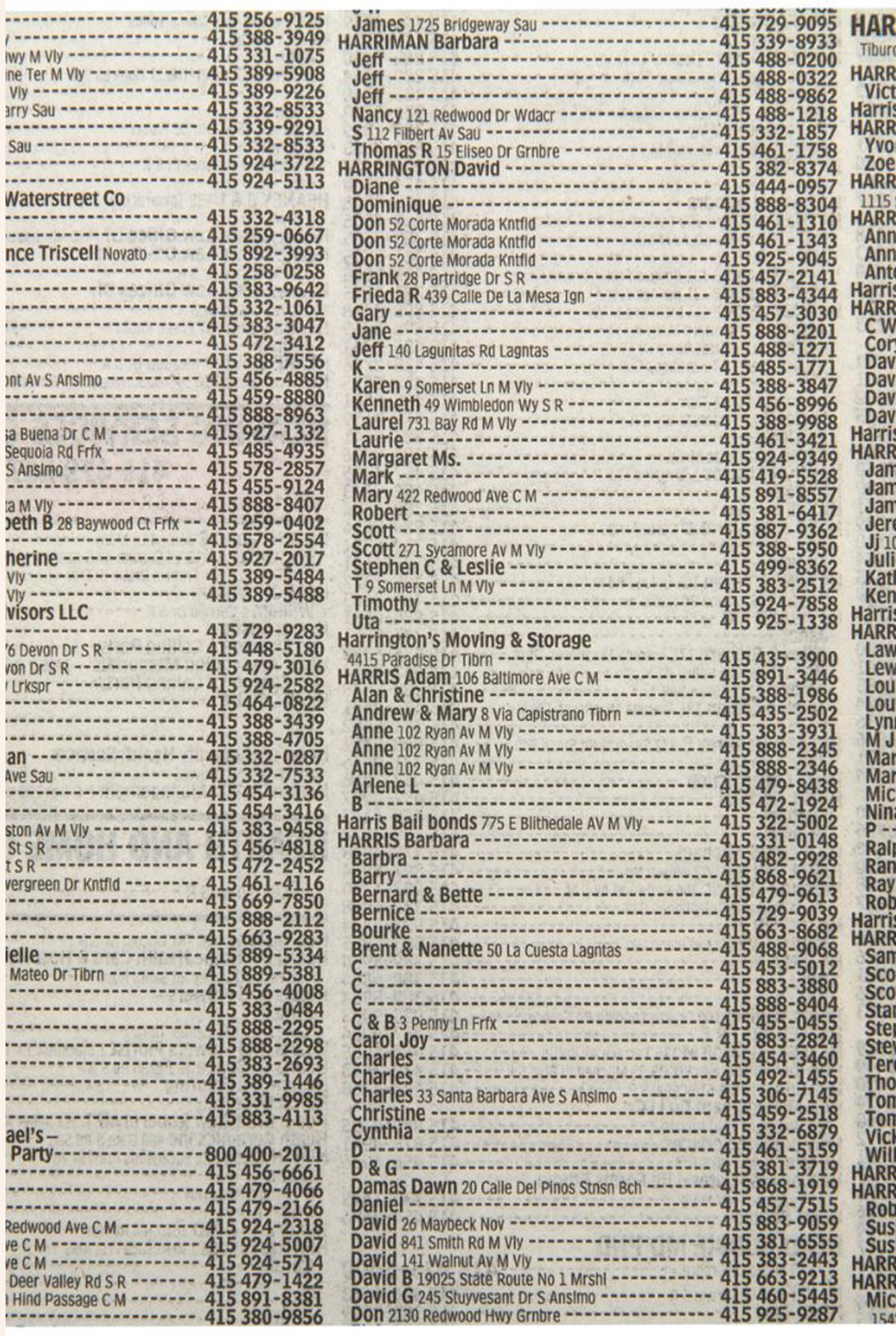
Phone Directory using Doubly Linked List

This presentation explores the concept of a phone directory implemented using a doubly linked list, discussing its structure, operations, memory efficiency, and real-life use cases.

by Gautam Taneja(RA2211003010812)

Amithrajith Premesh (RA2211003010811)

