

Activity No. <n>	
<Replace with Title>	
<b>Course Code:</b> CPE010	<b>Program:</b> Computer Engineering
<b>Course Title:</b> Data Structures and Algorithms	<b>Date Performed:</b> 9/11/25
<b>Section:</b> CPE21S4	<b>Date Submitted:</b> 9/11/25
<b>Name(s):</b> QUIYOYO, ANGELO M	<b>Instructor:</b> Jimlord Quejado
<b>6. Output</b>	
1. What is a search tree in data structures?	
- It is a kind of Binary tree in data structure where each of the nodes contains some unique key and satisfies specific ordering properties. Search trees are widely used because they offer a good balance of efficient searching, dynamic insertion/deletion, and the ability to easily retrieve data in sorted order.	
2. What are the Different types of search algorithm in data structures? Differentiate each type of search.	
- Some of the types of search algorithms in data structure has distinct characteristics and applications, some has a different requirements and how it compromise the algorithm. Some type like linear search where it checks each elements from beginning to end until it identifies or reach its target element. The difference, is that it does not require the data to be sorted since its efficiency is low for some large datasets. For Binary search, its algorithm divides the search interval in half where it compares the target value with the middle element of the sorted data. This type requires the data to be sorted not like linear search and it is more efficient than linear search.	
3. What operations / implementations can be performed using binary and linear search operations?	
- For Linear search, it's implementations is that, it can support the sequential traversal like arrays and linked list. For Binary search, it implemented the sorted arrays. Linear search can easily adapted to find its minimum or maximum element in some sorted list, while binary search sorted the list at the minimum element where it starts at the beginning and for maximum is at the end.	
4. What are the advantages in using binary search tree as data structure?	
- The advantages while using binary search is that it can sort out data, making it easier to retrieve data in a specific order. Adding and removing elements in binary search is much more simpler and efficient since it looks like its taking a shortcut. One advantages is that binary is much more quicker than linear because of the data that needs to be searched half of the elements in each steps then locating the element at the center dividing it to 2.	
5. Give an example program using binary search and Linear search.	
<pre>#include &lt;iostream&gt; using namespace std;  int linearSearch(int arr[], int size, int target) {     for(int i = 0; i &lt; size; i++) {         if(arr[i] == target)             return i;     }     return -1; }  int main() {     int data[] = {34, 78, 12, 56, 89, 23};</pre>	

```
int size = sizeof(data)/sizeof(data[0]);
int target = 56;
int result = linearSearch(data, size, target);
if(result != -1)
    cout << "Element found at index " << result << endl;
else
    cout << "Element not found in the array." << endl;
return 0;
}
```

#### References:

GeeksforGeeks. (2025, September 4). *Binary search tree*. GeeksforGeeks. <https://www.geeksforgeeks.org/dsa/binary-search-tree-data-structure/>

---

#### **7. Supplementary Activity**

---

---

#### **8. Conclusion**

---

---

#### **9. Assessment Rubric**

---