2018.11.13 周二

# DOCUMENT OF PROJECT

## 1. 简介

本次 project 的所有文件都在 17302010042_夏禹天.zip 的 src 文件夹下，包括本次 project 所提供的 3 个 txt 文件：1_initial.txt ， 2_delete.txt ， 3_insert,txt。以及我所写的 8 个文件：analysis.java ， BplusNode.java ，BplusTree.java ， controller.java ， RB_Node.java ， RB_Tree.java ， UI.java ， Untitled.fxml。

### A. 文件简介

i. Analysis.java：对本次 pj 的 Analysis work 所做的实现（无接口版），使用多个 test function 对红黑树以及 B+树的基本功能进行了检测，没有出现问题，具体的函数功能写在了 analysis.java 的注释当中。因为是无接口版，所以结果都输出到了控制台上面。可以进行修改测试。

ii. BplusNode.java：B+树的节点类，用于构建 B+树的节点

iii. BplusTree.java：B+树的树类，用于构建 B+树

iv. controller.java：对 Untitled.fxml 文件进行响应控制，主要是为 fxml 里面的各个控件写入响应

v. RB_Node.java：红黑树的节点类 ， 用于构建红黑树的节点

vi. RB_Tree.java：红黑树的树类，用于构建红黑树

vii. UI.java：对 Untitled.fxml 的载入，是一个简单的 java 文件，没有特殊功能

viii. Untitled.fxml：与 UI.java、controller.java 两个文件一起构成了本次的 UI 接口

## 2. ANALYSIS 分析（均调用 ANALYSIS.JAVA 中的 TEST 函数实现，PS：电脑配置可能对具体时间有一定的影响）

1、 Initial 时间分析（取一次初始化的时间测试结果，每 100 次插入计算一次时间，总共 50 次）：

红黑树的 50 次插入时间（ns）：

1243612 822623 754966 9826839 1252542 450976 350019 1280968 345802 300698 390801 2519070 315315 292698 332077 1119778 220427 283318 283721 2174314 295119 268382 244133 1306743 334481 328835 338672 11530858 4512349 255305 462966 224403 2721894 453130 1036001 282590 228021 278152 562254 765134 239308 221175 459756 216943 1335508 249671 471804 270836 243000 5119620

平均时间：1196352

B+树的 50 次插入时间（ns）：

2439757 2571241 3465157 1522940 3095416 781565 2822363 901417 810421 831043 483434 598916 368985 601002 343728 1992231 611220 344748 351658 597837 361768 329511 322472 624124 630603 359993 321283 560344 327056 2244920 1100082 362133 594816 377760 605319 339893 319596 325023 564325 344265 363377 336660 480317 2687812 330213 313199 571640 349000 298292 499147

平均时间：855600

在初始化过程中，虽然对比所有的数据发现红黑树和 B+树在插入 100 个数据时时长有一定的波动，但是通过计算两者的插入时间的均值我们发现 B+树的插入时间均值还是要低于红黑树的插入时间均值的。

2、Delete 时间分析（取一次删除的时间测试结果，每 100 次删除计算一次时间，共十次）：

红黑树的 10 次删除时间（ns）：

397281 266519 281194 267488 281859 276708 262663 876271 365799 328734

平均时间：72090

B+树的 10 次删除时间（ns）：

1233428 5741489 1025601 483950 1700181 397877 371514 371208 480314 483442

平均时间：245780

在多次测量证明之后，红黑树的删除时间大致在 200000-300000 之间，而 B+树的删除时间大致在 300000-500000 之间。也就是说，就删除操作而言，B+树的删除时间是要高于红黑树的删除时间的。

3、 Insert 时间分析（取一次插入的时间测试结果，每 100 次计算一次时间，共 10 次）：

红黑树的 10 次插入时间（ns）：

278392 262085 242267 599984 261429 250157 247433 449901 255661 248843

平均时间：61923

B+树的 10 次插入时间（ns）：

479920 382682 374273 3043674 367715 349641 344240 525397 1975640 348647

平均时间：163827

经过多次测量，红黑树的插入时间大致在 200000-350000 之间，B+树的插入时间大致在 300000-400000 之间。也就是说，对插入操作而言，B+树的插入时间是要略长于红黑树的插入时间的。