Anirudhan Mani

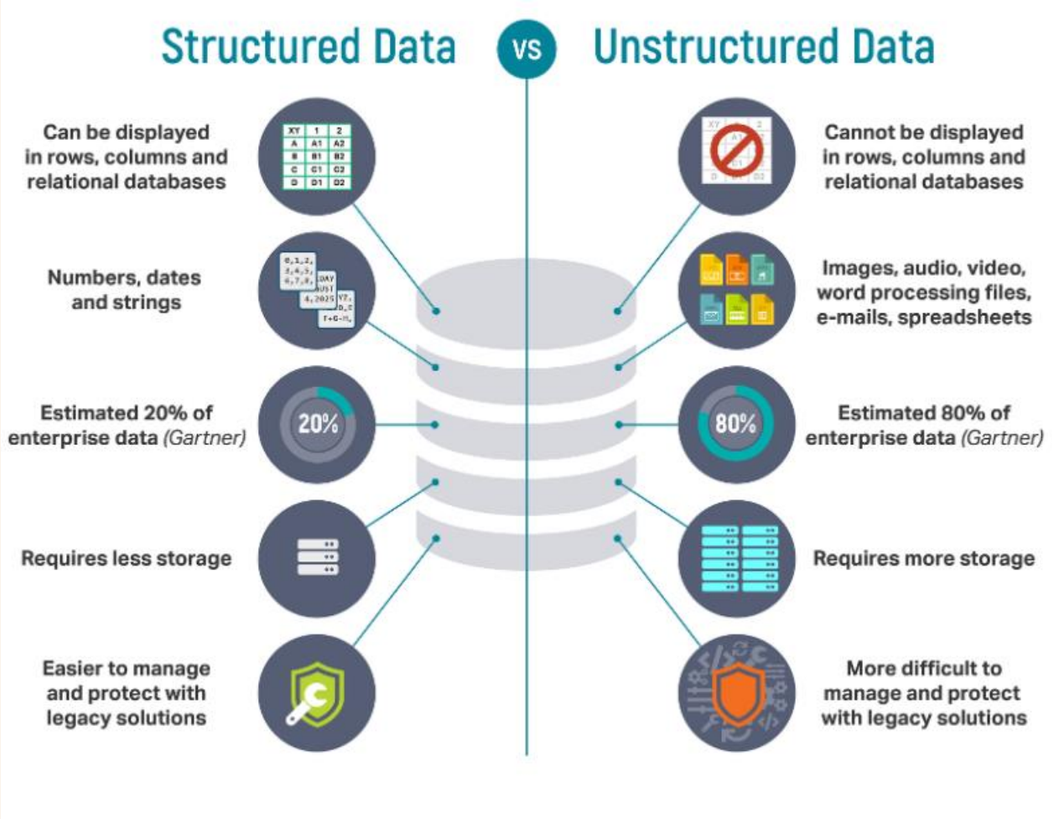


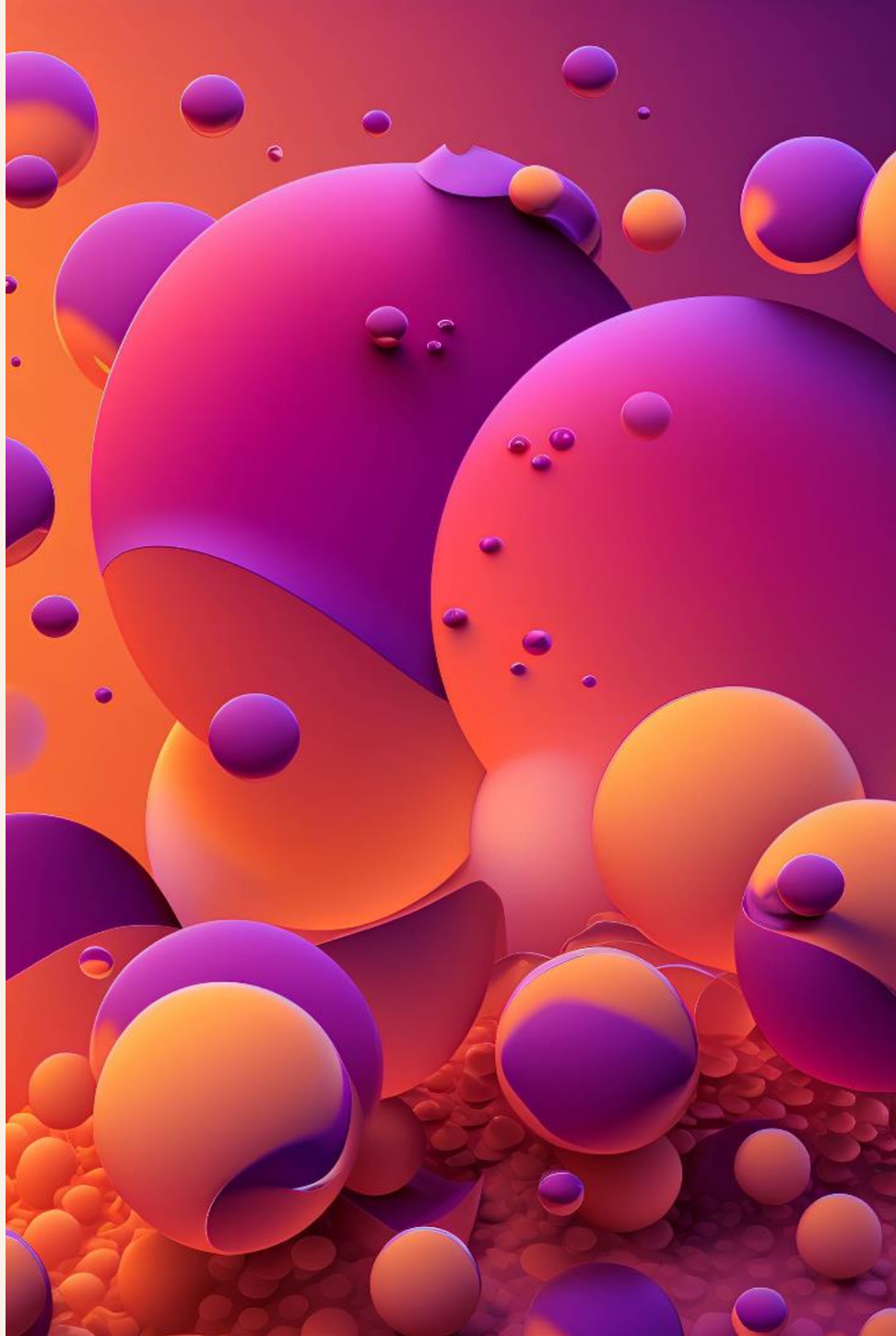
Introduction

Welcome to the world of phone directories! In this presentation, we'll deep dive into dive into the fascinating world of doubly linked lists and how they can be leveraged to leveraged to build an efficient and practical phone directory system.

Data Structures Overview

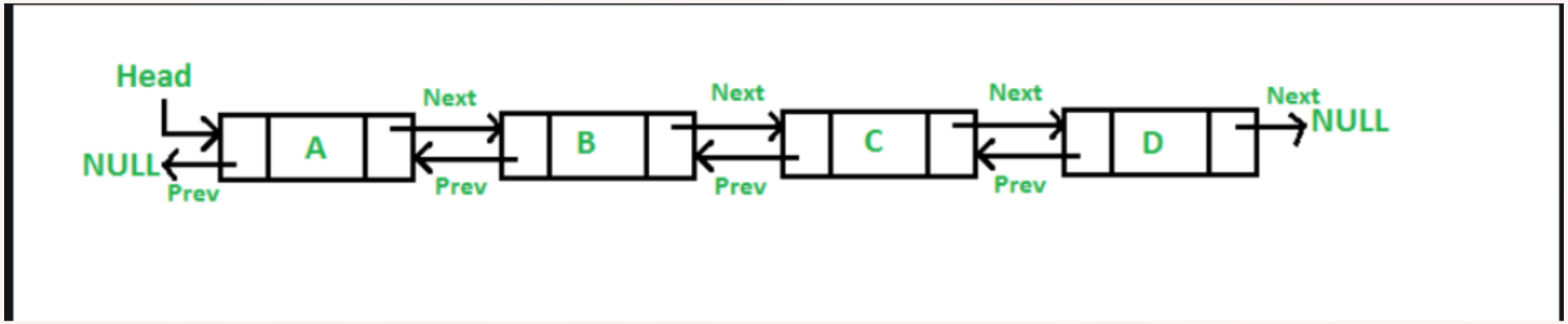
Before we dive into the details, let's take a quick look at various data structures structures commonly used in computer science and how they differ from one from one another in terms of storing and organizing data efficiently.





Doubly Linked Lists

Explore the power of doubly linked lists, a data structure that allows us to efficiently navigate both forward and forward and backward through a sequence of elements. Discover why they are the perfect fit for implementing a implementing a phone directory.



