NULL)
            return LIST_EMPTY
```

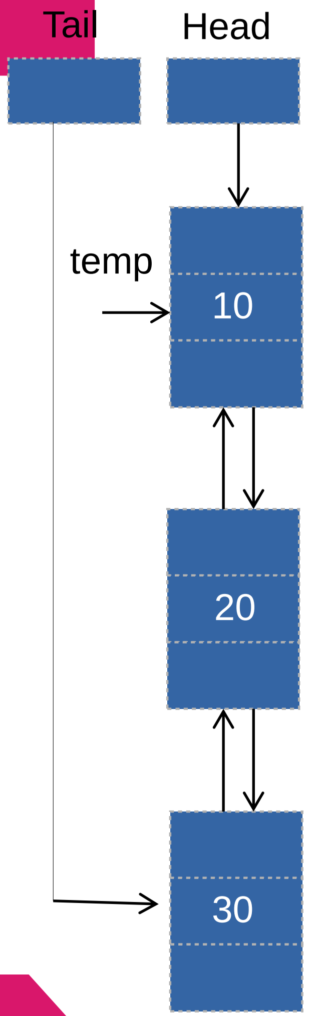


## Algorithm : delete_at_first(Head)

```
if (Head = NULL)
            return LIST_EMPTY
```

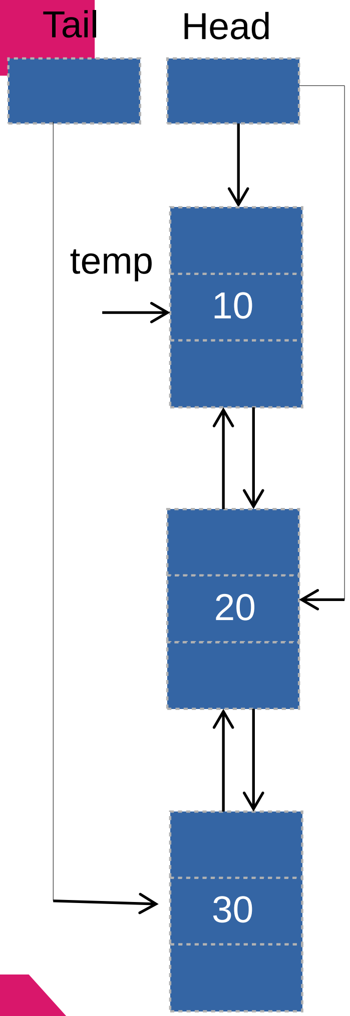
Algorithm

delete at first



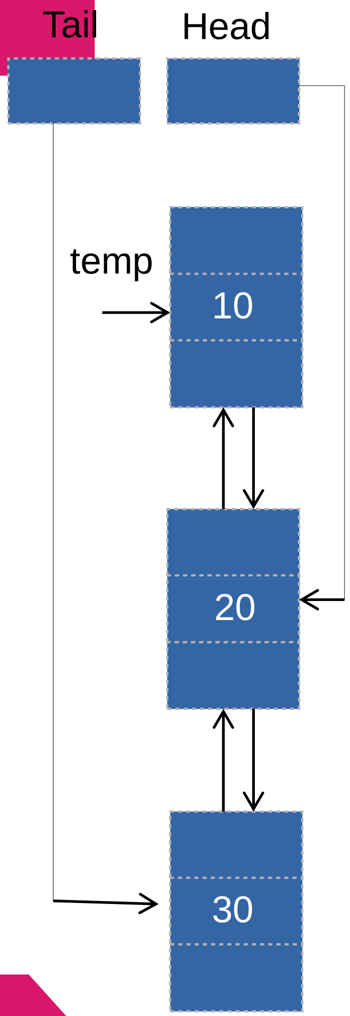
**Algorithm : delete_at_first(Head)**

```
if (Head = NULL)
                return LIST_EMPTY
temp = Head
```



## Algorithm : delete_at_first(Head)

