

KJ's Educational Institutes  
**K J College Of Engineering & Management Research, Pune.**  
**Department of E & TC**

---

**CLASS: S. E. (E &TC)**

**SUBJECT:-DSA**

**Ex. No: 7**

**Date:**

**AIM**

**Searching Techniques**

You are building a contact manager app. A user wants to search for a contact either by scanning one by one or by using a fast lookup if the list is sorted.

Write a program that locates a specific name using both sequential and binary search techniques.

**OBJECTIVES**

To build a simple contact manager where a user can search for a name in a contact list using

- Sequential (Linear) Search if the list is unsorted
- Binary Search if the list is sorted

**THEORY**

Searching is one of the most fundamental operations in computer science. It refers to the process of finding the presence and location of a given element (called the **key**) within a collection of data. In this experiment, we are building a simple **Contact Manager App**, where a user wants to find a contact name in the list.

There are two main searching approaches covered here:

**1. Sequential (Linear) Search**

Sequential search is the simplest searching technique. It works by comparing each element in the list with the target element one by one until the desired element is found or the list ends.

**2. Binary Search**

Binary search is a fast searching algorithm that works only on **sorted lists**. It uses the divide-and-conquer principle to reduce the search space by half in every step.

**ALGORITHM**

**Sequential Search Algorithm**

1. Start.
2. Input the list of contact names.
3. Input the name to search.

4. Compare each name one by one with the target.
5. If a match is found, return the position.
6. If end of list is reached without a match, return "not found".
7. End.

### **Binary Search Algorithm (for sorted list)**

1. Start.
2. Input the sorted list of contact names.
3. Input the name to search.
4. Initialize low = 0, high = n-1.
5. Repeat until low <= high:
  - o Find mid = (low + high) / 2.
  - o If target == list[mid], return position.
  - o If target < list[mid], search in left half (high = mid - 1).
  - o Else, search in right half (low = mid + 1).
6. If not found, return "not found".
7. End.

### **Input:**

- No. of contacts
- Contact names
- Name to search

### **Output:**

Enter number of contacts: 5

Enter 5 contact names:

Vikas Vinod Sagar Vishal Amit

Enter name to search: Sagar

Sequential Search: Charlie found at position 3

Binary Search: Charlie found at position 3

### **CONCLUSION:-**