

Data Structures

Circular Queue– Linked list

Team Emertxe



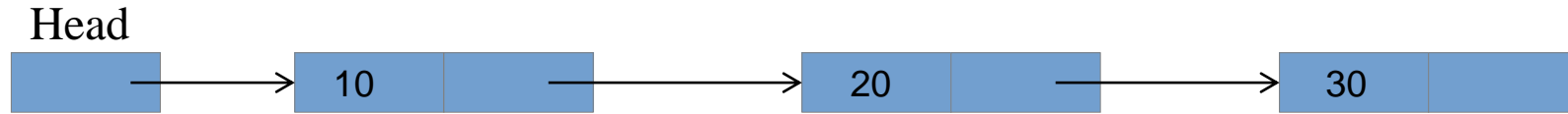
Circular Queue -Linked List



Linked List

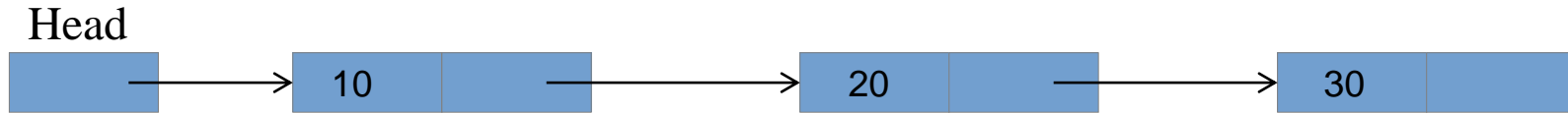


Single Linked List

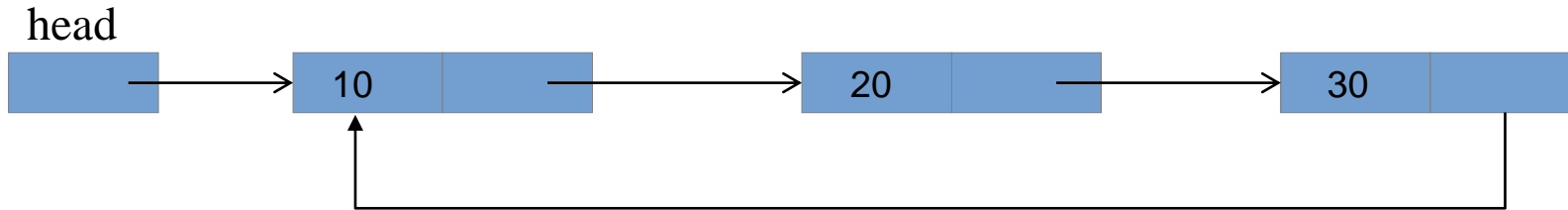


Linked List

Single Linked List



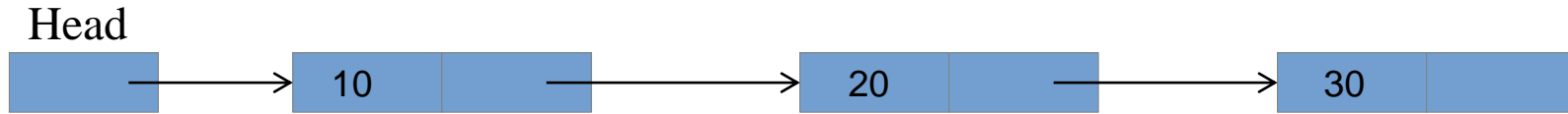
Circular Linked List



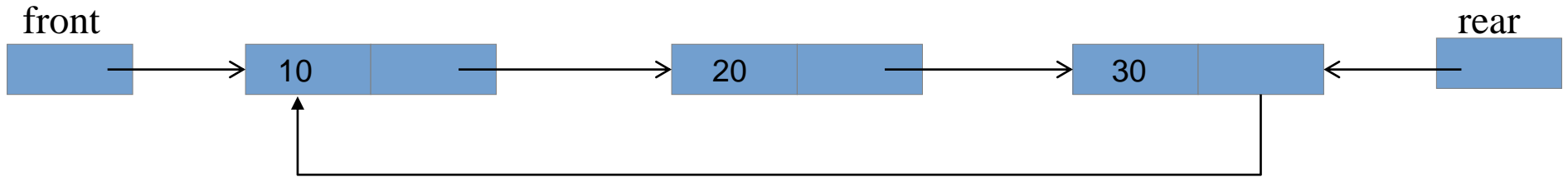
Circular Linked list is a list where all nodes are connected to form a circle

Linked List

Single Linked List



Circular Queue



Circular Linked list is a list where all nodes are connected to form a circle

enqueue

Algorithm :enqueue(front,rear,data)

Input Specification :

- .front : Pointer contains the first node address
- .rear : Pointer contains the last node address
- .data : Item to be added

Output Specification :

