

## 鏈結串列自編教材（四）

- 將由小排到大的鏈結串列中，指定的 **node** 刪除（例如，將值為 5 的 **node** 刪除，或將值為 32 的 **node** 刪除）

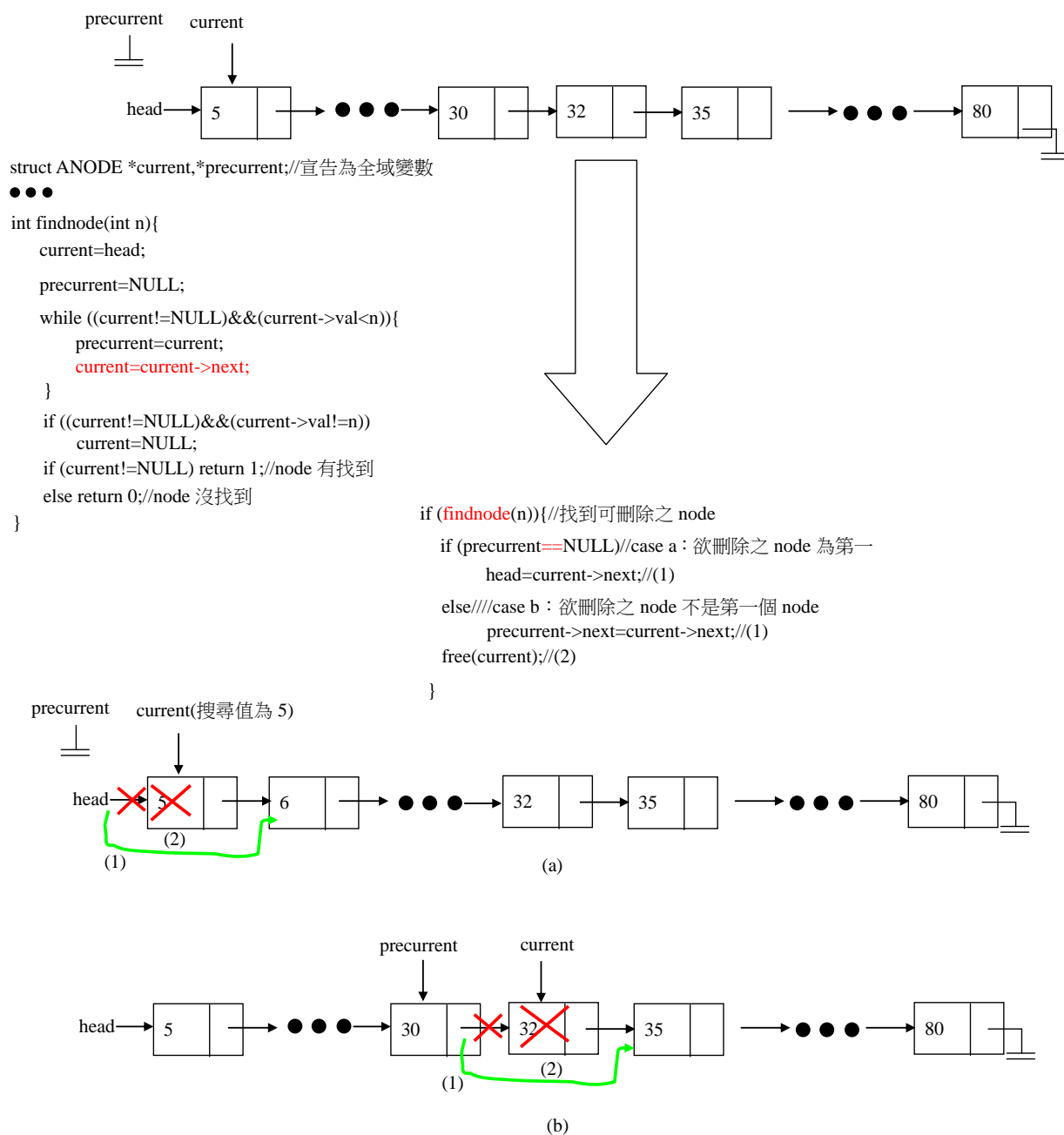


圖 4-1

- 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。建一 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）；再由鍵盤輸入一整數（若<0 則結束執行），若在 linked list 中有 node 含有該值，則將該 node 刪除（有多個 node 含同值，則刪除同值之第一個 node）後印出串列內容。(s3-15.c)

```
-----
struct ANODE *head=NULL,*temp,*current,*precurrent;
...
//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();

    if (head==NULL){//原來為空串列
        head=temp;
    }
    else{//將新產生的 node 加入串列適當的位置（值由小至大排列）
        current=head;
        precurrent=NULL;
        while ((current!=NULL)&&(current->val<temp->val)){
            precurrent=current;
            current=current->next;
        }
        if (precurrent==NULL)
            head=temp;//(1),適用圖 3-3 的情形
        else
            precurrent->next=temp; //(1),適用圖 3-1 及圖 3-2 的情形
        temp->next=current;//(2)
    }
} while (temp->val>100);

//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{
```

```
if (findnode(n)){//找到可刪除之 node
    if (precurrent==NULL)//case a：欲刪除之 node 為第一
        head=current->next;//(1)
    else////case b：欲刪除之 node 不是第一個 node
        precurrent->next=current->next;//(1)
    free(current);//(2)
}
}
```

-----