

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

James 1725 Bridgeway Sau	415 729-9095	HARR
HARRIMAN Barbara	415 339-8933	Tiburo
Jeff	415 488-0200	HARR
Jeff	415 488-0322	Vict
Jeff	415 488-9862	Harris
Nancy 121 Redwood Dr Wdacr	415 488-1218	HARR
S 112 Filbert Av Sau	415 332-1857	Yvo
Thomas R 15 Eliseo Dr Grnbre	415 461-1758	Zoe
HARRINGTON David	415 382-8374	HARR
Diane	415 444-0957	1115
Dominique	415 888-8304	HARR
Don 52 Corte Morada Kntfld	415 461-1310	An
Don 52 Corte Morada Kntfld	415 461-1343	Ann
Don 52 Corte Morada Kntfld	415 925-9045	Anto
Frank 28 Partridge Dr S R	415 457-2141	Harris
Frieda R 439 Calle De La Mesa Ign	415 883-4344	HARR
Gary	415 457-3030	C W
Jane	415 888-2201	Cor
Jeff 140 Lagunitas Rd Lagntas	415 488-1271	Dav
K	415 485-1771	Dav
Karen 9 Somerset Ln M Vly	415 388-3847	Dav
Kenneth 49 Wimbledon Wy S R	415 456-8996	Harris
Laurel 731 Bay Rd M Vly	415 388-9988	HARR
Laurie	415 461-3421	Jam
Margaret Ms.	415 924-9349	Jam
Mark	415 419-5528	Jam
Mary 422 Redwood Ave C M	415 891-8557	Jer
Robert	415 381-6417	Jj 10
Scott	415 887-9362	Juli
Scott 271 Sycamore Av M Vly	415 388-5950	Kat
Stephen C & Leslie	415 499-8362	Ken
T 9 Somerset Ln M Vly	415 383-2512	Harris
Timothy	415 924-7858	HARR
Uta	415 925-1338	Law
Harrington's Moving & Storage		Lew
4415 Paradise Dr Tibrn	415 435-3900	Lou
HARRIS Adam 106 Baltimore Ave C M	415 891-3446	Lou