- .Status: e_true/e_false

enqueue

```
new = Memalloc(sizeof(Slist))
if (new = NULL)
    return e_false
new → data = data
if(front = NULL)
    front = rear = new
else
    rear → link = new
    rear = new
new → link = front
return e_true
```

Algorithm

```
new = Memalloc(sizeof(Slist))
```

```
if (new = NULL)
```

```
    return e_false
```

```
new → data = data
```

```
if(front = NULL)
```

```
    front = rear = new
```

```
else
```

```
    rear → link = new
```

```
    rear = new
```

```
new → link = front
```

```
return e_true
```

new



Algorithm

```
new = Memalloc(sizeof(Slist))
```

```
if (new = NULL)
```

```
    return e_false
```

```
new → data = data
```

```
if(front = NULL)
```

```
    front = rear = new
```

```
else
```

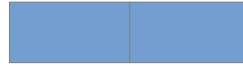
```
    rear → link = new
```

```
    rear = new
```

```
new → link = front
```

```
return e_true
```

new



Algorithm

```
new = Memalloc(sizeof(Slist))
```

```
if (new = NULL)
```

```
    return e_false
```

```
new → data = data
```

```
if(front = NULL)
```

```
    front = rear = new
```

```
else
```

```
    rear → link = new
```

```
    rear = new
```

```
new → link = front
```

```
return e_true
```

new

10	
----	--

Algorithm

```
new = Memalloc(sizeof(Slist))
```

```
if (new = NULL)
```

```
    return e_false
```

```
new → data = data
```

```
if(front = NULL)
```

```
    front = rear = new
```

```
else
```

```
    rear → link = new
```

```
    rear = new
```

```
new → link = front
```

```
return e_true
```

front

NULL

new

10

Algorithm

```
new = Memalloc(sizeof(Slist))
```

```
if (new = NULL)
```

```
    return e_false
```

```
new → data = data
```

```
if(front = NULL)
```

```
    front = rear = new
```

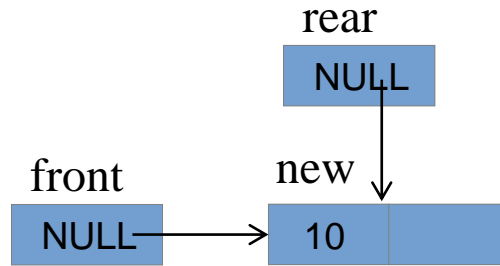
```
else
```

```
    rear → link = new
```

```
    rear = new
```

```
new → link = front
```

```
return e_true
```



Algorithm

```
new = Memalloc(sizeof(Slist))
```

```
if (new = NULL)
```

```
    return e_false
```

```
new → data = data
```

```
if(front = NULL)
```

```
    front = rear = new
```

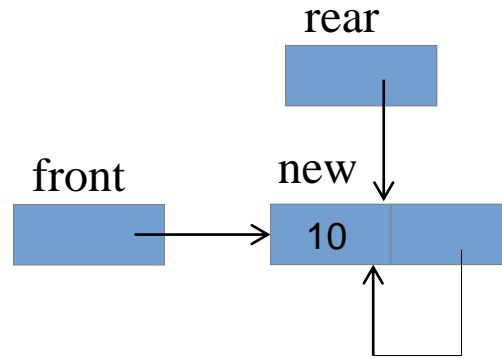
```
else
```

```
    rear → link = new
```

```
    rear = new
```

```
new → link = front
```

```
return e_true
```



Algorithm

```
new = Memalloc(sizeof(Slist))
```

```
if (new = NULL)
```

```
    return e_false
```

```
new → data = data
```

```
if(front = NULL)
```

```
    front = rear = new
```

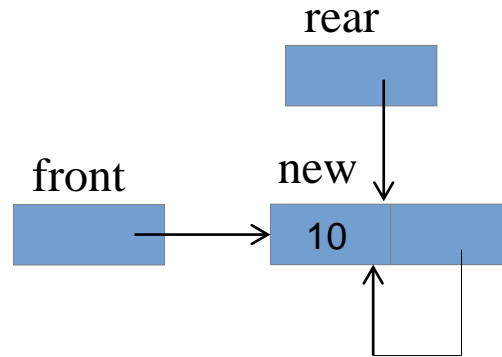
```
else
```

```
    rear → link = new
```

```
    rear = new
```

```
new → link = front
```

```
return e_true
```



Algorithm

```
new = Memalloc(sizeof(Slist))
```

```
if (new = NULL)
```

```
    return e_false
```

```
new → data = data
```

```
if(front = NULL)
```

```
    front = rear = new
```

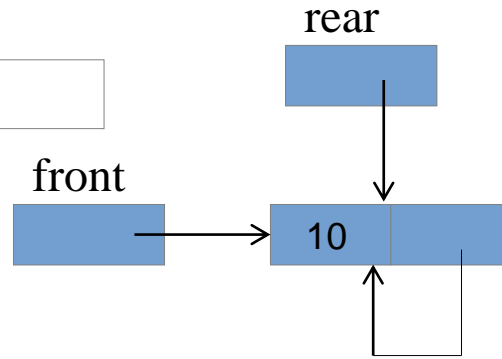
```
else
```

```
    rear → link = new
```

```
    rear = new
```

```
new → link = front
```

```
return e_true
```