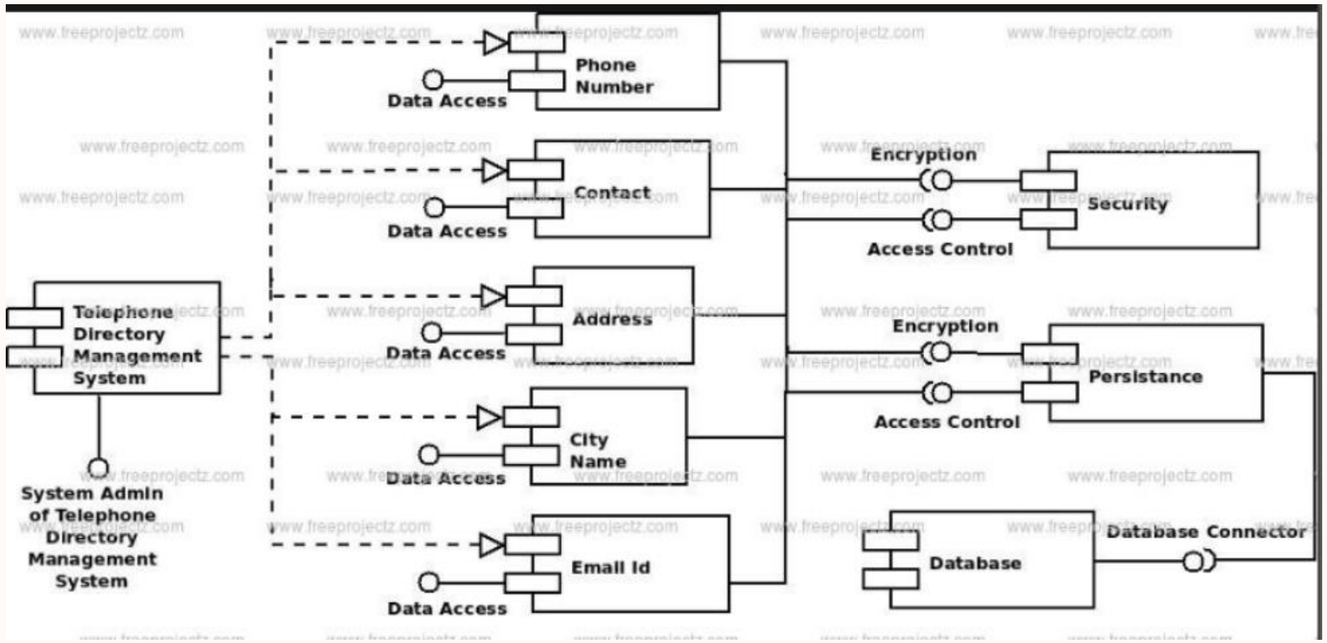
Phone Directory Structure

Delve into the inner workings of a phone directory implemented using a doubly linked list. Learn how each entry is linked to the previous and next entries, creating a flexible and scalable structure for storing contacts.

```
**************                                  PHONE BOOK                                  ********************

WHAT IS YOUR NAME?
Gautam taneja
!!!!!!!!!!!!!!!!!!!!!!   WELCOME Gautam taneja   !!!!!!!!!!!!!!!!!!!!!

LET'S CREATE OUR PHONEBOOK Gautam taneja

ENTER NAME      :Dad
ENTER NUMBER    :9891151155
ENTER G-MAIL    :apt1965@gmail.com
DO YOU WANT TO CONTINUE?????????yes
ENTER NAME      :ENTER NUMBER    :
```

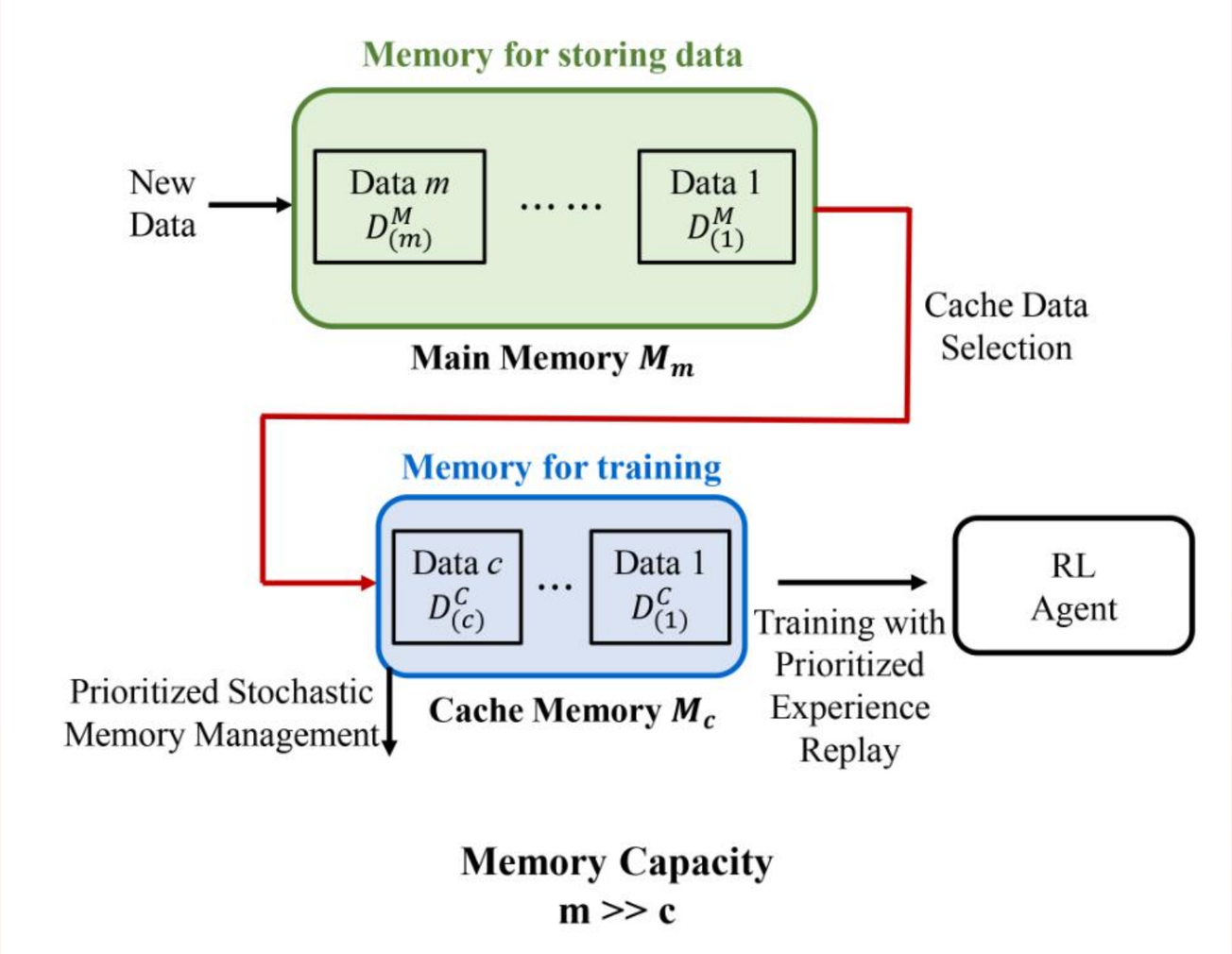
Operations on a Phone Directory

Unlock the potential of a phone directory by understanding how various operations, such as adding, removing, and searching for contacts, can be efficiently performed using the power of doubly linked lists.



