



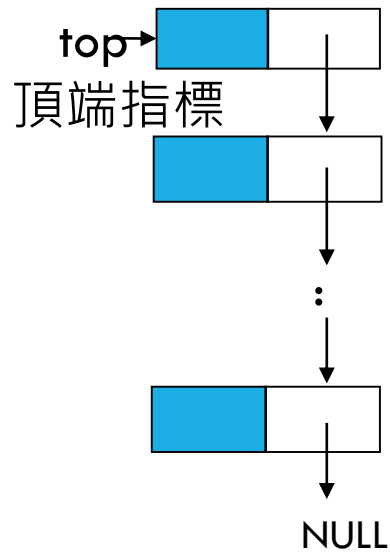
鏈結串列的應用

資料結構
鍾宜玲



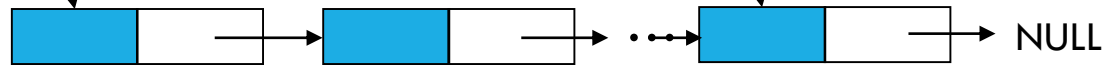
鏈結堆疊與鏈結佇列

鏈結堆疊



前端指標
front

尾端指標
rear



鏈結佇列

鏈結堆疊



假設節點有資料欄 **data** 與鏈結欄 **link**

加入新資料至鏈結堆疊

- 若新資料為 **d**，先製作新節點 **p**：

```
printf("輸入資料：");
```

```
scanf("%d", &d);
```

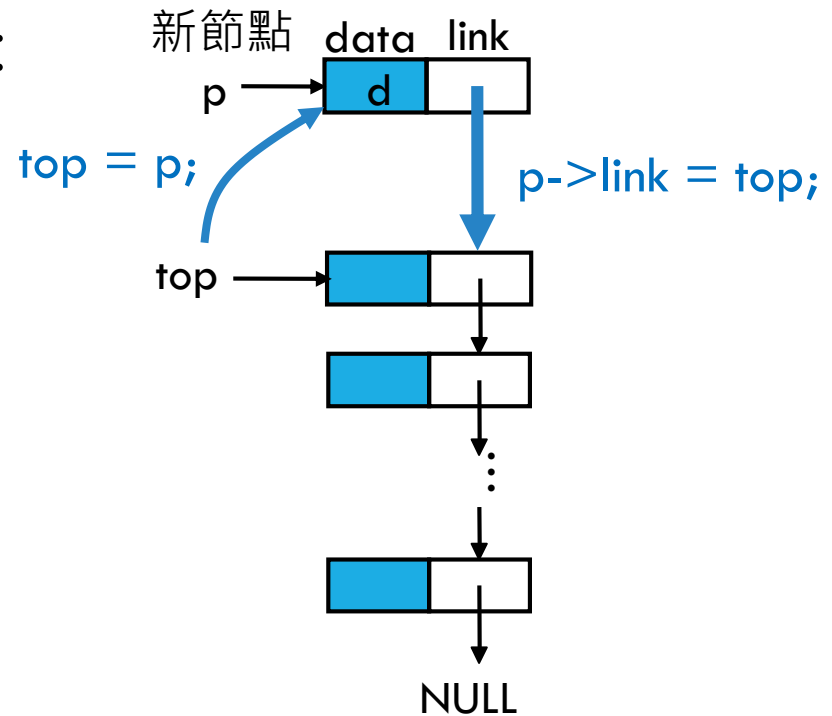
```
p=new();
```

```
p->data=d;
```

- 加入新資料至鏈結堆疊：

```
p -> link = top ;
```

```
top = p;
```

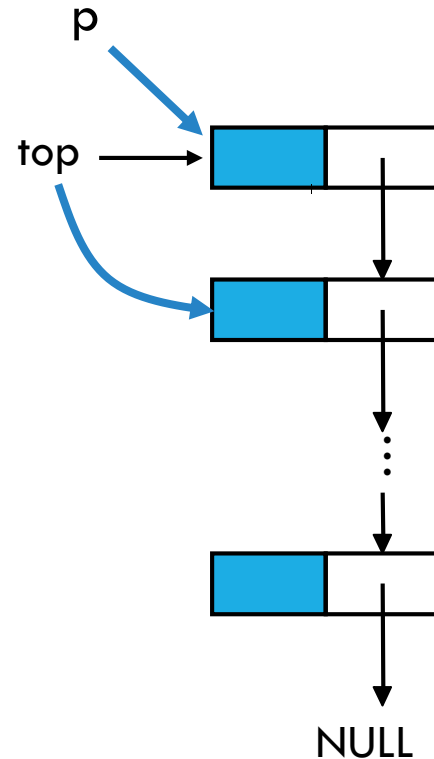