PS：有关 insert 和 initial 的分析相反的解释：由于 initial 时，数据量较大，而且测试结果较为不稳定，无法对某些数据进行取舍，因此不便于对数据的大致范围作出推断，只能进行取均值的操作,在大多数情况下 B+树的均值还是要低于红黑树的。

4、 Query a word 分析（随机抽取 10 个英文关键字进行查找，每次查找一个，共 10 次）：

红黑树的 10 次查找时间（ns）：13745 2507 1962 2211 1702 1671 952 523 1382 1715

平均时间：2837

B+树的 10 次查找时间（ns）：52891 12209 9853 7119 8466 8830 6759 6639 8476 7722

平均时间：12896

多次测量后，红黑树的单个查找大致在 1000-2000 左右，B+树的单个查找大致在 5000-8000 左右，所以就单个查找而言，B+树的单个查找时间是长于红黑树的单个查找时间的。

5、 Query some words 分析（就多个 some words 组进行单独测试）：

因为考虑到不同的 words 组的长度可能不一致，因此无法对同一种树的结果进行分析，只能进行比较分析。

就“ant”和“cat”这个 words 组进行测试：

红黑树 10 次组查找时间（ns）：

12897012 8608449 8003706 9082985 5529923 14589678 6491331 9191296 8049317 3857354

B+树 10 次组查找时间（ns）：

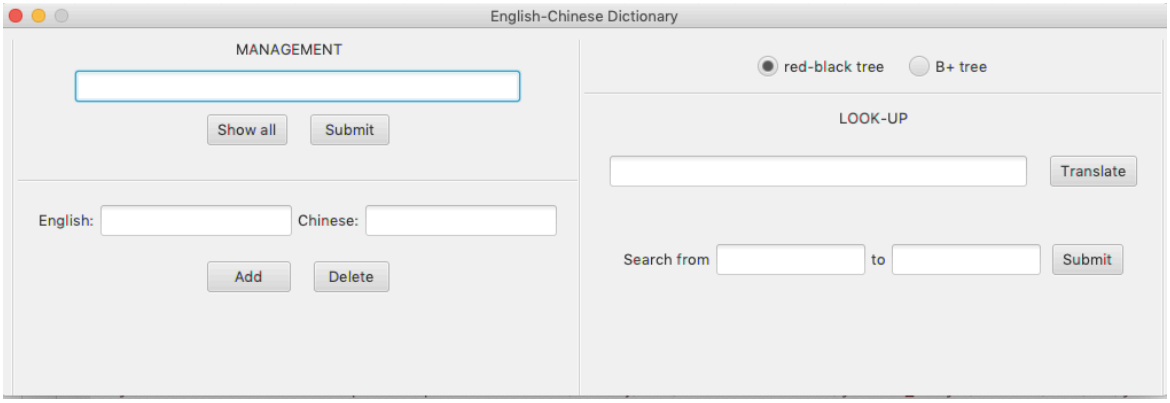
11394187 6483964 8741496 8812285 6605768 14378258 14551143 12127752 7212683 6914724

由于数据波动性仍然较大，但通过数据比对可以看出来大部分时候 B+树的组查找时间还是要长于红黑树的组查找时间的。

## 3. ANALYSIS 结果分析

就数据结果而言，除了初始化的平均时长 B+树是要短于红黑树外，其他的操作 B+树所需的时长都是长于红黑树的。就时间性能而言，还是红黑树要优于 B+树的，但 B+树也有它自己的优点，B+树的空间局部性还是更好的。

## 4. UI




大致的 UI 与助教给的范例一致，实现了该实现的功能。

### A. MANAGEMENT 部分的文本框

需要输入文件名（基于 src 目录下的文件），例如 1_initial.txt ，然后点击其下方的 submit 按钮，如果操作正确，将会在左下方出现一行例如：Initial successfully 的字段

