

# HW02

---

409410005 鍾天睿

## 完成內容

---

### 基本內容

- 生成N 筆隨機唯一測資
- 從 N 筆生成的測資 挑選1/10 供查詢
- 測量建立該資料結構所需時間
- 測量查詢該資料結構所需時間
- Linked list
- Array
- Array with Binary Search
- Binary Search Tree
- Hash code

### 額外內容

- 遞增唯一測資（用以測量不平衡的 BST）
- Treap

## 程式架構

---

### Method Interface

```
typedef struct Method {  
    void* (*constructor)(int size);  
    int (*add)(void* data, const char*);    // non-zero return if error  
    int (*query)(void* data, const char*);  // 1 if found, 0 if not found  
    void (*destructor)(void* data);  
} Method;  
  
extern Method m_binary_search, m_linear_search, m_bst, m_llist, m_treap;
```

各個資料結構實現以下函數，並暴露（expose）接口至 struct，提供 main 函數調用。

# 使用說明

## 參數

因為作業給的參數不怎麼漂亮，所以自己想了參數

```
usage: ./bin/hw02 <method> [size] [monotonic]
```

```
available methods: bst linked_list binary_search linear_search
```

## 編譯

`make` 產生 `bin/hw02`，`make debug` 產生 `bin/hw02_debug`

## 實驗

### 環境

```
OS: Arch Linux x86_64
Kernel: 5.11.2-arch1-1
CPU: Intel i5-9600K (6) @ 4.60
GPU: NVIDIA GeForce RTX 2080 T
Memory: 32049MiB
```

### 實驗一 $10^5$

實驗為測驗，取平均值。查詢大小為  $10^4$ 。資料插入時間為總生成時間減去字串生成時間。

資料結構	插入時間	查詢時間
linked list	0.0087 s	4.0577 s
array (linear search)	0.0081 s	1.8578 s
array (qsort + binary search)	0.0077 s	0.0025 s
BST	0.0499 s	0.0055 s
Treap	0.0557 s	0.0056 s

### 實驗二 $10^6$

那兩個需要  $O(n^2)$  的不測了，太慢了。

資料結構	插入時間	查詢時間
array (qsort + binary search)	0.0837 s	0.0557 s
BST	0.9110 s	0.1181 s
Treap	1.0321 s	0.1109 s

### 實驗三 $10^7$

資料結構	插入時間	查詢時間
array (qsort + binary search)	0.8583 s	1.0131 s
BST	16.4218 s	2.0889 s
Treap	17.7891 s	2.0632 s

### 實驗四 $10^6$ 遞增資料

其實沒真的很遞增，因為 hash code 計算方式有點神奇，但是結果看起來夠遞增啦。

資料結構	插入時間	查詢時間
array (qsort + binary search)	0.0763 s	0.0155 s
BST	13.4806 s	1.9685 s
Treap	0.1956s	0.0214 s

### 其他實現細節

字串相關 ( hash code, comparsion ) 請見 `astring.c`。