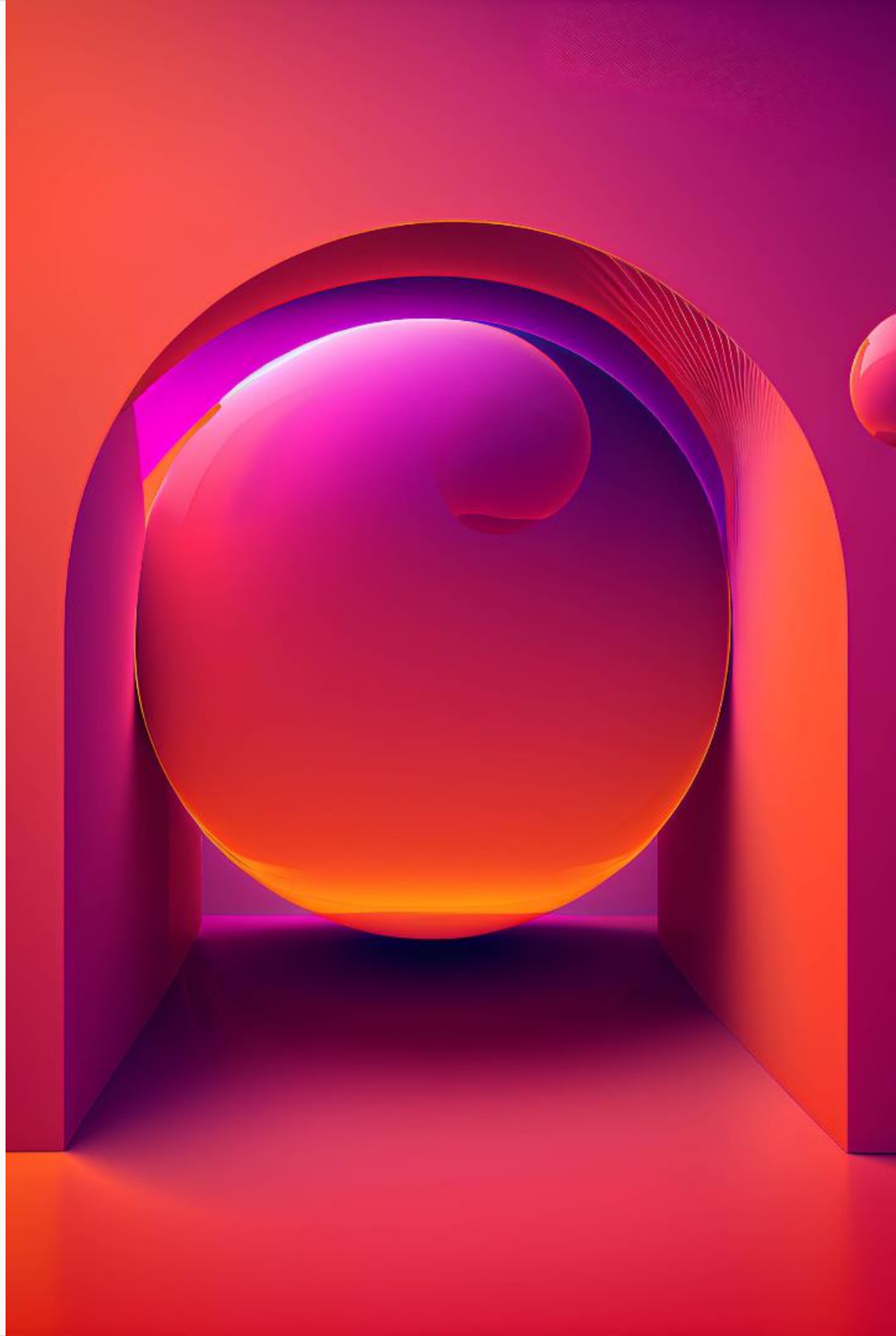
Memory Efficiency

Discover the memory-efficient nature of a phone directory implemented using a doubly linked list. Explore how data can be stored, linked, and organized in a way that optimizes memory usage and minimizes wastage.