Alan & Christine	415 388-1986	Lyn
Andrew & Mary 8 Via Capistrano Tibrn	415 435-2502	M J
Anne 102 Ryan Av M Vly	415 383-3931	Mar
Anne 102 Ryan Av M Vly	415 888-2345	Mar
Anne 102 Ryan Av M Vly	415 888-2346	Nin
Arlene L	415 479-8438	p --
B	415 472-1924	Ralp
Harris Bail bonds 775 E Blithedale AV M Vly	415 322-5002	Ran
HARRIS Barbara	415 331-0148	Ray
Barbra	415 482-9928	Rob
Barry	415 868-9621	Harris
Bernard & Bette	415 479-9613	HARR
Bernice	415 729-9039	San
Bourke	415 663-8682	Sco
Brent & Nanette 50 La Cuesta Lagntas	415 488-9068	Sco
C	415 453-5012	Star
C	415 883-3880	Step
C	415 888-8404	Stev
C & B 3 Penny Ln Frfx	415 455-0455	Ter
Carol Joy	415 883-2824	Tho
Charles	415 454-3460	Tom
Charles	415 492-1455	Tom
Charles 33 Santa Barbara Ave S Anslmo	415 306-7145	Tom
Christine	415 459-2518	Vick
Cynthia	415 332-6879	Will
D	415 461-5159	HARR
D & G	415 381-3719	HARR
Damas Dawn 20 Calle Del Pinos Stnsn Bch	415 868-1919	Rob
Daniel	415 457-7515	Sus
David 26 Maybeck Nov	415 883-9059	Sus
David 841 Smith Rd M Vly	415 381-6555	HARR
David 141 Walnut Av M Vly	415 383-2443	HARR
David B 19025 State Route No 1 Mrshl	415 663-9213	Mic
David G 245 Stuyvesant Dr S Anslmo	415 460-5445	154
Don 2130 Redwood Hwy Grnbre	415 925-9287	

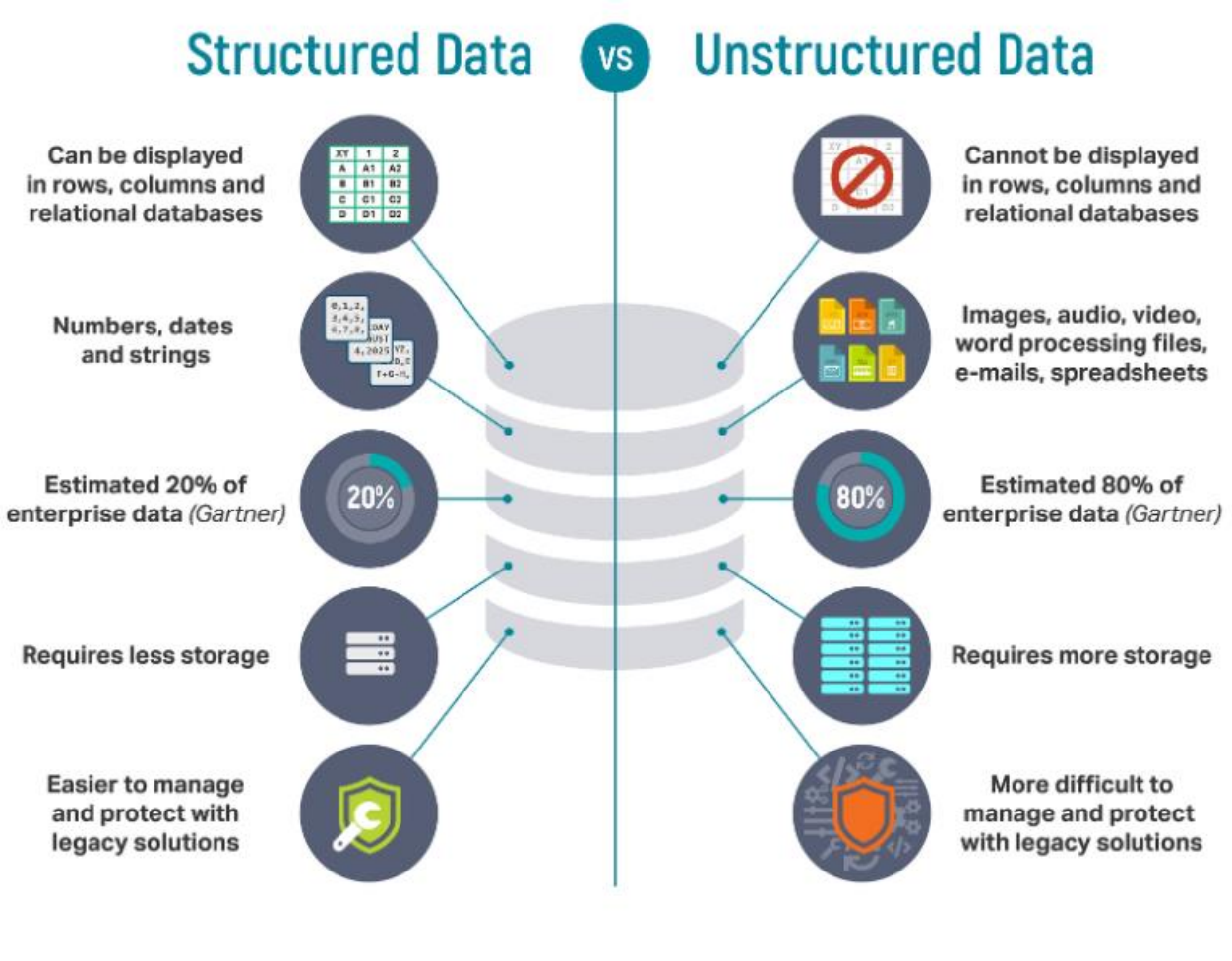
Introduction

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.