Algorithm

```
new = Memalloc(sizeof(Slist))
```

```
if (new = NULL)
```

```
    return e_false
```

```
new → data = data
```

```
if(front = NULL)
```

```
    front = rear = new
```

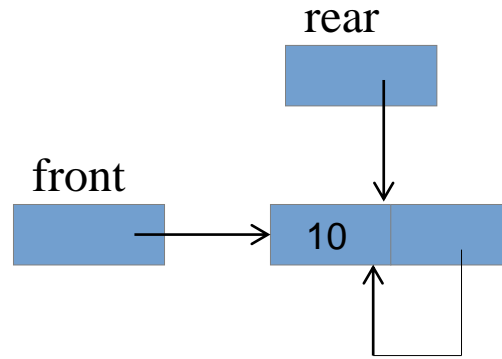
```
else
```

```
    rear → link = new
```

```
    rear = new
```

```
new → link = front
```

```
return e_true
```



Algorithm

```
new = Memalloc(sizeof(Slist))
```

```
if (new = NULL)
```

```
    return e_false
```

```
    new → data = data
```

```
    if(front = NULL)
```

```
        front = rear = new
```

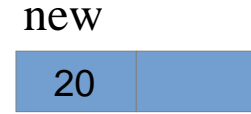
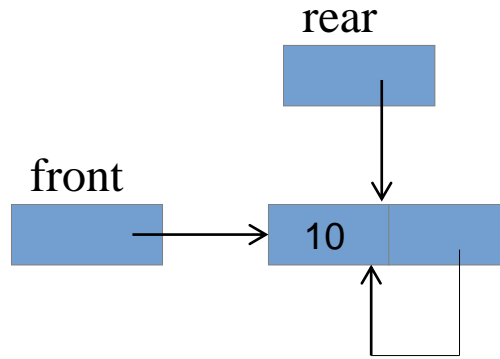
```
    else
```

```
        rear → link = new
```

```
        rear = new
```

```
    new → link = front
```

```
    return e_true
```



Algorithm

```
new = Memalloc(sizeof(Slist))
```

```
if (new = NULL)
```

```
    return e_false
```

```
new → data = data
```

```
if(front = NULL)
```

```
    front = rear = new
```

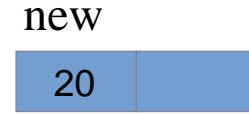
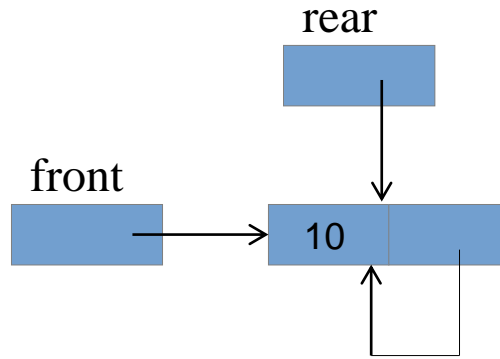
```
else
```

```
    rear → link = new
```

```
    rear = new
```

```
new → link = front
```

```
return e_true
```



Algorithm

```
new = Memalloc(sizeof(Slist))
```

```
if (new = NULL)
```

```
    return e_false
```

```
new → data = data
```

```
if(front = NULL)
```

```
    front = rear = new
```

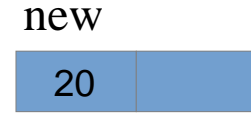
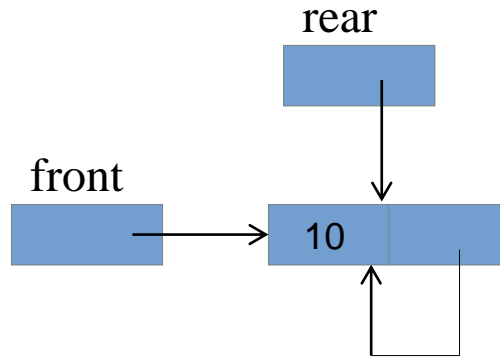
```
else
```

```
    rear → link = new
```

```
    rear = new
```

```
new → link = front
```

```
return e_true
```



Algorithm

```
new = Memalloc(sizeof(Slist))
```

```
if (new = NULL)
```

```
    return e_false
```

```
new → data = data
```

```
if(front = NULL)
```

```
    front = rear = new
```