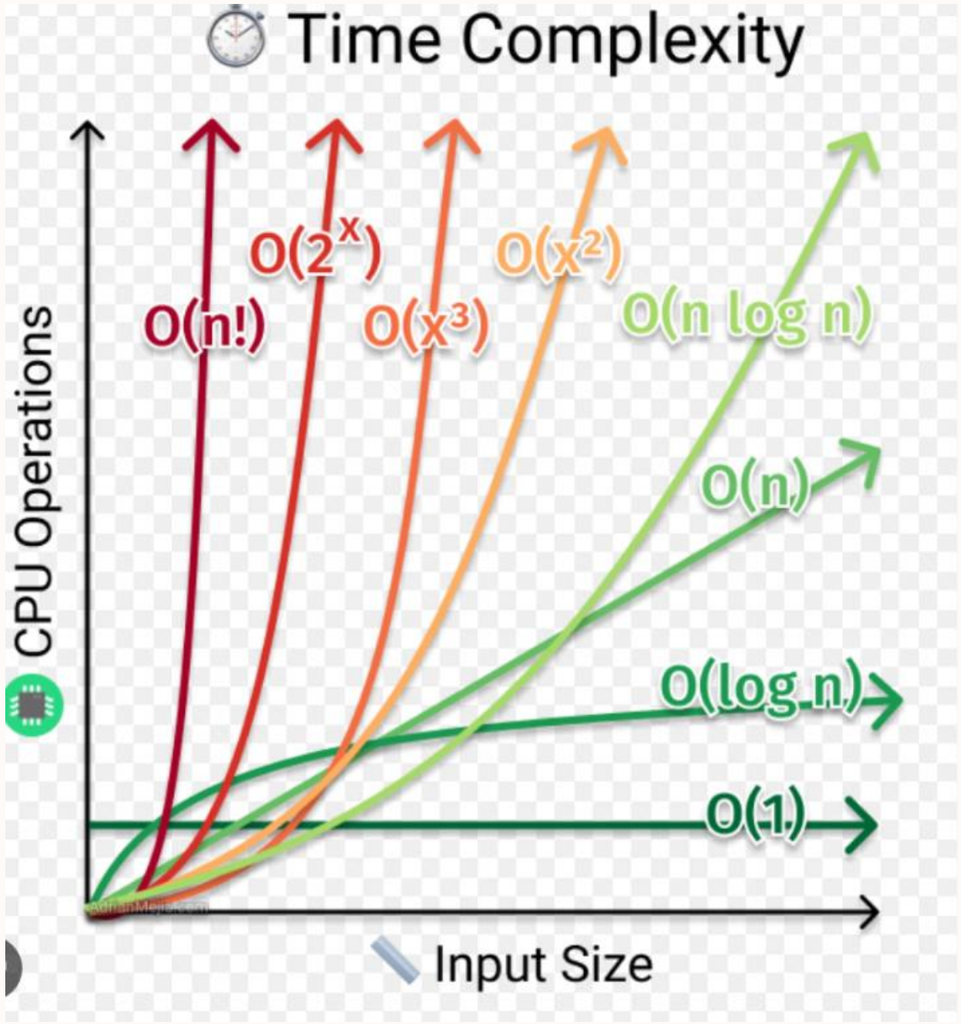
Implementation Details

Uncover the nitty-gritty details of implementing a phone directory using a doubly linked list. Dive into the code, discuss the required data structures, and explore the algorithms utilized for performing various operations on the directory.



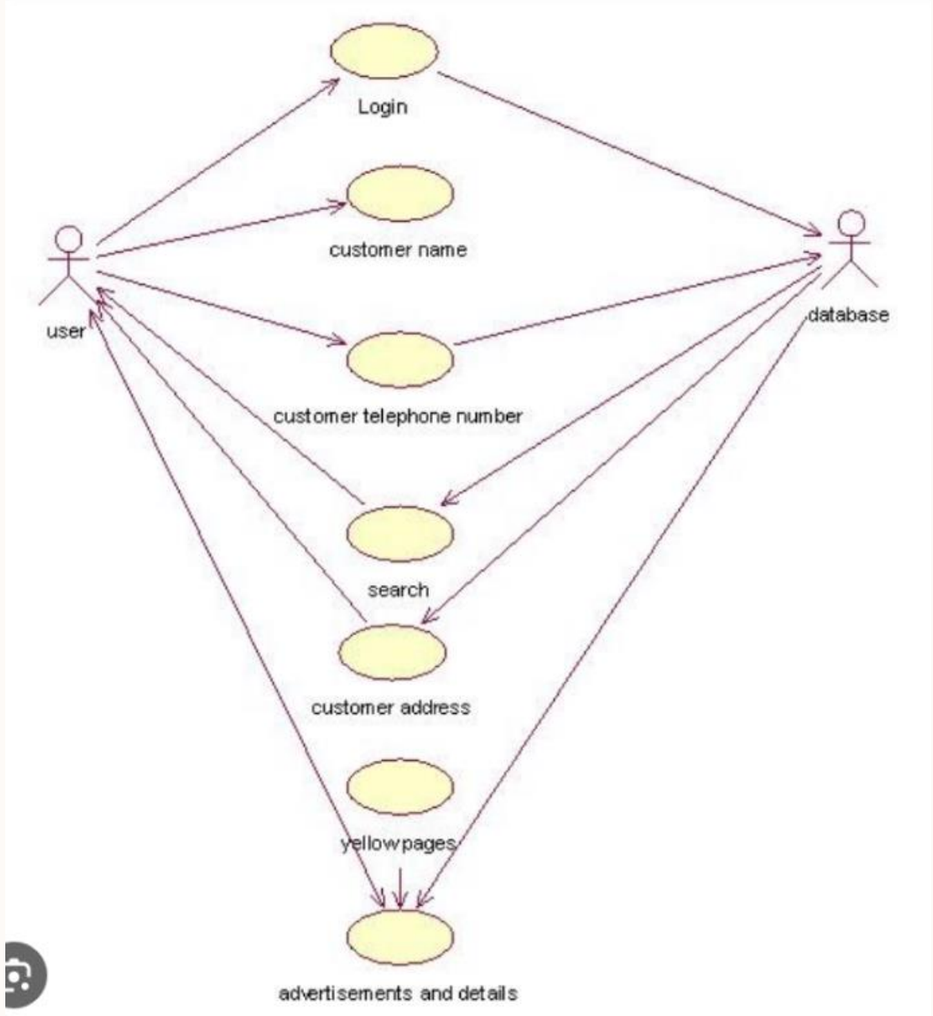
Time Complexity Analysis

Get ready for some algorithmic analysis! We'll evaluate the time complexity of the operations performed on the phone directory implemented with a doubly linked list. Understand the trade-offs and efficiency of each operation.



