|  |  |  |
| --- | --- | --- |
| Searching Report | March 12  2009 | |
| Sequential, Binary, Interpolation, BST-trees, AVL-trees | |  |

###### 

Alexandria University

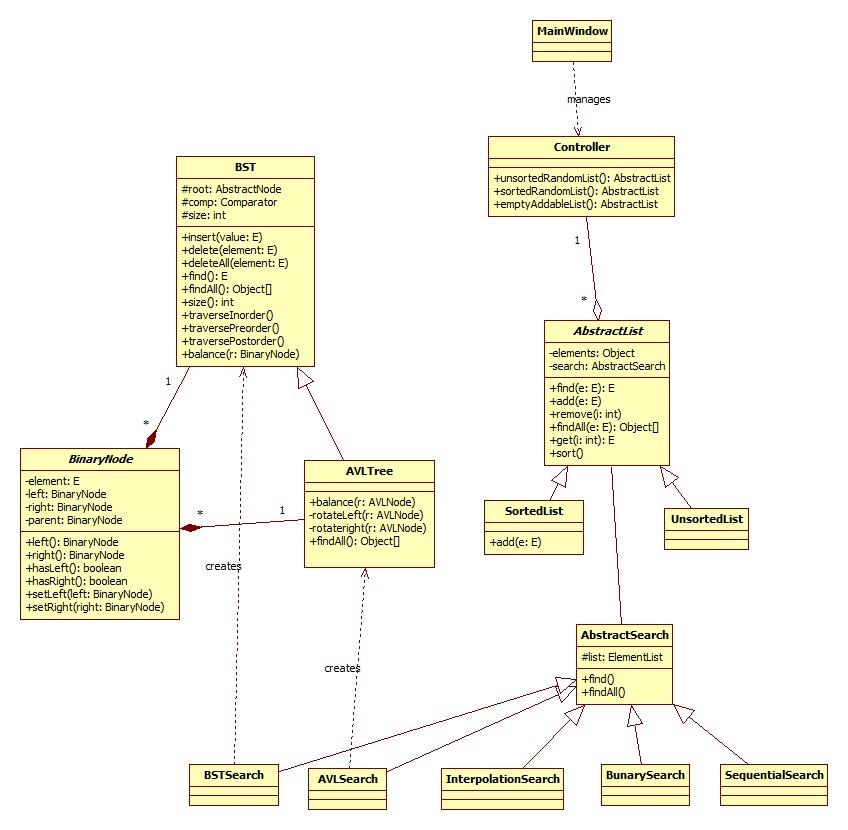
Faculty of Engineering

Computer and Systems Engineering Department

Automatic Control

Name: Mostafa Mahmod Mahmod Eweda

Seat No.: 65

**Class Diagram**

**Sequential Search**

**Advantages**

* Simple to implement.
* Maintainable and easy to change and use its internal data.
* Used in ever day’s program for simple non-requiring operations.
* Consumes no external or internal space for recursive operations as it can be easily implemented iteratively.
* Goes the same on sorted and unsorted lists.

**Disadvantages**

* Slow.
* Always eliminating one item per iteration even if it is much far from the desired destination. (Normal step = 1).
* O (n).

**Binary Search**

**Advantages**

* Fast as it eliminates half of the elements in the list by one iteration.
* Can be implemented both iteratively and recursively.
* O (log2(n))
* Runs the same for any distribution of the list.

**Disadvantages**

* Requiring a previously sorted list
* Always eliminating half of the list even if the desired element is in one of the edges if the list.

**Interpolation Search**

**Advantages**

* Fast as it eliminates most of the elements in the list by one iteration.
* Can be implemented both iteratively and recursively.
* O (log2log2(n)) if the list is normally distributed.

**Disadvantages**

* Requires previous knowledge that the data is normally distributed to achieve the best performance.

**BST (Binary Search Trees)**

**Advantages**

* Structured search.
* Can be used in implementing decision trees as in games or artificial intelligence.
* Having the data ready for searching instead of having the method ready for the list.
* O(log2n) for randomly inserted lists.

**Disadvantages**

* Can reach to a linked list behavior when the data sequence is already sorted O (n).
* Consumes additional memory to save the tree nodes.
* Requires additional effort for implementation.

**AVL Trees**

**Advantages**

* Structured search.
* Can be used in implementing decision trees as in games or artificial intelligence.
* Having the data ready for searching instead of having the method ready for the list.
* O(log2n) for the worst case.
* Consumes less memory than BST for dereferenced nodes (not noticeable).

**Disadvantages**

* Consumes additional memory to save the tree nodes. O(n).
* Requires much more additional effort for implementation.

**Comparison between BST and AVL trees**

|  |  |  |
| --- | --- | --- |
| Construction time | More time is needed for balancing the tree but when the tree gets excessively big, the steps done for balancing is negligible. | Good in time management but when the tree gets too big, the insertion becomes costly either. |
| so they are two much equal. |
| Search time | Better | Good |
| Sorted list (worst) | Doesn’t matter. O(n log2n) | Changes to a linked list. O(n) |
| Unordered | Doesn’t matter. O(n log2n) | O(log2n) for average. |

Sorted List

Unsorted List