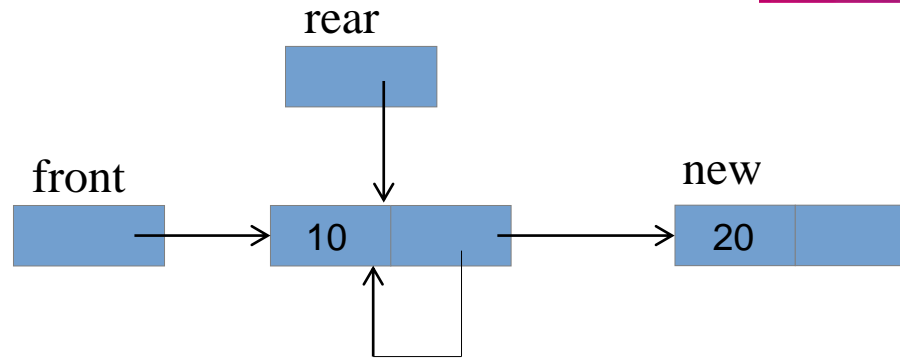
```
else
```

```
    rear → link = new
```

```
    rear = new
```

```
new → link = front
```

```
return e_true
```



Algorithm

```
new = Memalloc(sizeof(Slist))
```

```
if (new = NULL)
```

```
    return e_false
```

```
new → data = data
```

```
if(front = NULL)
```

```
    front = rear = new
```

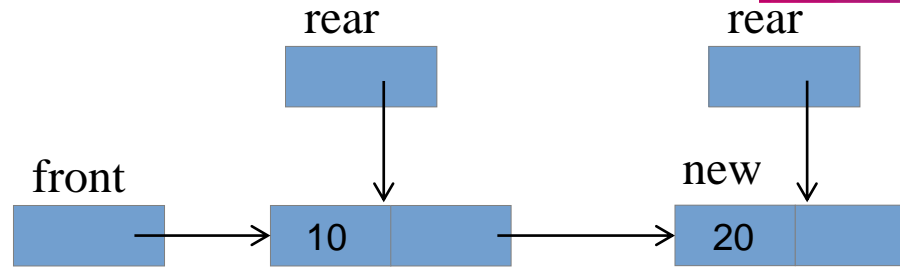
```
else
```

```
    rear → link = new
```

```
    rear = new
```

```
new → link = front
```

```
return e_true
```



Algorithm

```
new = Memalloc(sizeof(Slist))
```

```
if (new = NULL)
```

```
    return e_false
```

```
new → data = data
```

```
if(front = NULL)
```

```
    front = rear = new
```

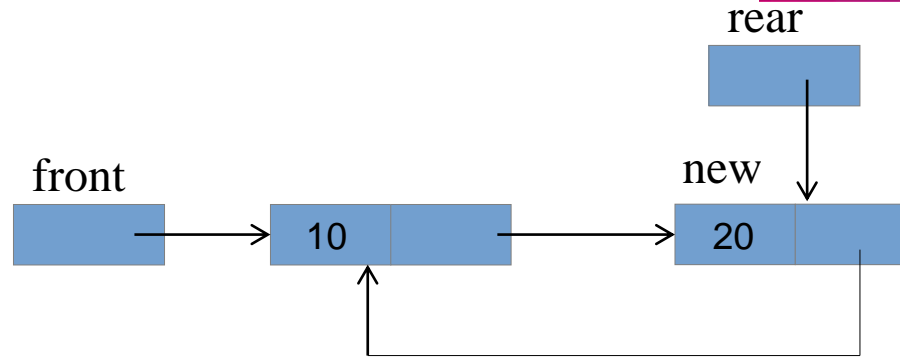
```
else
```

```
    rear → link = new
```

```
    rear = new
```

```
new → link = front
```

```
return e_true
```



Algorithm

```
new = Memalloc(sizeof(Slist))
```

```
if (new = NULL)
```

```
    return e_false
```

```
new → data = data
```

```
if(front = NULL)
```

```
    front = rear = new
```

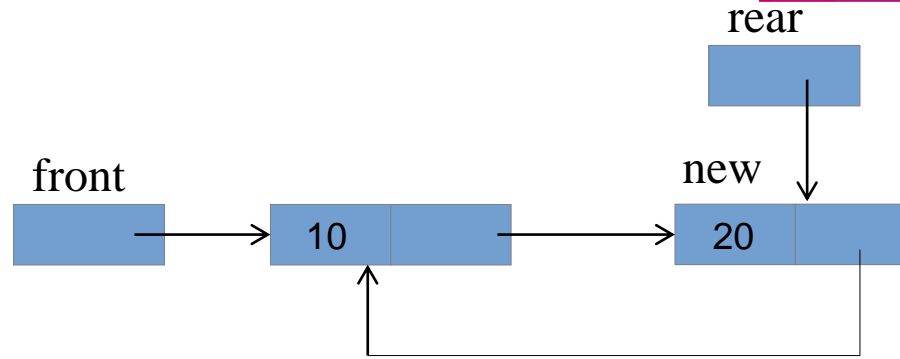
```
else
```

```
    rear → link = new
```

```
    rear = new
```

```
new → link = front
```

```
return e_true
```