```
if (Head = NULL)
        return LIST_EMPTY
temp = Head
Head = temp ⟶ next
```

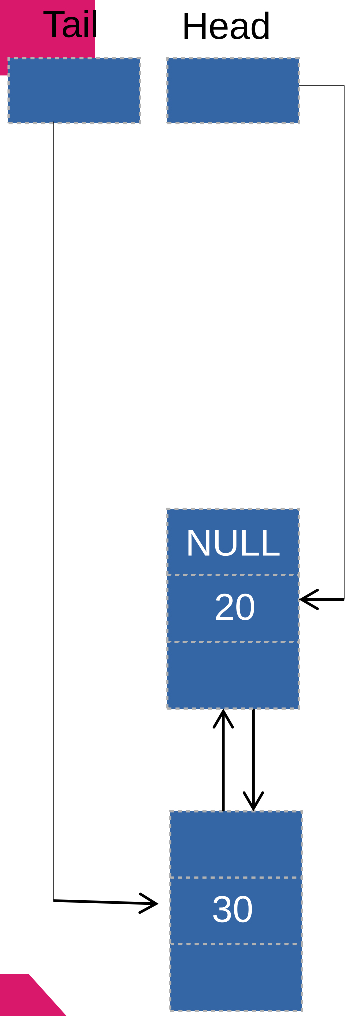


## Algorithm : delete_at_first(Head)

```
if (Head = NULL)
            return LIST_EMPTY
temp = Head
Head = temp→next
free temp
```

# Algorithm

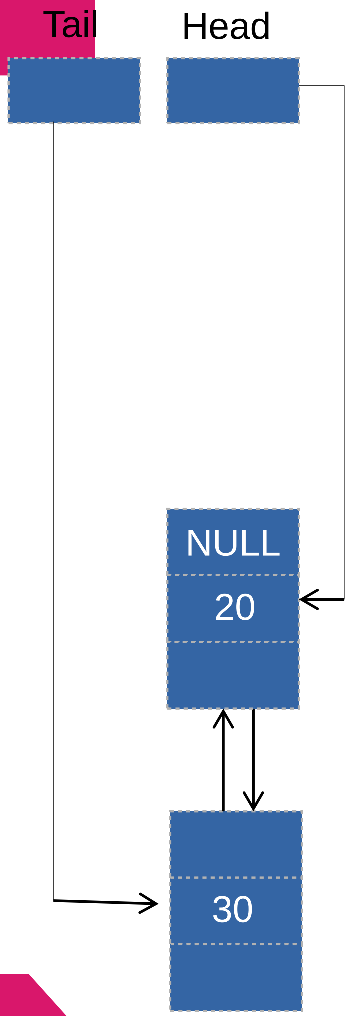
delete at first



**Algorithm : delete_at_first(Head)**

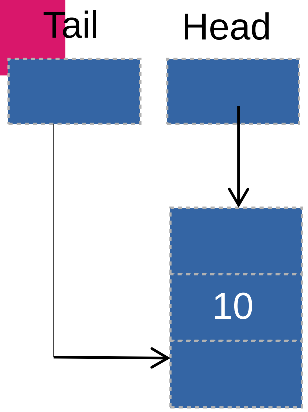
```
if (Head = NULL)
        return LIST_EMPTY
temp = Head
Head = temp ⟶ next
free temp
Head ⟶ prev =  NULL
```

# Algorithm
delete at first



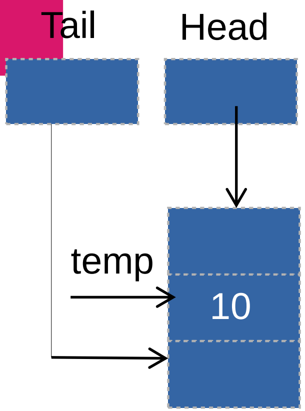
## Algorithm : delete_at_first(Head)

```
if (Head = NULL)
        return LIST_EMPTY
temp = Head
Head = temp→next
free temp
Head→prev =  NULL
return SUCCESS
```



## Algorithm : delete_at_first(Head)

```
if (Head = NULL)
              return LIST_EMPTY
temp = Head
Head = temp → next
free temp
Head → prev = NULL
return SUCCESS
```

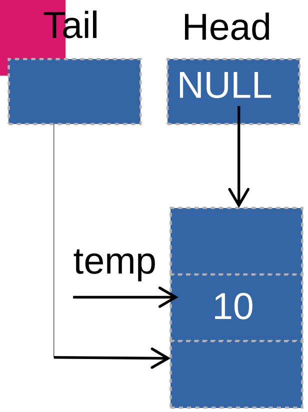


## Algorithm : delete_at_first(Head)

