

Computer Programming 1 Lab

2022/12/29 Andy Hung

Outline

- Link list
- Debug

Linked List

```
1  typedef struct node Node;
2
3  struct node {
4      int value;
5      Node* nextPtr;
6  }
```

Linked List

```
1  typedef struct node Node;
2
3  struct node {
4      int value;
5      Node* nextPtr;
6  }
```

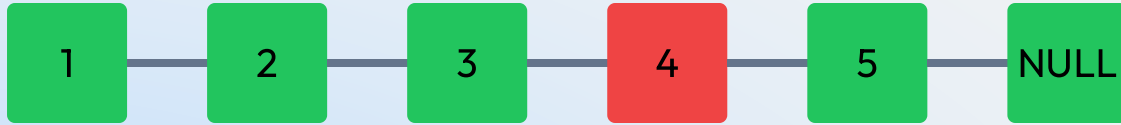
- other concept
- insert
- delete
- remove

Linked List - other concept



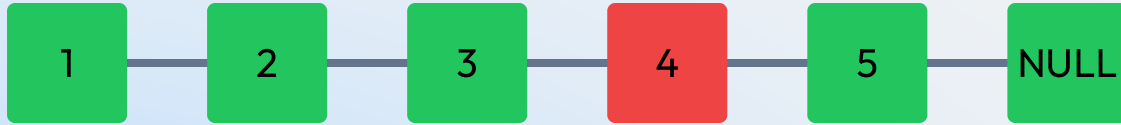
Use a dummy header to avoid strange pointer problem

Linked List - insert



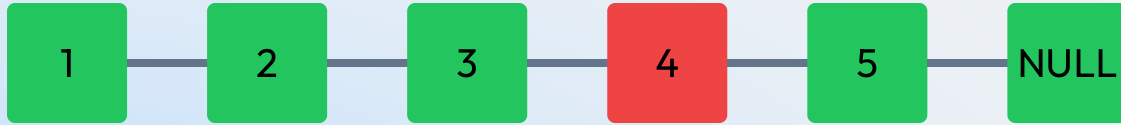
```
1 Node* newPtr = malloc(sizeof(Node));  
2 newPtr -> value = 0;  
3 newPtr -> nextPtr = currPtr -> nextPtr;  
4 currPtr -> nextPtr = newPtr;
```

Linked List - insert



```
1 Node* newPtr = malloc(sizeof(Node));  
2 newPtr -> value = 0;  
3 newPtr -> nextPtr = currPtr -> nextPtr;  
4 currPtr -> nextPtr = newPtr;
```

Linked List - insert



```
1  Node* newPtr = malloc(sizeof(Node));  
2  newPtr -> value = 0;  
3  newPtr -> nextPtr = currPtr -> nextPtr;  
4  currPtr -> nextPtr = newPtr;
```


Linked List - delete



Linked List - delete



```
1 Node* tmpPtr = currPtr -> nextPtr;  
2 currPtr -> nextPtr = tmpPtr -> nextPtr;  
3 free(tmpPtr);
```

Linked List - remove

```
1  while(ptr != NULL) {  
2      Node* nextPtr = ptr -> nextPtr;  
3      free(ptr);  
4      ptr = nextPtr;  
5  }
```

Whenever use malloc, use free then.

Debug

- IO first, then **Logic**
 - 避免有時候邏輯寫完發現input根本就是錯的。
 - 處理IO時，處理一個程序print一次，循序漸進。

Debug

- IO first, then **Logic**
 - 避免有時候邏輯寫完發現input根本就是錯的。
 - 處理IO時，處理一個程序print一次，循序漸進。
- TLE
 - 某個while沒檢查到，ex: recursive, i-
 - 用斷點 `while(1)` 丟oj

Debug

- IO first, then **Logic**
 - 避免有時候邏輯寫完發現input根本就是錯的。
 - 處理IO時，處理一個程序print一次，循序漸進。
- TLE
 - 某個while沒檢查到，ex: recursive, i-
 - 用斷點 `while(1)` 丟oj
- Segmentation fault
 - array index錯誤存取，link list存取 `NULL` 的 `nextPtr`
 - 費神，用以上方法逐步排解

Debug

- IO first, then **Logic**
 - 避免有時候邏輯寫完發現input根本就是錯的。
 - 處理IO時，處理一個程序print一次，循序漸進。
- TLE
 - 某個while沒檢查到，ex: recursive, i-
 - 用斷點 `while(1)` 丟oj
- Segmentation fault
 - array index錯誤存取，link list存取 `NULL` 的 `nextPtr`
 - 費神，用以上方法逐步排解
- Stack Overflow
 - 遞迴過深，檢查終止條件

Debug

- array 開不起來
 - Stack vs heap
 - 用 `malloc` 或用全域變數

Debug

- array 開不起來
 - Stack vs heap
 - 用 `malloc` 或用全域變數
- local vs ghost vs oj
 - gcc version
 - `llvm` vs `gcc`

Debug friend - GDB

- Vscode instruction: WSL setup
- You can also use `gdb` command: GeekForGeeks.
- Other platforms have other good tools.

No exercise this year