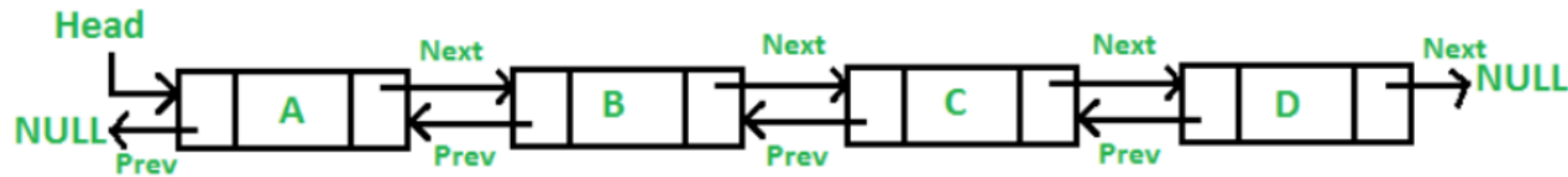
Data Structures Overview

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 backward through a sequence of elements. Discover why they are the perfect fit for 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                                     *****

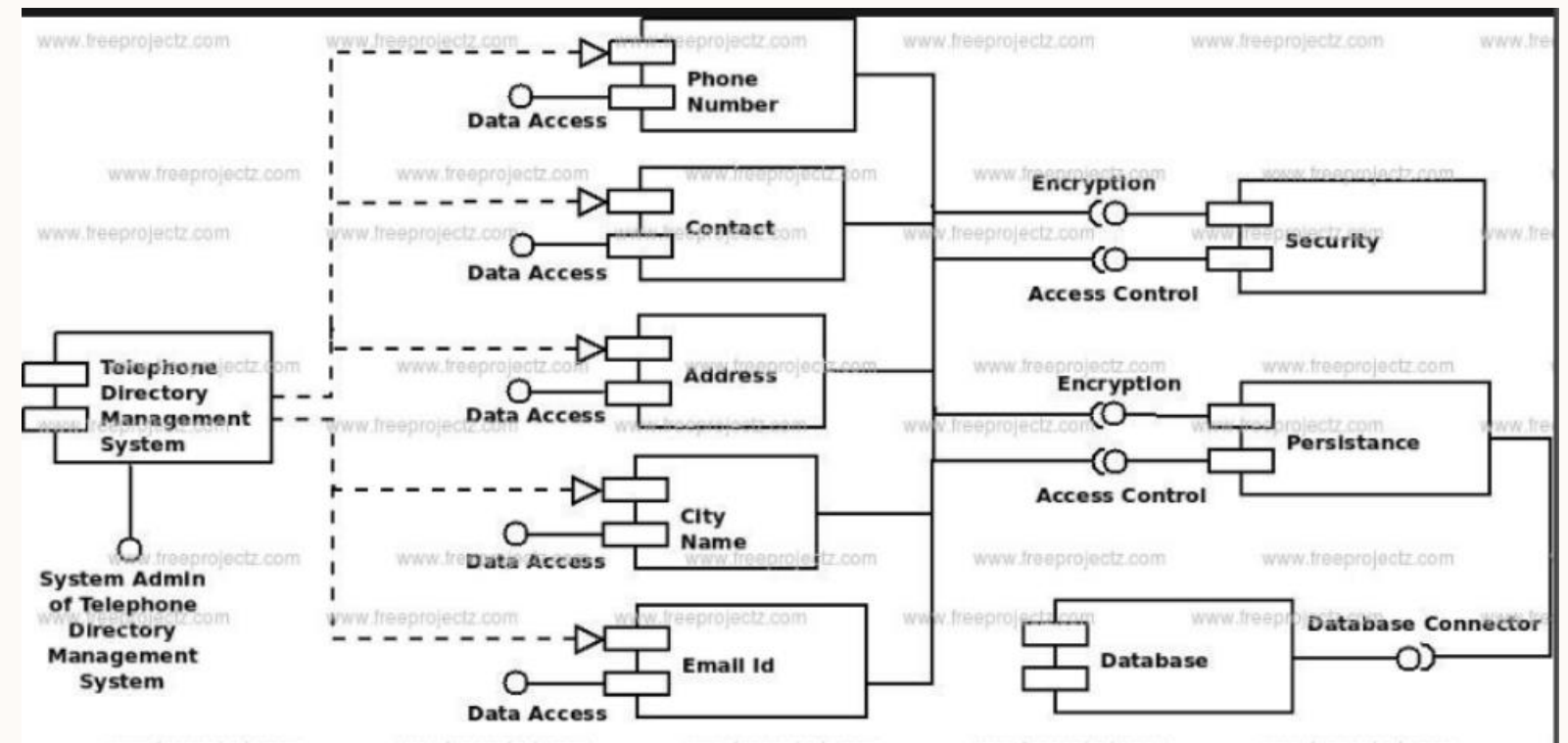
WHAT IS YOUR NAME?
