

## 鏈結串列自編教材（七）

- 在環狀雙向鏈結串列（使用 dummy node）中適當點，加入新的 node（串列中 nodes 的 val 值由小排到大）

環狀雙向鏈結串列

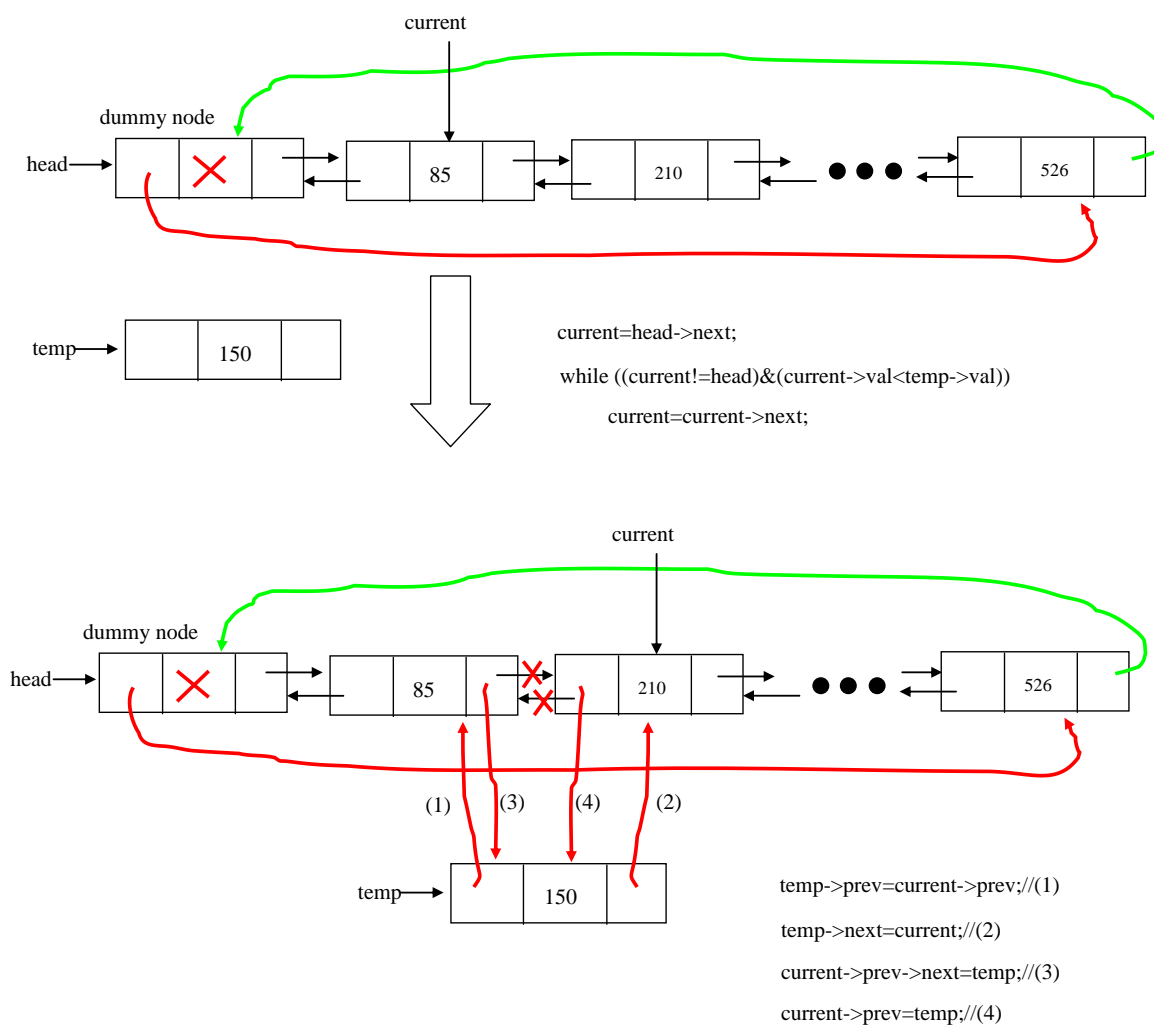


圖 7-1

- 每次以亂數產生一[0,1000]之整數值，若該值>100，則以同方式繼續產生下一亂數值，若該值≤100，則停止此產生值之程序。建立一使用 dummy node 的環狀雙向 linked list，儲存所有產生之值，且該 linked list 中所有 node 之值以由小至大排列；再列印所有產生的值（由小至大排列，與由大至小排列）。(s3-20.c)