Algorithm

```
new = Memalloc(sizeof(Slist))
```

```
if (new = NULL)
```

```
    return e_false
```

```
new → data = data
```

```
if(front = NULL)
```

```
    front = rear = new
```

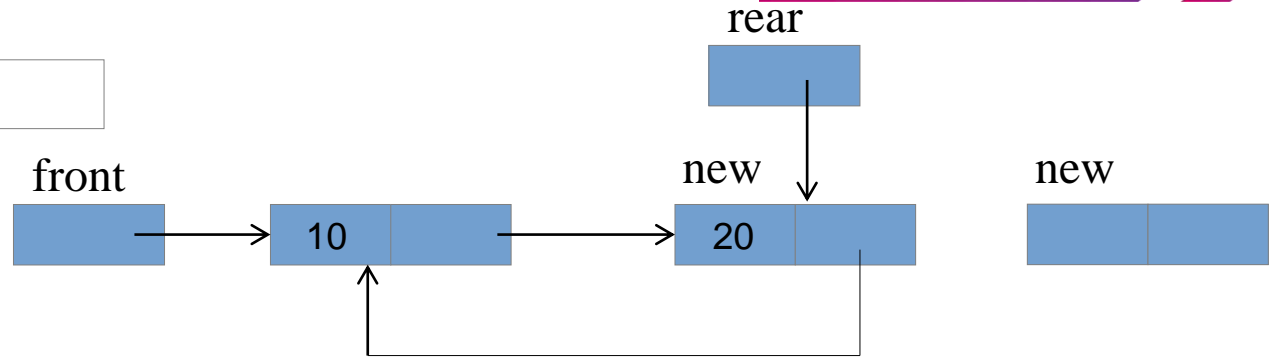
```
else
```

```
    rear → link = new
```

```
    rear = new
```

```
new → link = front
```

```
return e_true
```



Algorithm

```
new = Memalloc(sizeof(Slist))
```

```
if (new = NULL)
```

```
    return e_false
```

```
new → data = data
```

```
if(front = NULL)
```

```
    front = rear = new
```

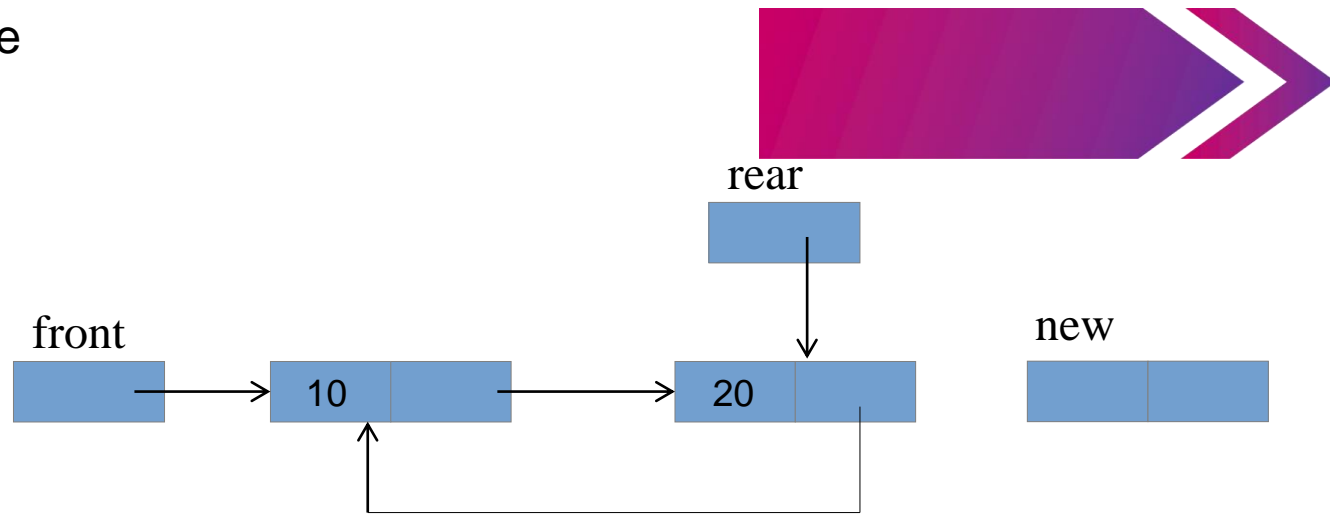
```
else
```

```
    rear → link = new
```

```
    rear = new
```

```
new → link = front
```

```
return e_true
```



Algorithm

```
new = Memalloc(sizeof(Slist))
```

```
if (new = NULL)
```

```
    return e_false
```

```
    new → data = data
```

```
    if(front = NULL)
```

```
        front = rear = new
```