刪除鏈結堆疊的頂端資料



程式片段如下：

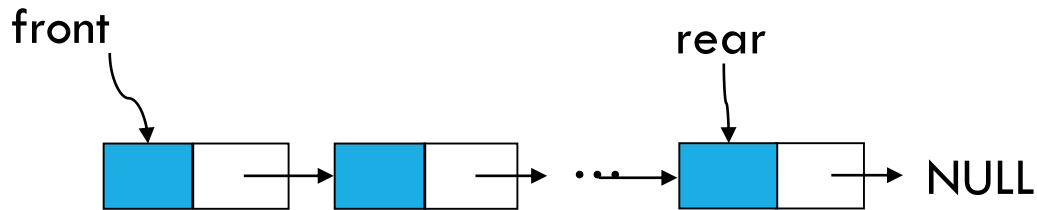
```
p = top;  
top = top->link ;  
free(p);
```



鏈結佇列



假設鏈結佇列之節點含資料欄data及鏈結欄link
加入新資料於鏈結佇列的尾端



- 先製作一新節點 p :

```
printf("輸入資料 : " );
```

```
scanf("%d", &d);
```

```
p=new();
```

```
p->data=d;
```

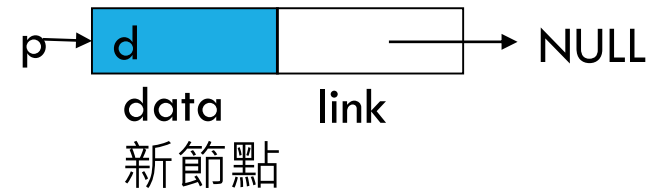
```
p->link=NULL;
```

```
/*若新資料為d*/
```

```
/*分配新節點空間*/
```

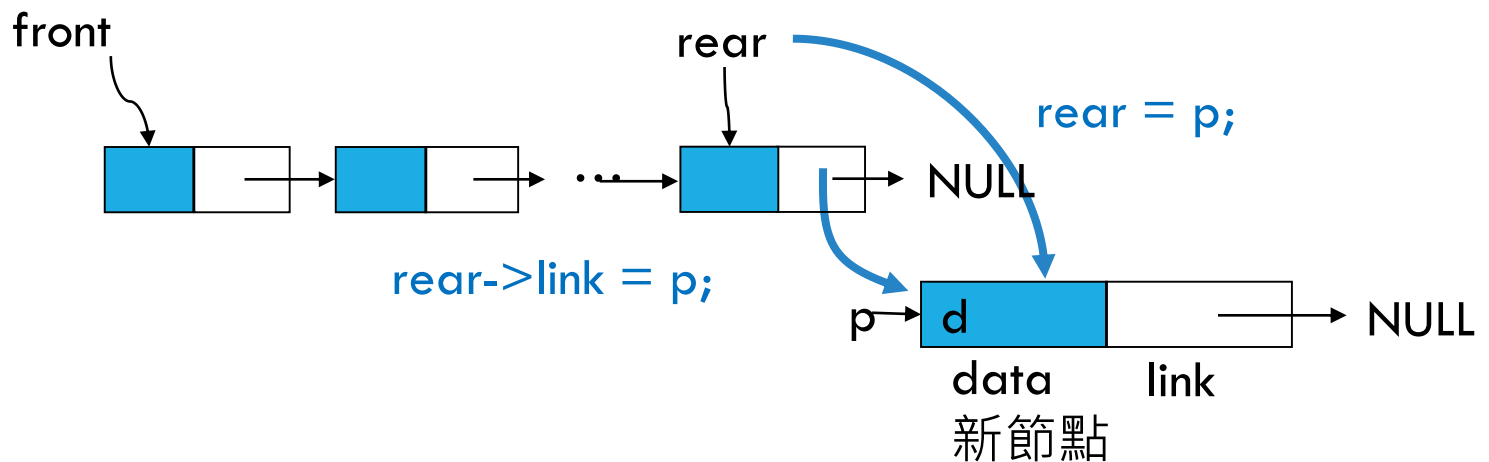
```
/*新節點資料欄為d*/
```

```
/*新節點鏈結欄為NULL*/
```

