# Searching Algorithm – Binary Search Tree

For the implementation of a searching algorithm into the Bluebrook Airport system, the chosen algorithm is a Binary Search Tree. The purpose of using a searching algorithm in the system is to search for data within the system. Data can include flight data, plane data and customer data, staff rotas.  
  
Each vertex has each 0, 1 or 2 branches, which holds a record/key. At each vertex, smaller keys than the vertex are added to a subtree on the left, larger or equal keys are added to the subtree on the right.

Main Operations  
1. Search  
2. Insert  
3. Pre-order Traversal  
4. In-order Traversal  
5. Post-order Traversal  
  
Steps – (example - search for flight time)  
1. User enters a flight time  
2. System uses the first flight time element in the xml file, this time becomes the root node  
2. The system cycles through the next flight time, if it is less than the root, the time is added to the left side of the tree  
3. If the time is greater or equal to the previous flight time, add to the right side of the node.

Search: 15:32

Nottingham – Paris  
15:32 – Terminal A

Advantages  
- Simple to Implement  
  
- Can be used for a variety of different data searches – e.g. flight, customers, staff.  
  
Disadvantages  
-Data must be sorted first