Gautam taneja
!!!!!!!!!!!!!!!!!!!!!!!!!!!! WELCOME Gautam taneja !!!!!!!!!!!!!!!!!!!!!!!!!!!!!

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      :
```

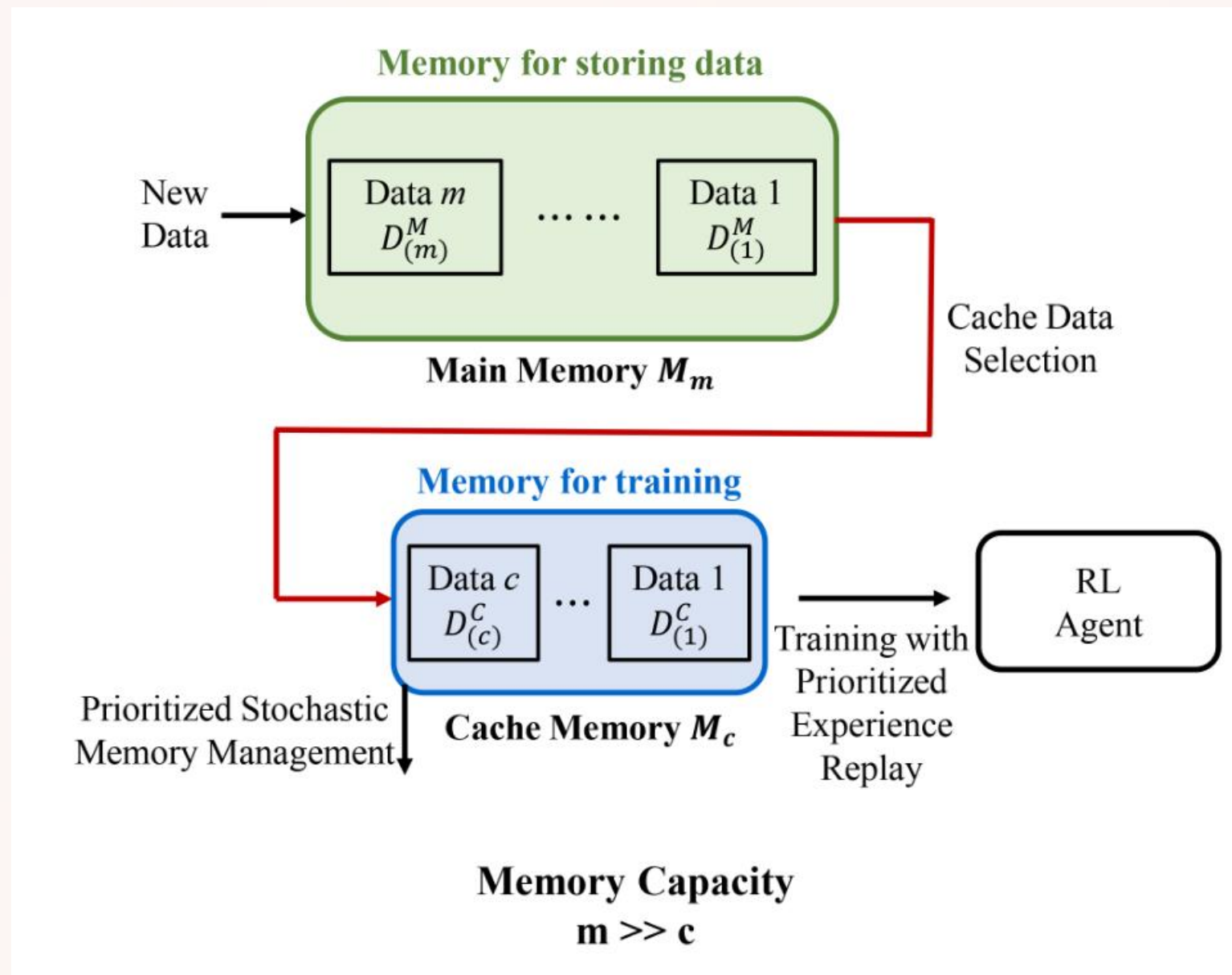
Operations on a Phone Directory

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.