```
if (Head = NULL)
            return LIST_EMPTY
temp = Head
Head = temp→next
free temp
Head→prev =  NULL
return SUCCESS
```

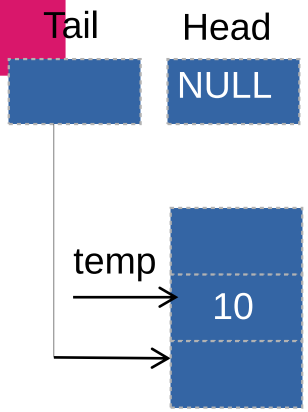
Algorithm

delete at first



## Algorithm : delete_at_first(Head)

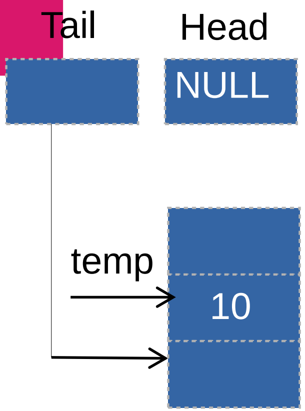
```
if (Head = NULL)
            return LIST_EMPTY
temp = Head
Head = temp → next
free temp
Head → prev =  NULL
return SUCCESS
```



## Algorithm : delete_at_first(Head)

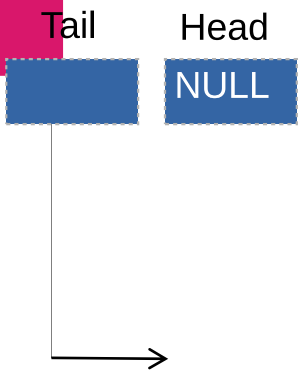
```
if (Head = NULL)
        return LIST_EMPTY
temp = Head
Head = temp→next
free temp
Head→prev = NULL
return SUCCESS
```

Algorithm
delete at first



## Algorithm : delete_at_first(Head)

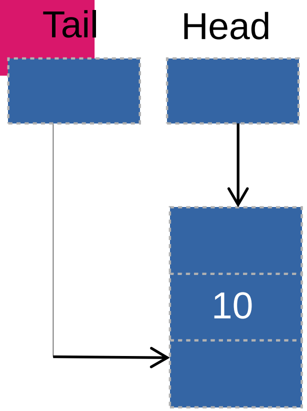
```
if ( Head = NULL)
        return LIST_EMPTY
temp = Head
Head = temp→next
free temp
Head→prev =  NULL
return SUCCESS
```



## Algorithm
delete at first

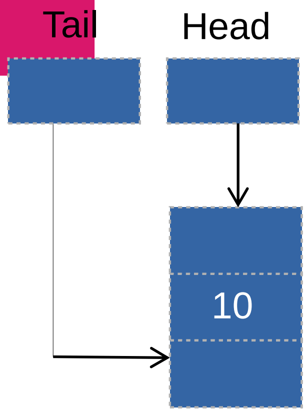
### Algorithm : delete_at_first(Head)

```
if ( Head = NULL)
        return LIST_EMPTY
temp = Head
Head  =  temp ⟶ next
free temp
Head ⟶ prev =  NULL
return SUCCESS
```



## Algorithm : delete_at_first(Head,Tail)

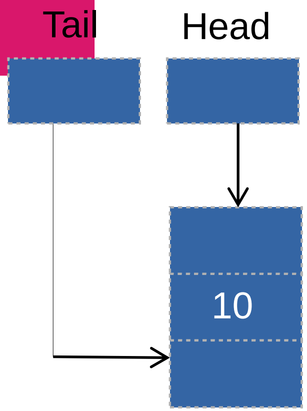
```
if (Head = NULL)
  return LIST_EMPTY
if (Head → next = NULL)
             free Head
             Head = Tail = NULL
             return SUCCESS
temp = Head
Head = temp → next
free temp
Head → prev = NULL
return SUCCESS
```



## Algorithm : delete_at_first(Head,Tail)

```
if ( Head = NULL)
  return LIST_EMPTY
if ( Head → next = NULL)
        free Head
        Head = Tail = NULL
        return SUCCESS
temp = Head
Head = temp → next
free temp
Head → prev = NULL
return SUCCESS
```

Algorithm
delete at first

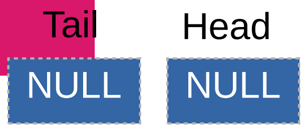


## Algorithm : delete_at_first(Head,Tail)