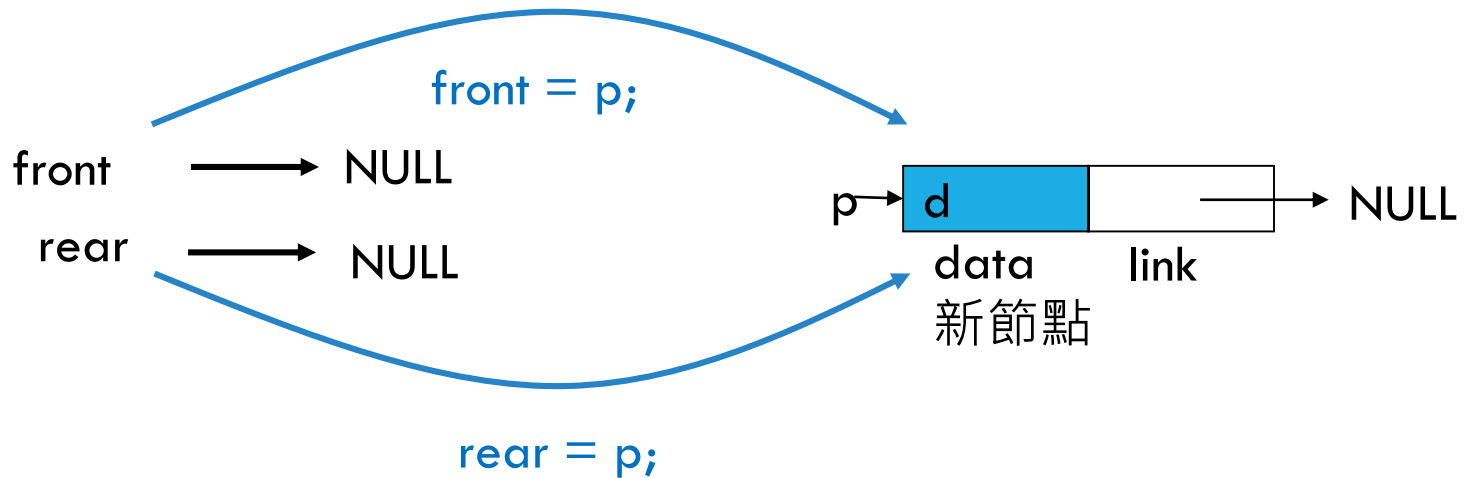


加入節點p至鏈結佇列尾端：

- (1) rear的鏈結欄link改指向新節點p
- (2) rear改指向新節點p



考慮新節點加入至空鏈結佇列時，
必須 **front** 與 **rear** 皆改指向新節點 **p**



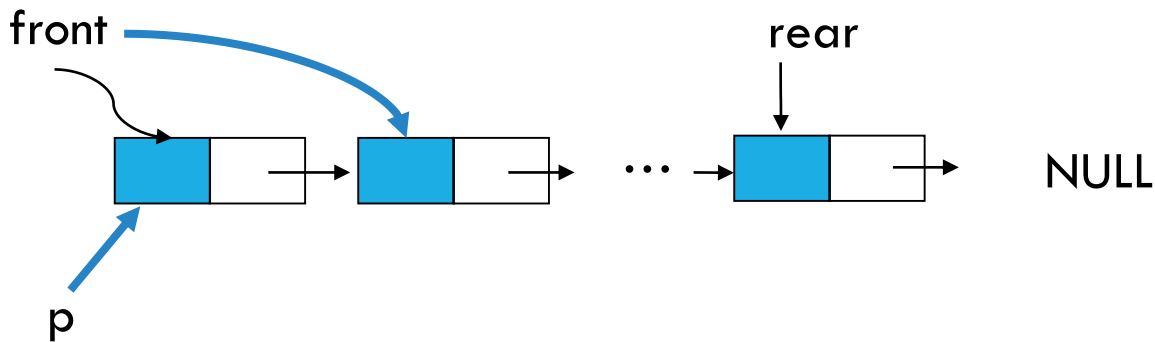
新節點加入至空鏈結佇列

```
if ( front == NUL L ){  
    front = p;  
    rear = p;  
} else {  
    rear -> link = p;  
    rear = p;  
}
```

可改寫為：

```
if ( front == NULL )  
    front = p;  
else  
    rear -> link = p;  
rear = p;
```


刪除鏈結佇列的前端資料



```
p = front;          /* p 指向目前的前端節點front */  
front = front -> link; /*front改指向第二個節點 */  
free(p);           /*歸還原來的前端節點空間*/
```