B. 插入和删除部分（左下半部分）

分别在 English 和 Chinese 后面的文本框中输入想插入的英文和中文，点击 add/delete 按钮将分别进行添加和删除操作。当正确添加或删除后，会在左下方出现一行例如：add successfully 的字段

C. Rb tree 和 B+ tree 的单选框组合

是切换红黑树和 B+树的部分

D. 翻译部分（右侧中间部分）

向 translate 按钮左侧的文本框中输入一段英文后，点击 translate 按钮，如果树中有这个英文，将会在整个框子的左下方出现它的中文，如果没有这个英文，将在左下方出现：no such word 的字段。

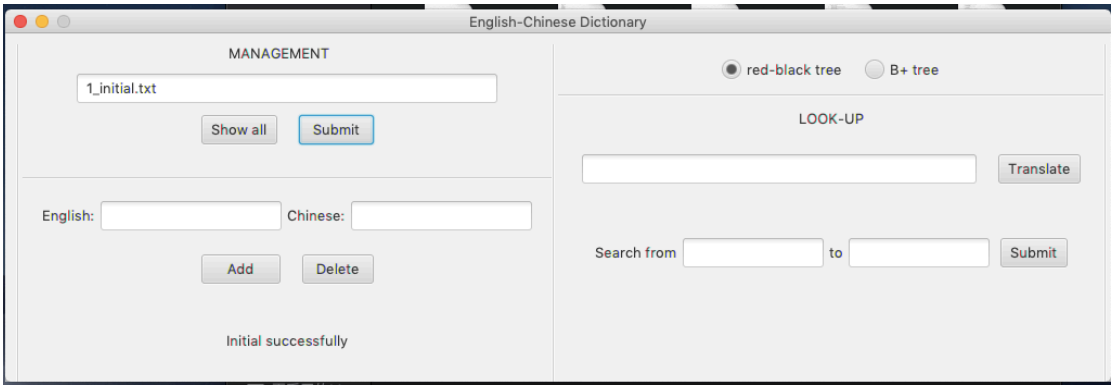
E. 翻译一组单词（右侧下部分）

分别往两个文本框内输入两个英文，如果前一个确实比后一个小，将会弹出一个文本框，里面是找到的所有介于这两个英文之间的结果（按照从小到大的顺序）。如果前一个比后一个大，将在左下方出现错误提示。

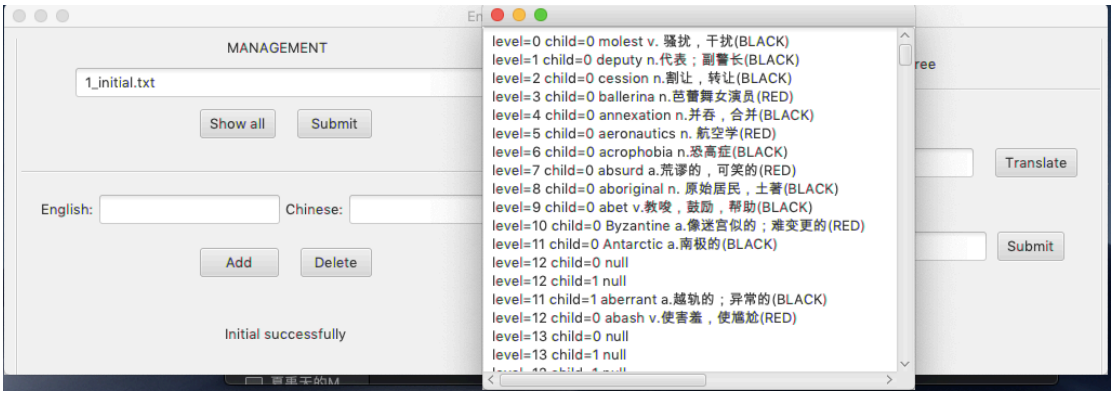
F. Show all 按钮：

前序打印该树，并会弹出一个文本框，将结果打印在文本框当中

## 5. 附实验图（UI 部分）




初始化后的显示




初始化后点击 SHOW ALL 按钮的显示


插入数据后的显示


查找刚插入的数据的显示



删除并查找后的显示


组查找后的显示