Real-Life Use Cases

Discover the wide range of real-life scenarios where a phone directory implemented using a doubly linked list shines. From contact management apps to customer databases, explore the myriad of applications where this data structure comes into play.



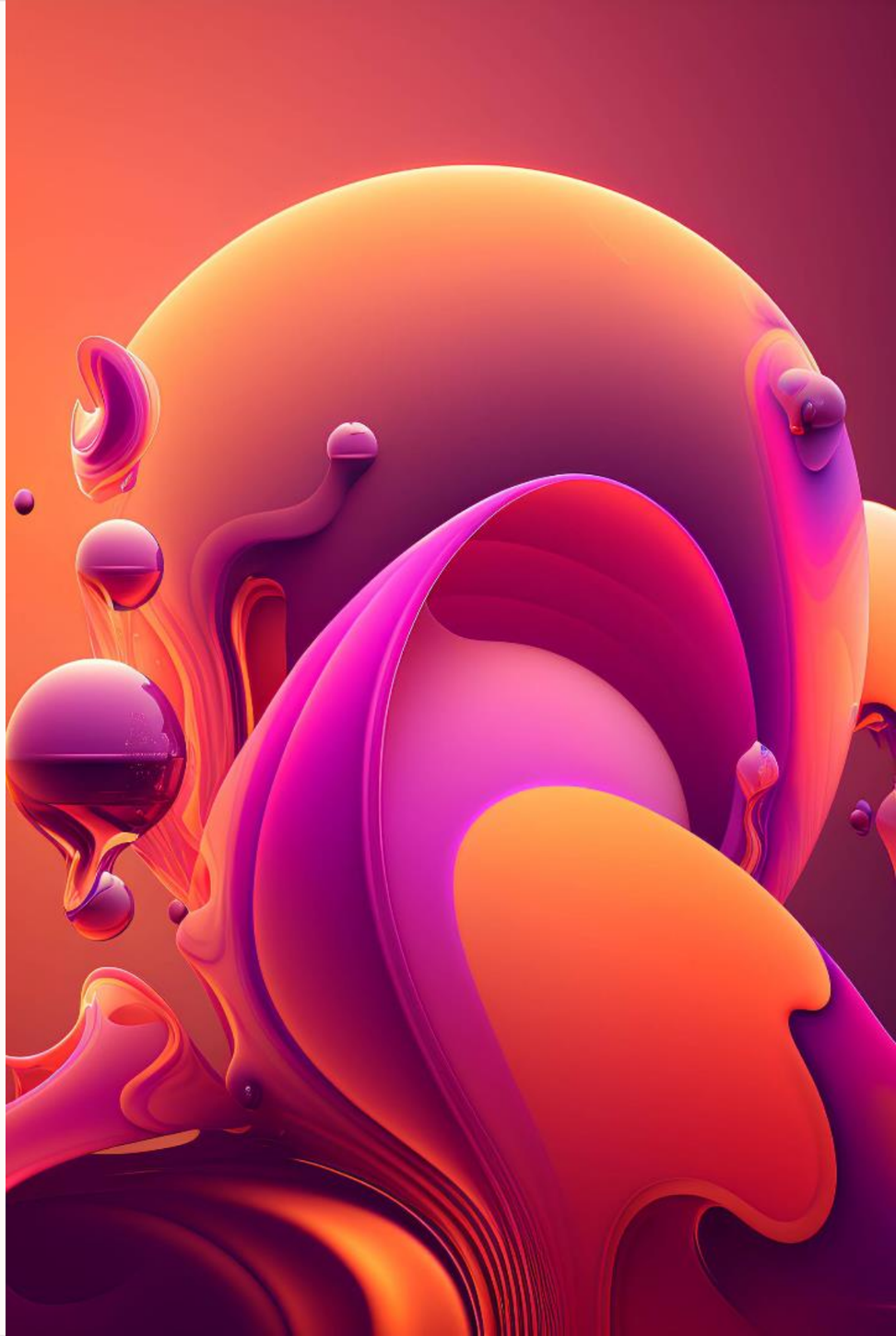
Code Examples

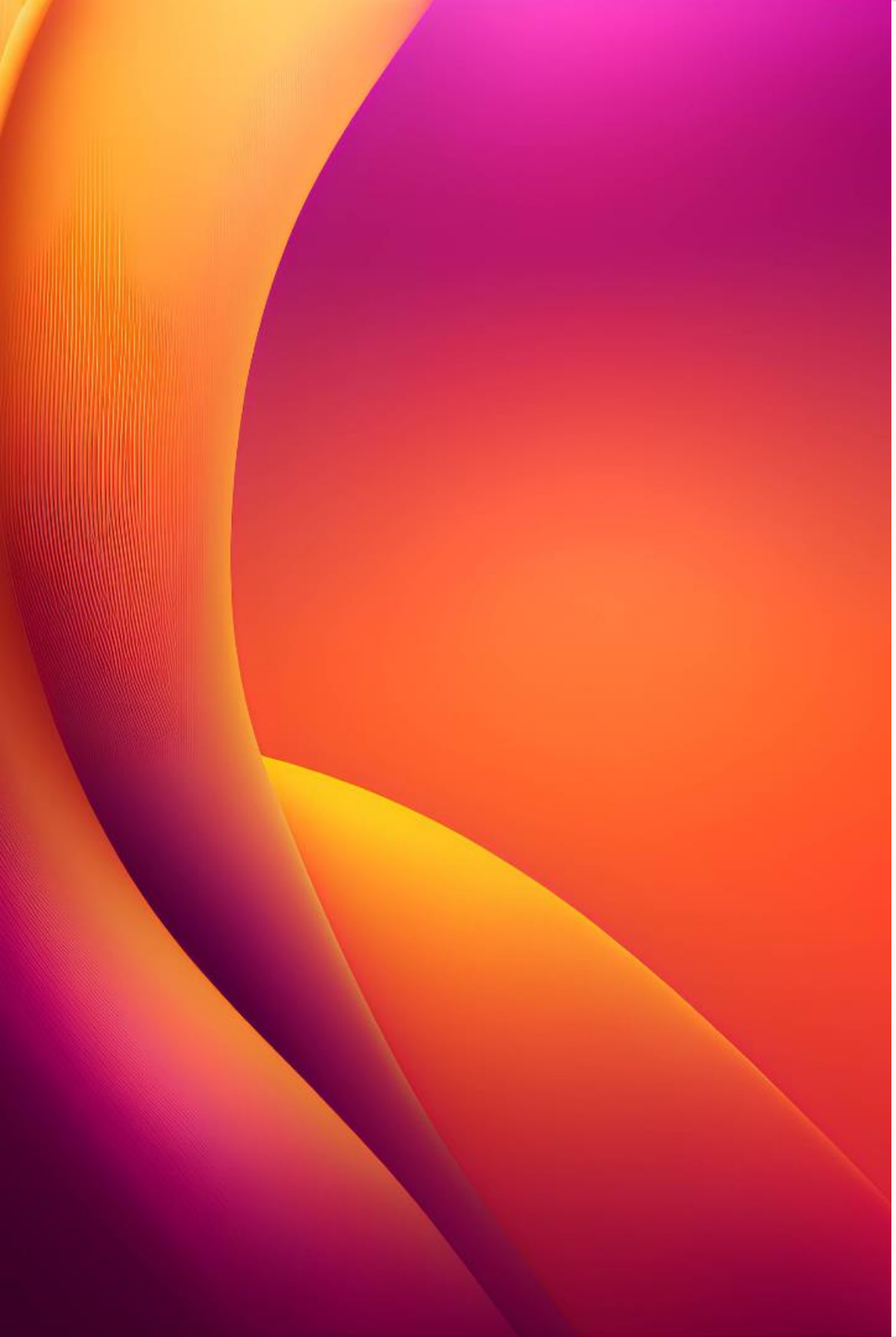
A picture is worth a thousand words, but code examples are priceless. Witness live code snippets that demonstrate the power and simplicity of implementing a phone directory using a doubly linked list.

```
cin.getline(name,20);
cout<<"\n\n!!!!!!!!!!!!!!!!!!!!!!    WELCOME "<<name<<"
    !!!!!!!!!!!!!!!!!!!!!";
cout<<"\n\n\nLET'S CREATE OUR PHONEBOOK "<<name<<"  \n\n";
d1.accept();
d1.sort();
do
{
cout<<"\n\n\n\n1) DISPLAY YOUR PHONE BOOK\n2) INSERT NEW
    CONTACT\n3) UPDATE DETAILS ON EXISTING CONTACT\n4)
    DELETE CONTACT\n5) DELETE SAME NAME IN PHONEBOOK\n6)
    DELETE SAME NUMBERS IN PHONEBOOK\n7) SEARCH\n";
cin>>ch;
switch(ch)
{
case 2:
d1.insert();
d1.sort();
break;

case 1:
// d1.sort();
d1.display();
break;
case 3:

    cout<<"\n\nENTER THE NAME OF PERSON WHOSE DETAILS YOU
        WANT TO UPDATE...\n";
```

Challenges and Optimizations

Every implementation has its challenges. We'll dive into the various obstacles faced when working with a phone directory using a doubly linked list and discuss clever optimizations to enhance its performance.





Conclusion

As we conclude our journey into the world of phone directories and doubly linked lists, reflect on the knowledge gained and the endless possibilities this data structure brings. Build your own phone directory and unlock new realms of efficiency!

Q&A Session

It's time to address any lingering doubts or questions you may have. Ask away and let's explore further together!