

Link list

1 Introduction

鏈結串列 (link list) 是一種資料結構，使用指標 (pointer) 將串連節點而成。list 由節點 (node) 組成，節點包含儲存資料的變數以及連結其他節點的指標。與陣列 (array) 相比，不需要連續記憶體空間，意思是說存成陣列形式的資料，例如一些數字，這些數字會存在相連的位置。這使得陣列的資料形式較方便查詢資料，因為知道陣列的開頭和要找資料在陣列中第幾個位置後，指定陣列讀取該位置就能取得。但是在資料不多時容易造成記憶體的浪費。

相對而言，link list 有新資料加入時才會分配到新資料記憶體儲存空間，資料刪除後就把空間釋放。link list 使用動態記憶體存放資料，適合新增或刪除資料，資料可能存放在非連續空間，但不適合查詢資料，有可能要從頭訪問整個 list。

2 Node

節點 (node) 由資料結構 (structure) 和指標 (pointer) 組成。

```
struct ListNode{
    int val;
    ListNode *next;
};
```

結構可以將不同型別的資料收集起來，形成一種新的資料型別或說是新型態的變數；指標是存放位址的地方。所以 ListNode 表示一個單一方向的節點，裡面存了一個數字和指標。可以使用 new operator 分配記憶體給 node，藉由以下方式完成設定一個節點。

```
ListNode *list=new ListNode;
list->val=number;
list->next=NULL;
```

number 是給予的資料，並設定這個 node 指向 NULL，表示這個 list 的結尾。

3 List

將多個節點以指標相連組成鏈結串列。以下舉例說明：

```

ListNode *list=new ListNode;
list->val=10;
list->next=NULL;

ListNode *node1=new ListNode;
node1->val=1;
node1->next=NULL;

node1->next=list->next;
list->next=node1;

ListNode *node2=new ListNode;
node2->val=6;
node2->next=NULL;

node2->next=node1->next;
node1->next=node2;

```

藉由修改前一個節點指向的位置，指向現在的節點，將兩個節點接起來。若要刪除節點可以改變指標方向，跳過中間的節點，並釋放記憶體。

4 References

For further references:

- 程式設計微知識 (十一) 鏈結串列
- [C/C++] 鏈結串列 (Linked List)
- LeetCode 雙刀流：206. Reverse Linked List