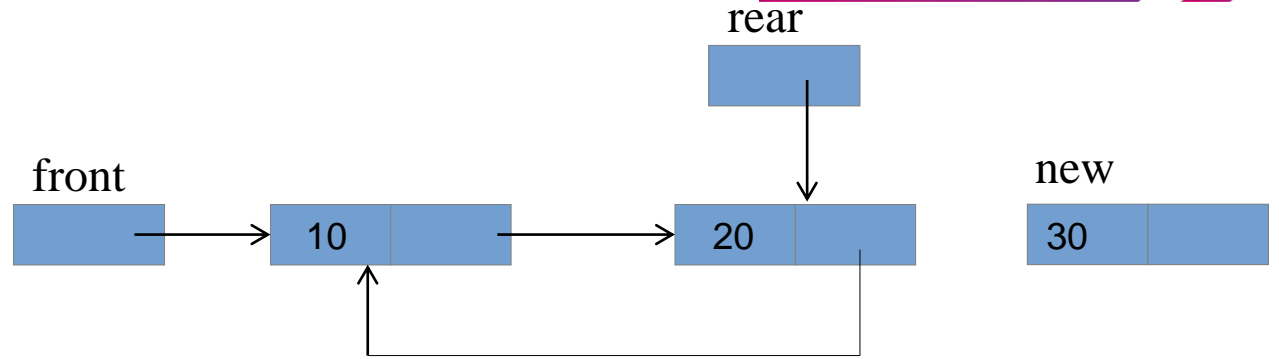
```
    else
```

```
        rear → link = new
```

```
        rear = new
```

```
    new → link = front
```

```
    return e_true
```



Algorithm

```
new = Memalloc(sizeof(Slist))
```

```
if (new = NULL)
```

```
    return e_false
```

```
new → data = data
```

```
if(front = NULL)
```

```
    front = rear = new
```

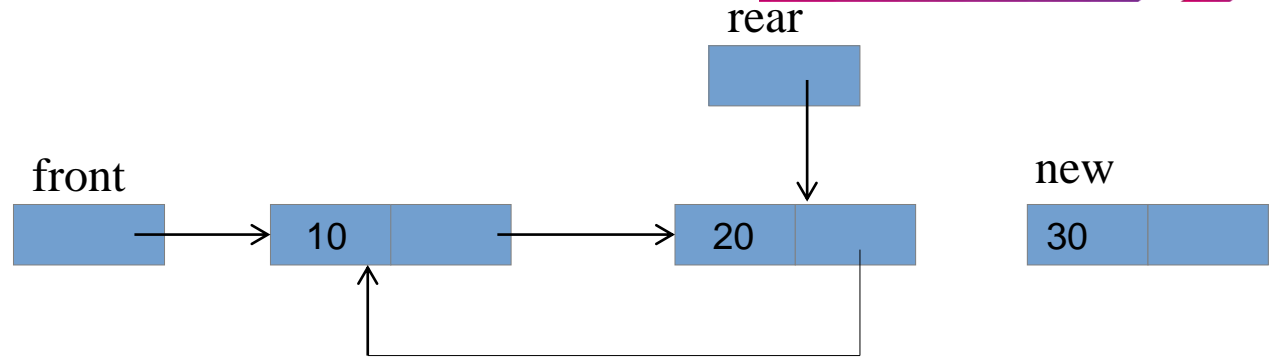
```
else
```

```
    rear → link = new
```

```
    rear = new
```

```
new → link = front
```

```
return e_true
```



Algorithm

```
new = Memalloc(sizeof(Slist))
```

```
if (new = NULL)
```

```
    return e_false
```

```
new → data = data
```

```
if(front = NULL)
```

```
    front = rear = new
```

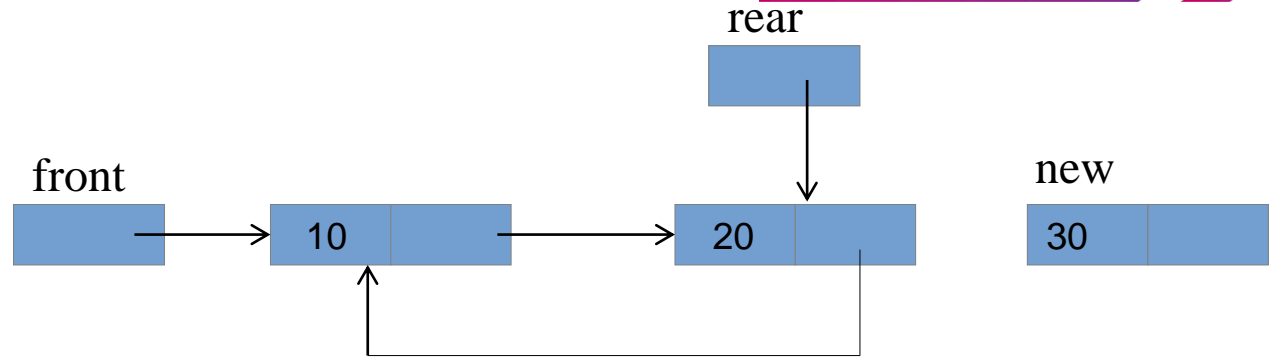
```
else
```

```
    rear → link = new
```

```
    rear = new
```

```
new → link = front
```

```
return e_true
```