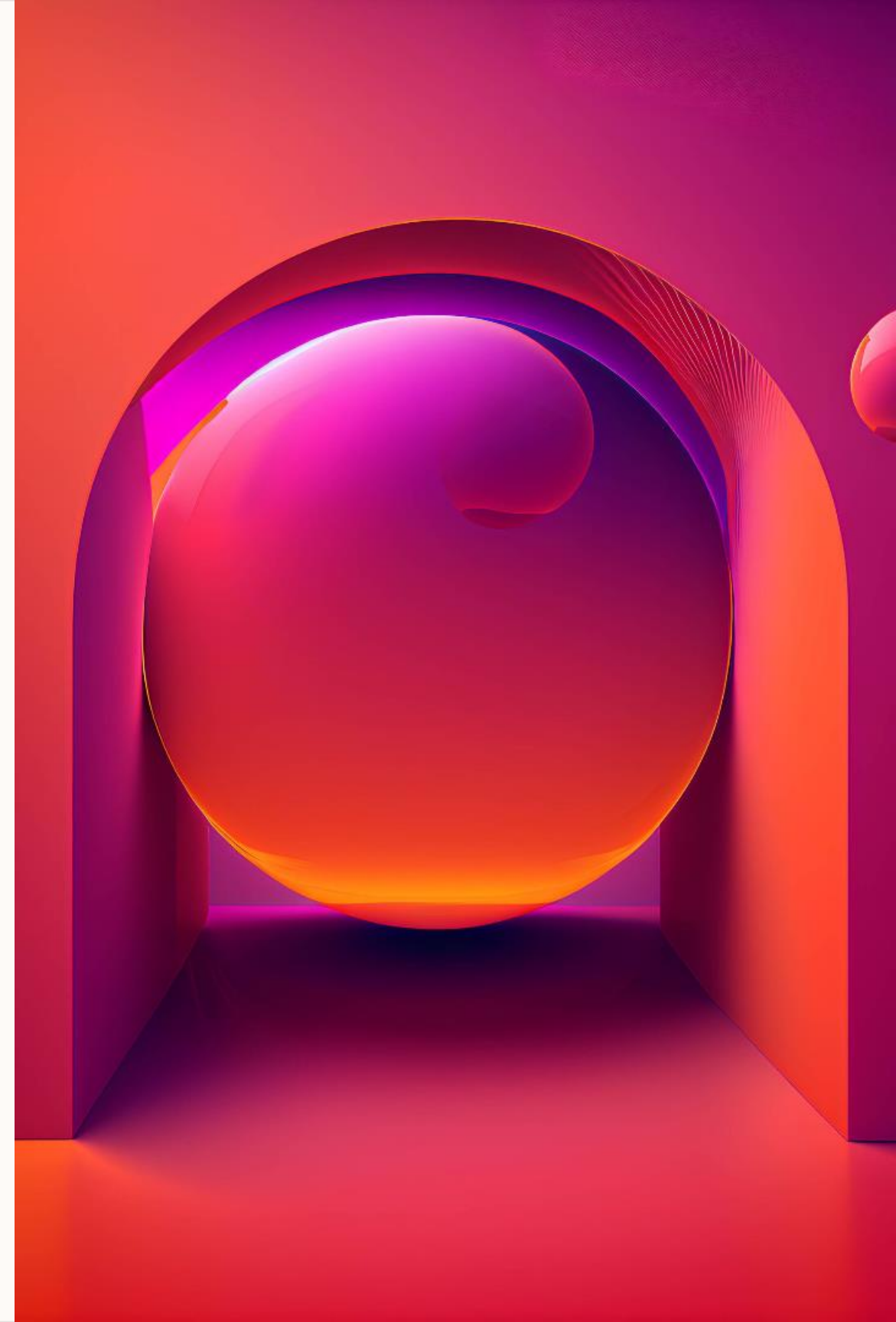
Memory Efficiency

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.