```
if ( Head = NULL)
  return LIST_EMPTY
if ( Head → next = NULL)
        free Head
        Head = Tail = NULL
        return SUCCESS
temp = Head
Head = temp → next
free temp
Head → prev = NULL
return SUCCESS
```

Algorithm

delete at first

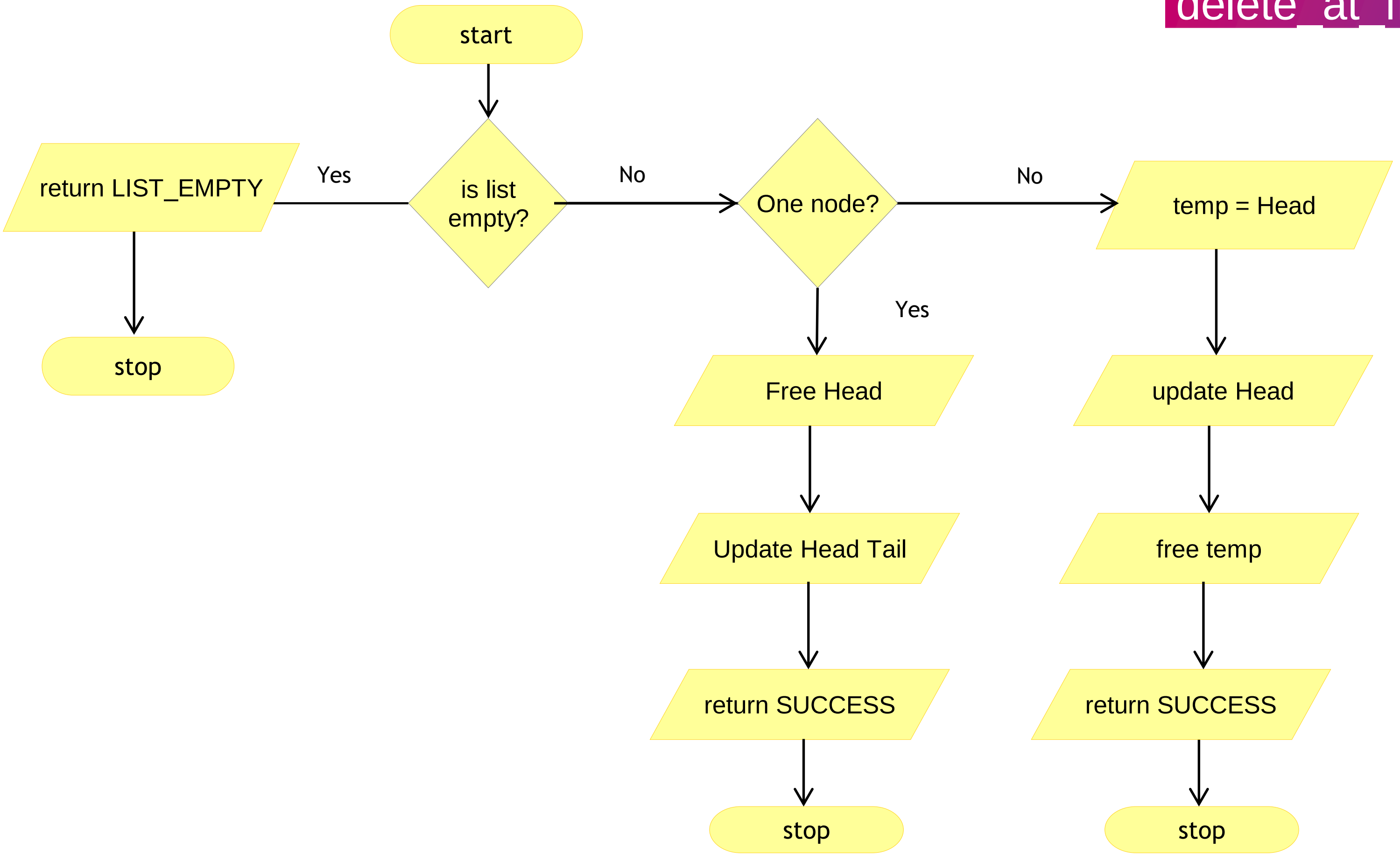


## Algorithm : delete_at_first(Head,Tail)

```
if ( Head = NULL)
  return LIST_EMPTY
if ( Head → next = NULL)
              free Head
              Head = Tail = NULL
              return SUCCESS
temp = Head
Head = temp → next
free temp
Head → prev = NULL
return SUCCESS
```

# Flowchart

## delete_at_first



# Double Linked List – delete_at_first -Code