Algorithm

```
new = Memalloc(sizeof(Slist))
```

```
if (new = NULL)
```

```
    return e_false
```

```
new → data = data
```

```
if(front = NULL)
```

```
    front = rear = new
```

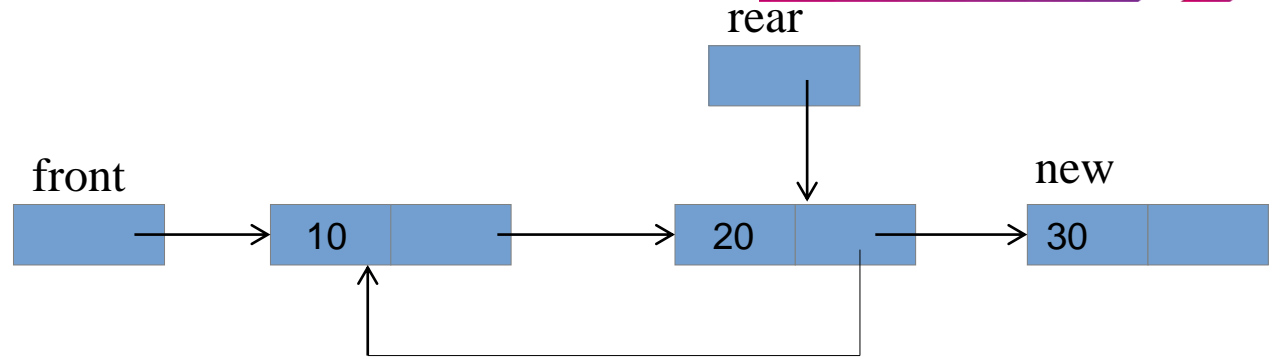
```
else
```

```
    rear → link = new
```

```
    rear = new
```

```
new → link = front
```

```
return e_true
```



Algorithm

```
new = Memalloc(sizeof(Slist))
```

```
if (new = NULL)
```

```
    return e_false
```

```
new → data = data
```

```
if(front = NULL)
```

```
    front = rear = new
```

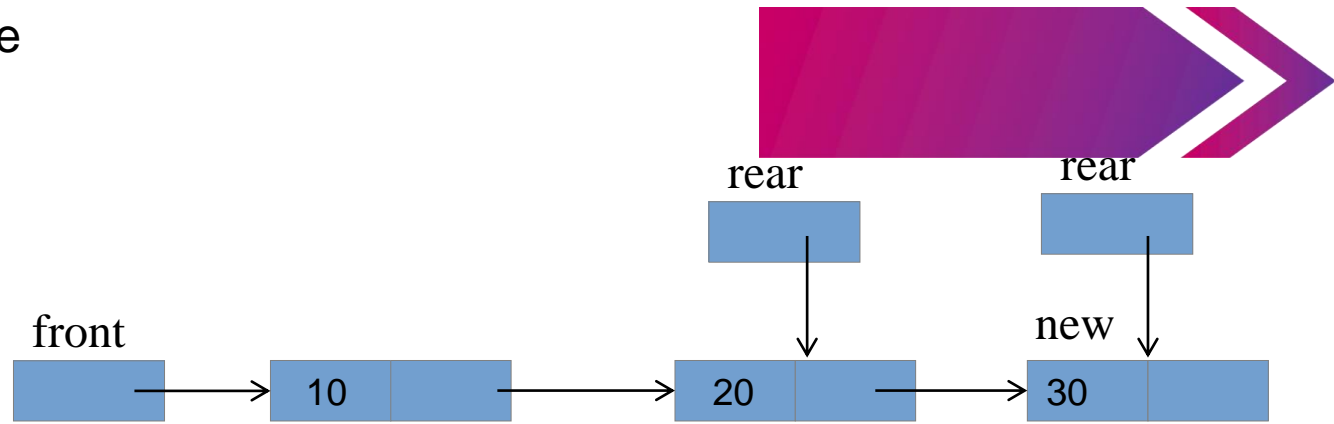
```
else
```

```
    rear → link = new
```

```
    rear = new
```

```
new → link = front
```

```
return e_true
```



Algorithm

```
new = Memalloc(sizeof(Slist))
```

```
if (new = NULL)
```

```
    return e_false
```

```
new → data = data
```

```
if(front = NULL)
```

```
    front = rear = new
```