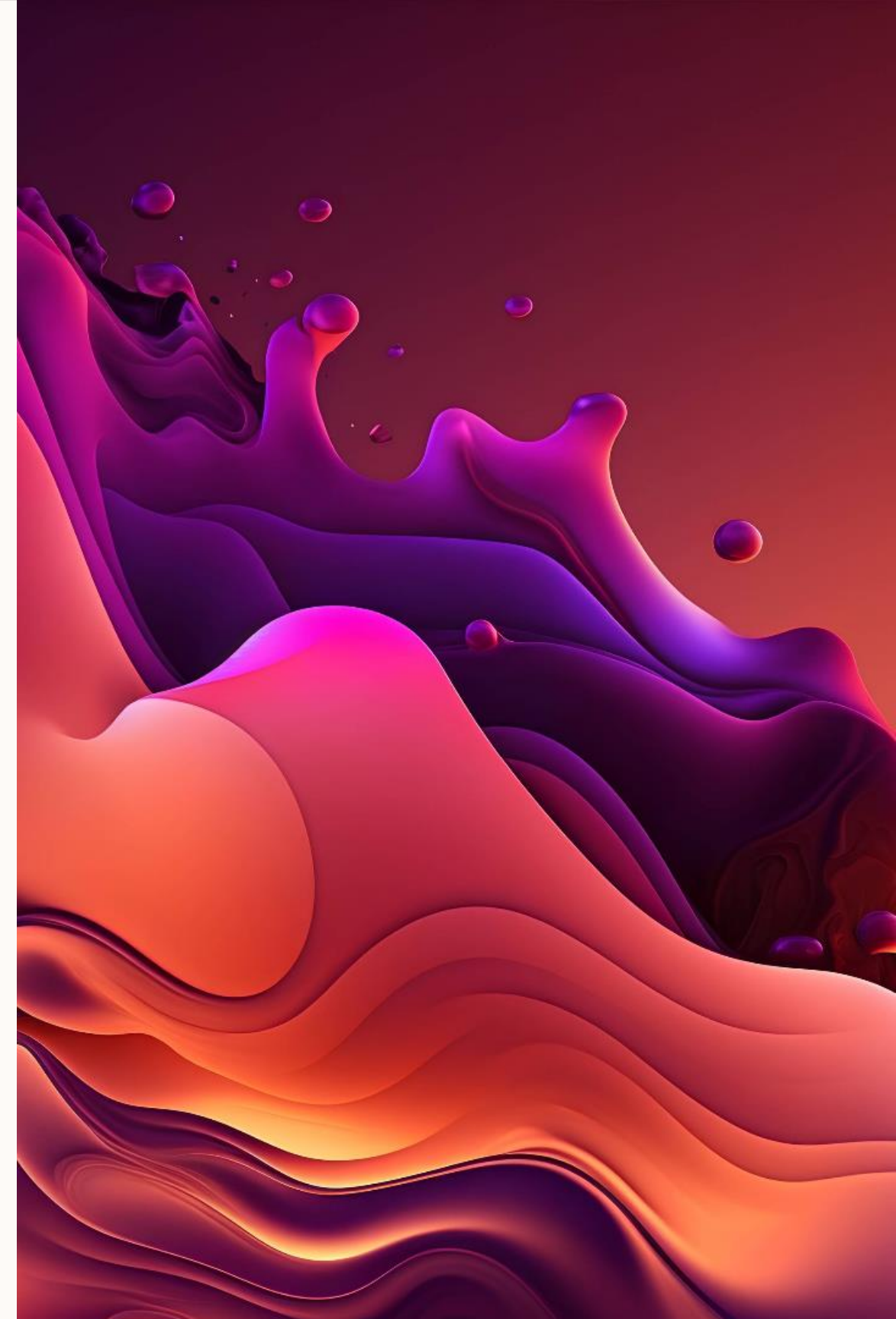
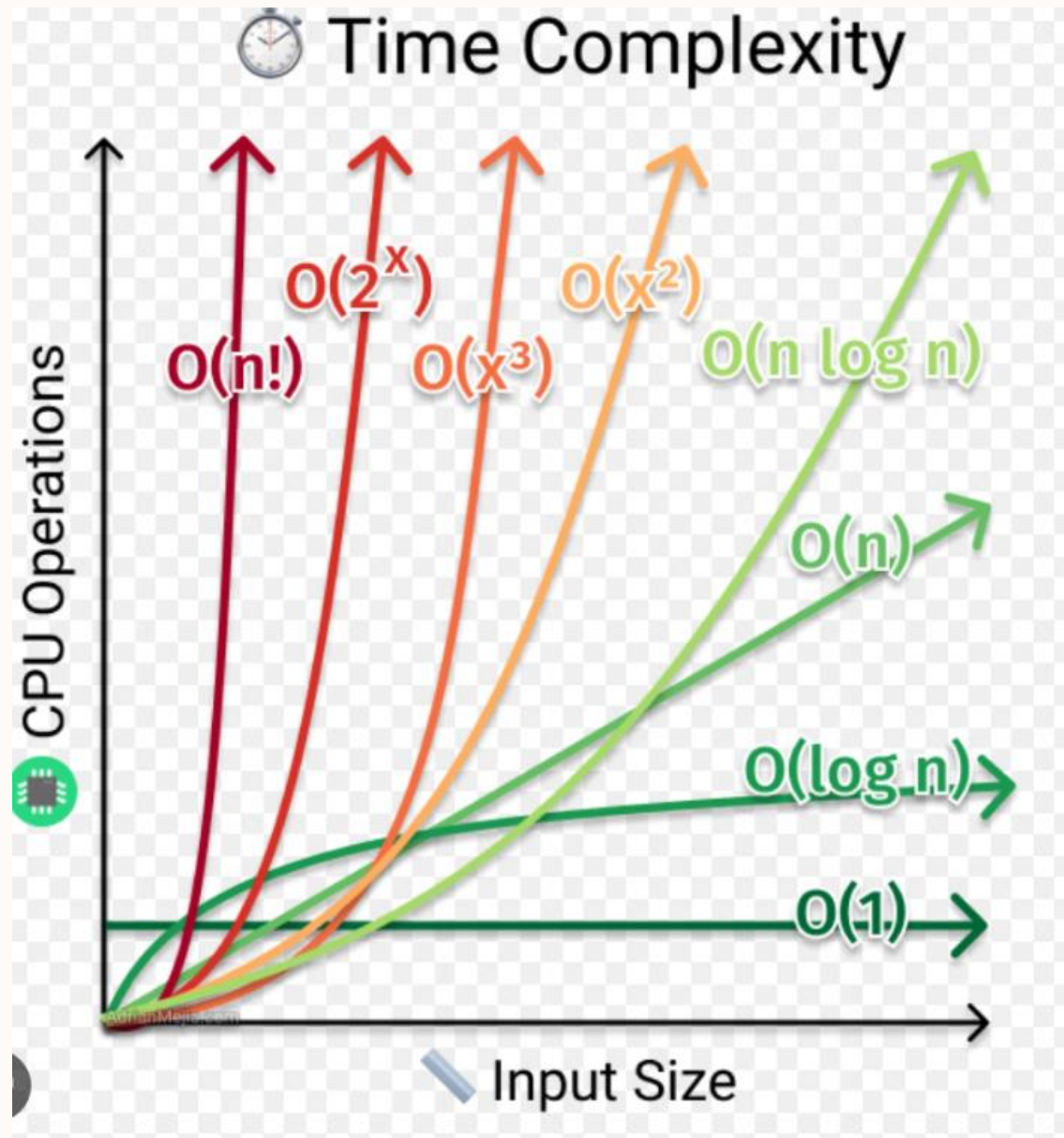
Implementation Details

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.



