

Contact Management Using Binary Search

DSA mini project

Austin Powers Newton(RA2211030010057)

M.Mathesh(RA2211030010053)

Yogesh(RA2211030010064)

C.Saravana(RA2211030010018)

Introduction

- ▶ Contact Management Using Binary Search is a project that involves designing and implementing a digital system for managing and organizing contact information, such as names, phone numbers, email addresses, and other relevant details, using the binary search algorithm. This project is particularly useful for individuals and organizations that need an efficient way to store, retrieve, and update their contact lists.

Introduction to BINARY SEARCH

- ▶ A Binary Search project typically involves the implementation of the binary search algorithm in a specific context or problem-solving scenario. Binary search is a widely used algorithm for efficiently finding a specific element in a sorted array or list. Here's an introductory overview of a Binary Search project

Abstract:

- ▶ The "Contact Management Using Binary Search" project presents a streamlined solution for managing contact information efficiently. By employing the binary search algorithm, it offers rapid contact retrieval, addition, updating, and deletion while maintaining a sorted contact list. The user-friendly interface, security measures, and data persistence enhance the project's utility. This practical application of binary search provides an organized and efficient contact management system for various users.

Challenges for CONTACT MANAGEMENT

- ▶ Managing contact information effectively in a Contact Management system presents challenges such as ensuring data integrity and security, optimizing search efficiency, standardizing data, handling user permissions, and addressing scalability and synchronization issues, among others. Successfully addressing these challenges is essential for providing a user-friendly and efficient contact management solution.

Implement the BINARY SEARCH

- ▶ In a Contact Management system implementing binary search, contact data is organized in a sorted structure, and the binary search algorithm is used for efficient contact retrieval. Contacts are typically stored in an array or list, sorted by a specific criterion like names. When a user searches for a contact, binary search narrows down the search space, providing faster and more efficient results. This enables quick and accurate access to contact information while maintaining an ordered contact list.

Operations

- ▶ Adding Contacts: Allow users to add new contact entries to the system. Ensure that the new contact is inserted at the appropriate position in the sorted list to maintain the order.
- ▶ Searching for Contacts: Provide the ability to search for contacts based on specific criteria such as names, phone numbers, or email addresses. Use binary search to quickly locate and retrieve the desired contact information.
- ▶ Updating Contacts: Enable users to update existing contact information. When a contact is updated, ensure that it remains in its correct sorted position within the list.
- ▶ Deleting Contacts: Allow users to remove unwanted contact entries from the system. Ensure that the deletion process maintains the sorted order of the remaining contacts.

Limitation

- ▶ Requirement for Sorted Data: The system assumes data is consistently sorted, making it less suitable for unsorted or frequently changing data.
- ▶ Inefficiency for Small Datasets: Binary search's efficiency may not be as pronounced for small contact lists, as sorting overhead can outweigh benefits.
- ▶ Handling Duplicates and Real-Time Collaboration: Dealing with duplicates and enabling real-time collaboration can be complex in binary search-based systems.

Output

```
Options:
1. Add a contact
2. Search for a contact
3. Display contacts in alphabetical order
4. Exit
Enter your choice: 2
Enter the name to search: YOGESH
Phone number for YOGESH: 1122-3344
```

```
Options:
1. Add a contact
2. Search for a contact
3. Display contacts in alphabetical order
4. Exit
Enter your choice: 4
PS C:\Users\ austi\Desktop> █
```

```
PS C:\Users\ austi> cd "c:\Users\ austi\Desktop\" ; if ($?) { gcc DSA.c -o DSA } ; if ($?) { .\DSA }

Options:
1. Add a contact
2. Search for a contact
3. Display contacts in alphabetical order
4. Exit
Enter your choice: 1
Enter contact name: AUSTIN
Enter phone number: 1234-4321

Options:
1. Add a contact
2. Search for a contact
3. Display contacts in alphabetical order
4. Exit
Enter your choice: 1
Enter contact name: MATHESH
Enter phone number: 4321-1234

Options:
1. Add a contact
2. Search for a contact
3. Display contacts in alphabetical order
4. Exit
Enter your choice: 1
Enter contact name: YOGESH
Enter phone number: 1122-3344

Options:
1. Add a contact
2. Search for a contact
3. Display contacts in alphabetical order
4. Exit
Enter your choice: 3
Contacts in alphabetical order:
Name: AUSTIN, Phone: 1234-4321
Name: MATHESH, Phone: 4321-1234
Name: YOGESH, Phone: 1122-3344
```

CONCLUSION

- ▶ the code offers a strong foundation for a contact management system using a Binary Search Tree, ensuring efficient organization and retrieval of contact information. While suitable for educational and basic use, real-world applications would necessitate more features and a user-friendly interface. With further improvements, it could become a valuable component in address book apps, business contact management systems, or any situation requiring organized and scalable contact storage and retrieval.



THANK YOU!