

HW2

Environment

IDE 使用 CLion 2021.3，專案採用c++14標準，cmake 的最低版本要求為 3.20

Build #CL-213.5744.254

Runtime version: 11.0.13+7-b1751.19 aarch64

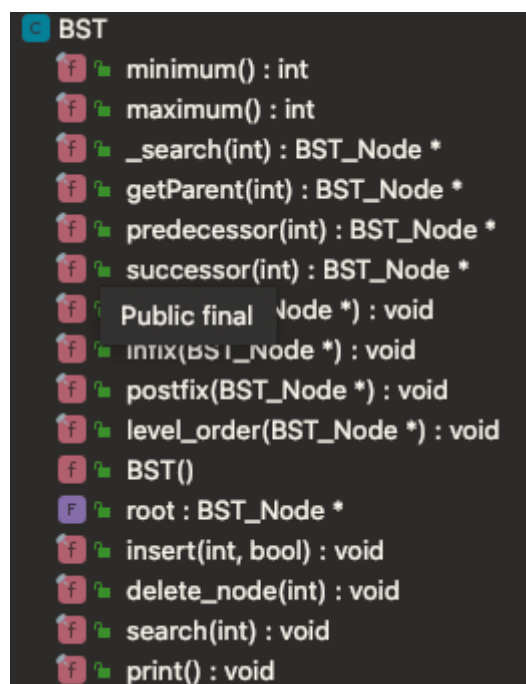
VM: OpenJDK 64-Bit Server VM by JetBrains s.r.o.

架構

`BST_Node` 是樹的單一節點

`BST` 是由 `BST_Node` 構成的 BST 樹

`BST` 中還有提供操作的功能，如下圖所示



```

C BST
f minimum() : int
f maximum() : int
f _search(int) : BST_Node *
f getParent(int) : BST_Node *
f predecessor(int) : BST_Node *
f successor(int) : BST_Node *
f Public final void (*): void
f infix(BST_Node *) : void
f postfix(BST_Node *) : void
f level_order(BST_Node *) : void
f BST()
F root : BST_Node *
f insert(int, bool) : void
f delete_node(int) : void
f search(int) : void
f print() : void

```

(1) Binary searching Tree

第一個項目中分別提供以下選項方法，因此我利用一個靜態的類將他們分別包起來，這樣在代碼閱讀及處理也比較方便

```

c Binary_searching_Tree
  f input_handler(int *) : void
  f insert_number(BST *) : void
  f delete_number(BST *) : void
  f search_number(BST *) : void
  f print(BST *) : void

```

(2)Finding Meaty.

則是一回合遊戲，因此只有單獨把處理陷阱數字的功能拉出來寫

```

c Meaty
  f broccoli_traps(int, int *, int *) : void

```

MAIN

進入時提供三個選項，分別為下圖三個函式，其中第三個沒有功能

```

  f binary_searching_tree() : void
  f finding_meaty() : void
  f escape_and_face_to_music_next_year() : void

```