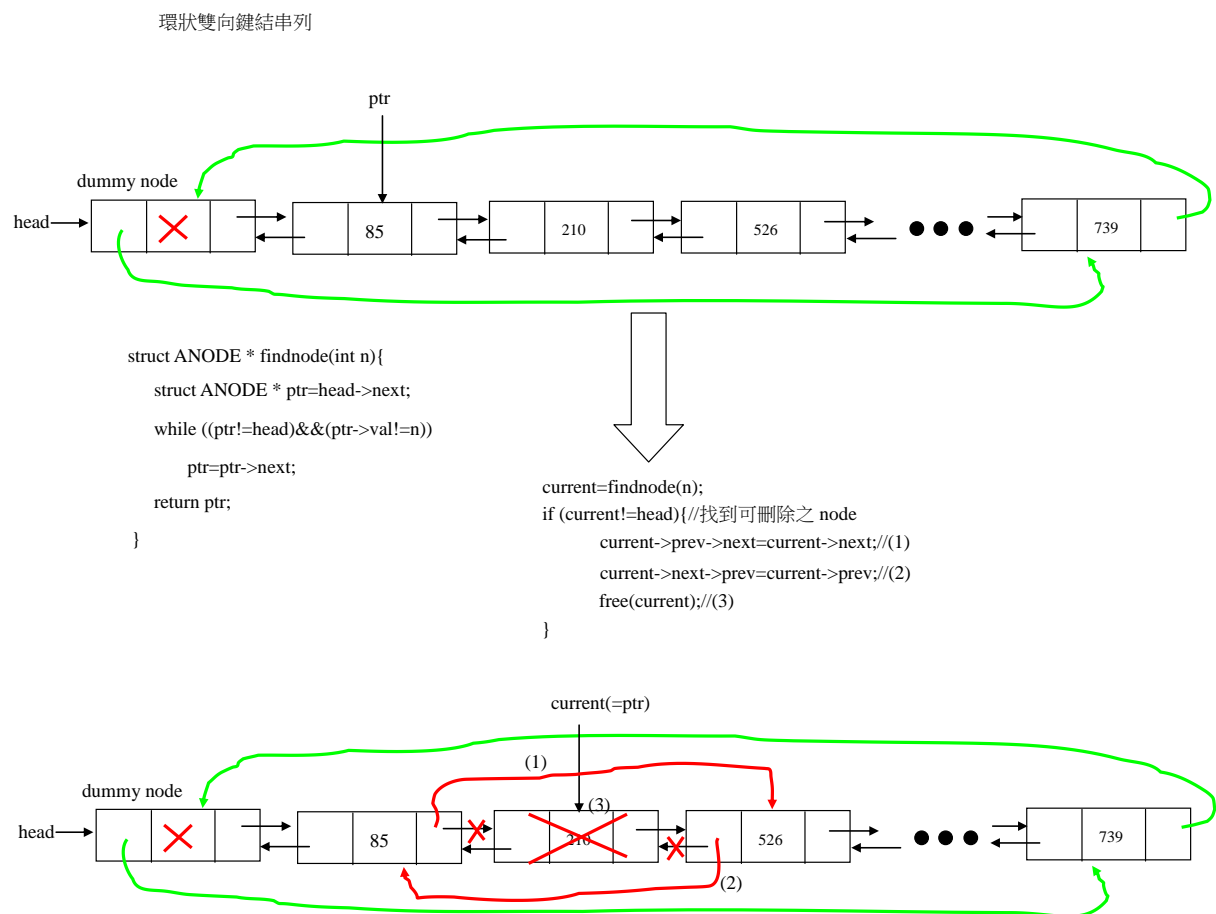
```
-----
struct ANODE *head=NULL,*temp,*current;
...
//若加入新 node 之值>100，則繼續產生下一新 node
do{
    //產生一個新的 node
    temp= generateanode();

    //找加入新 node 的適當位置
    current=head->next;
    while ((current!=head)&&(current->val<temp->val))
        current=current->next;

    //在 current 之前加入新的 node
    temp->prev=current->prev;//(1)
    temp->next=current;//(2)
    current->prev->next=temp;//(3)
    current->prev=temp;//(4)

} while (temp->val>100);
-----
```

- 將由小排到大的鏈結串列中，指定的 node 刪除（例如，將值為 210 的 node 刪除）



- 重做上題，將指定的 **node** 刪除（例如，將值為 210 的 **node** 刪除，若同值的 **node** 多於一個，則刪除第一個有該值的 **node**）。(s3-21.c)

```
-----  
//每次讀入一 n 值，尋找 linked list 中的 node 有該值者，刪除該 node  
while (1){  
    scanf("%d",&n);  
  
    if (n<0) break;//跳出迴圈，結束程式  
    else{  
        current=findnode(n);  
        if (current!=head){//找到可刪除之 node  
            current->prev->next=current->next;//(1)  
            current->next->prev=current->prev;//(2)  
            free(current);//(3)  
        }  
    }  
    printlist(1);  
  
} //end of while  
-----
```