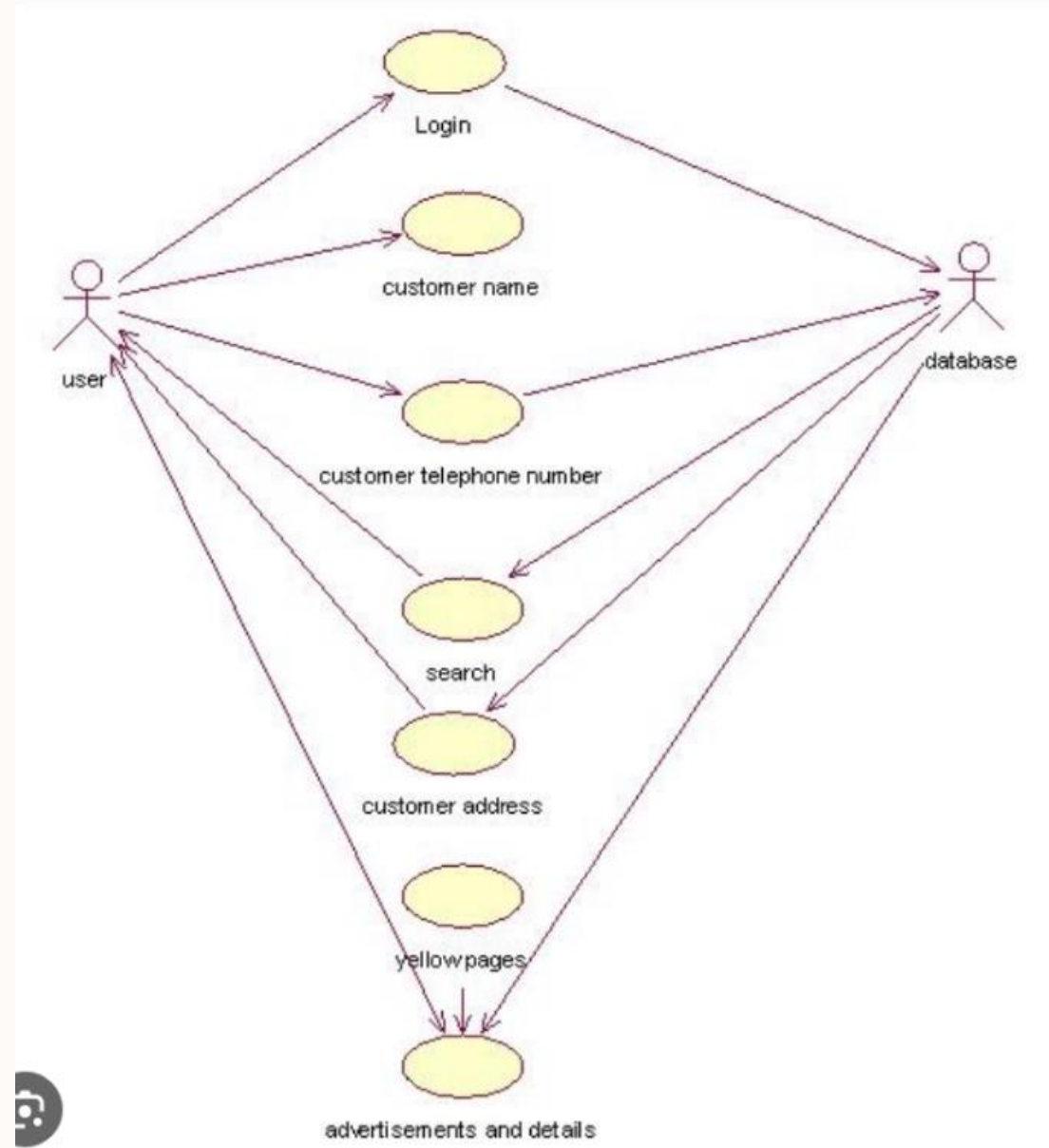
Time Complexity Analysis

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\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!