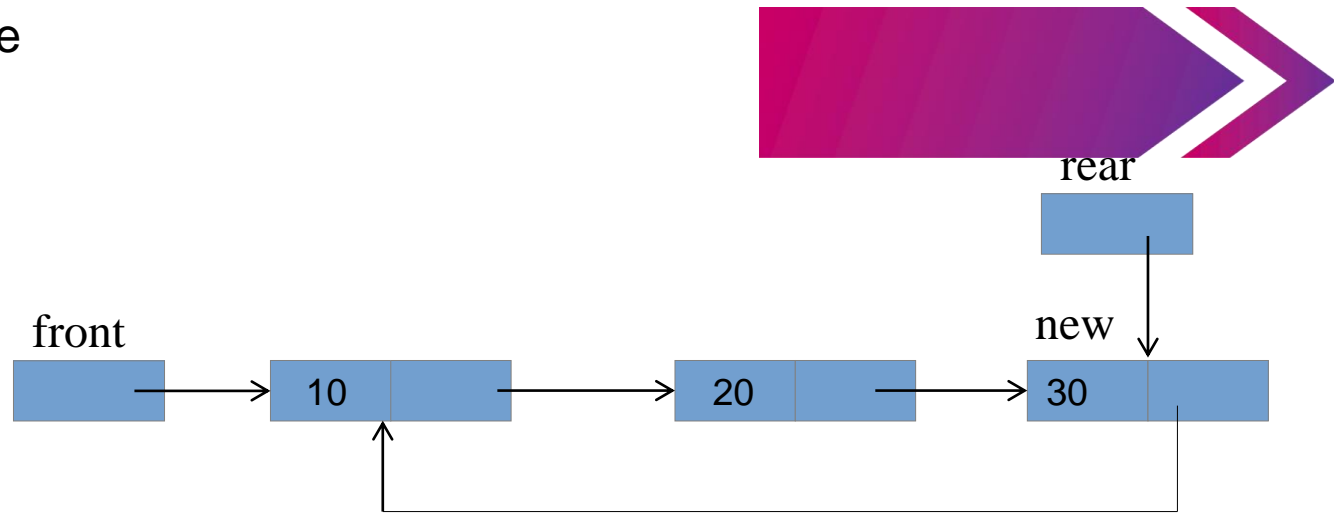
```
else
```

```
    rear → link = new
```

```
    rear = new
```

```
new → link = front
```

```
return e_true
```



Algorithm

```
new = Memalloc(sizeof(Slist))
```

```
if (new = NULL)
```

```
    return e_false
```

```
new → data = data
```

```
if(front = NULL)
```

```
    front = rear = new
```

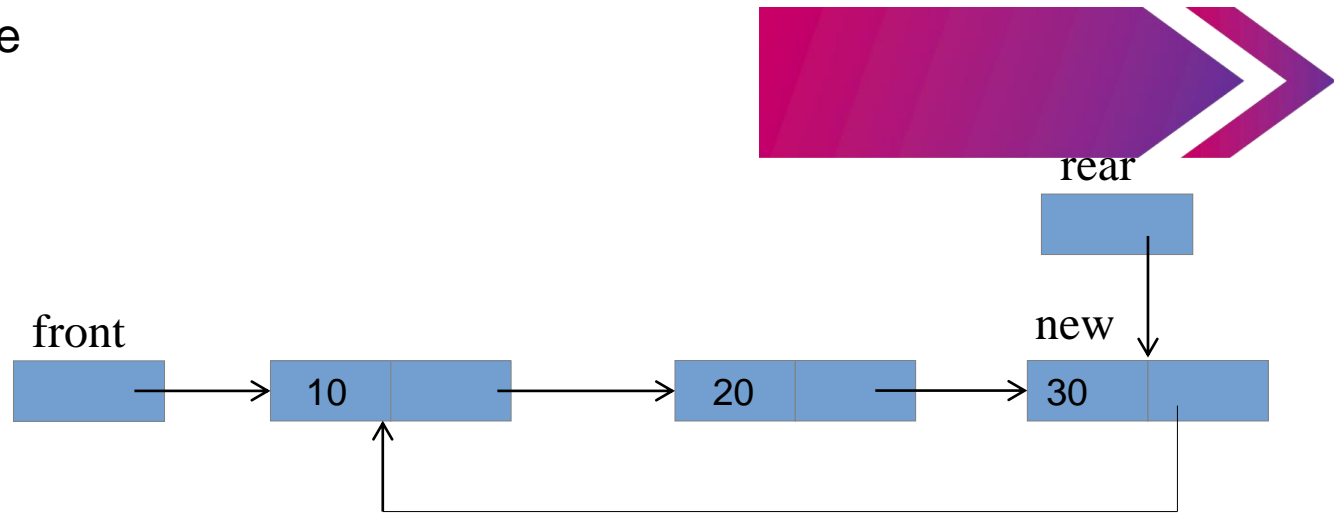
```
else
```

```
    rear → link = new
```

```
    rear = new
```

```
new → link = front
```

```
return e_true
```



Algorithm -dequeue