**By Chau Minh Nguyen**

**CMP 561 Project**

**Implementation**

I have solved the problem with the following 3 structures:

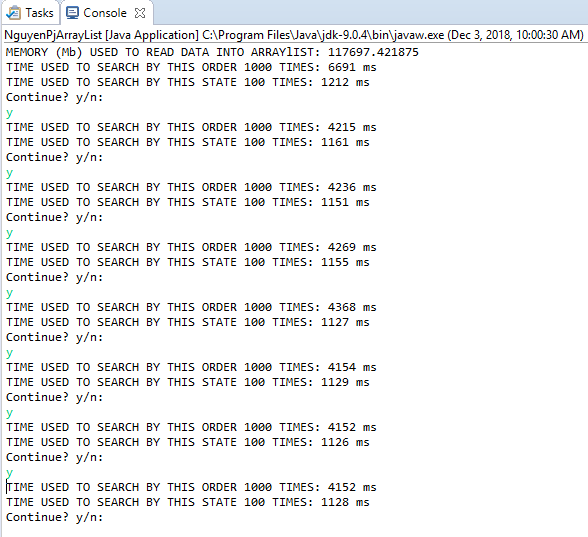
* ArrayList
* Binary Tree
* HashMap

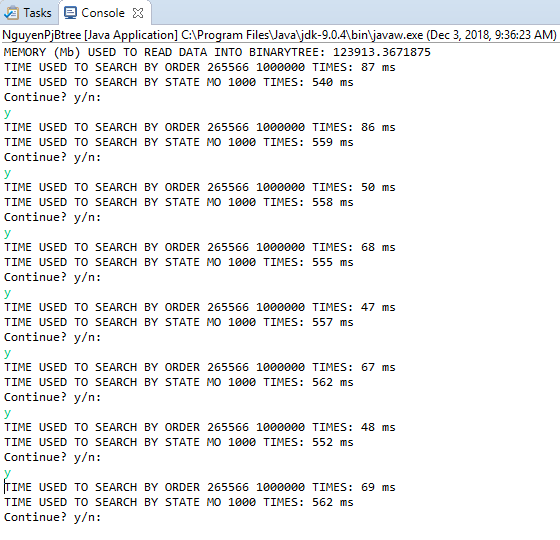
Please see the attached Java files for details

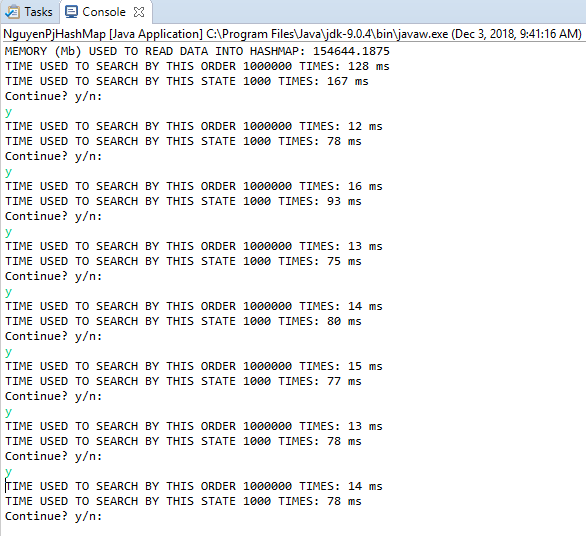
To test the programs, I have run the solutions with HashMap and Binary Tree to search by Order #265566 for 1000,000 times, and search by State “MO” 1000 times

The solution with ArrayList, due to too long run time, I have let it search 1000 times by Order and 100 times by State and projected up to compare with other solutions.

These are screenshots of the results:







**Summary:**

|  |  |  |  |
| --- | --- | --- | --- |
| **ArrayList (Projected)** | | | |
|  | **Memory used (Mb)** | **Search by Order 1000,000 times (Ms)** | **Search by State 1000 times (Ms)** |
| 1st | 116,756 | 4,269,000 | 11,550 |
| 2nd | 113,683 | 4,368,000 | 11,270 |
| 3rd | 119,827 | 4,154,000 | 11,290 |
| 4rd | 117,697 | 4,152,000 | 11,260 |
| 5rd | 109,075 | 4,152,000 | 11,280 |
| **Average** | **115,408** | **4,219,000** | **11,330** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Binary Tree** | | | |
|  | **Memory used (Mb)** | **Search by Order 1000,000 times (Ms)** | **Search by State 1000 times (Ms)** |
| 1st | 123,913 | 68 | 555 |
| 2nd | 124,431 | 47 | 557 |
| 3rd | 122,549 | 67 | 562 |
| 4rd | 124,229 | 48 | 552 |
| 5rd | 122,892 | 69 | 562 |
| **Average** | **123,603** | **60** | **558** |

|  |  |  |  |
| --- | --- | --- | --- |
| **HashMap** | | | |
|  | **Memory used (Mb)** | **Search by Order 1000,000 times (Ms)** | **Search by State 1000 times (Ms)** |
| 1st | 157,347 | 12 | 78 |
| 2nd | 156,178 | 13 | 75 |
| 3rd | 158,738 | 14 | 80 |
| 4rd | 157,937 | 15 | 77 |
| 5rd | 154,664 | 13 | 78 |
| **Average** | **156,973** | **13** | **78** |

**Conclusions**:

In term of Memory used to build data structure, ArrayList is the best, following by Binary Tree, then HashMap

In tern of Search speed, however, HashMap is superior to Binary Tree, and ArrayList is the worst.