Study Point

Data Structures & Algorithms
(Binary Search & its Complexity)

Presented By:

Sh. Muhammad Aamir
Lecturer
Department of Computer Science
GC University, Faisalabad

Binary Search

Study Point

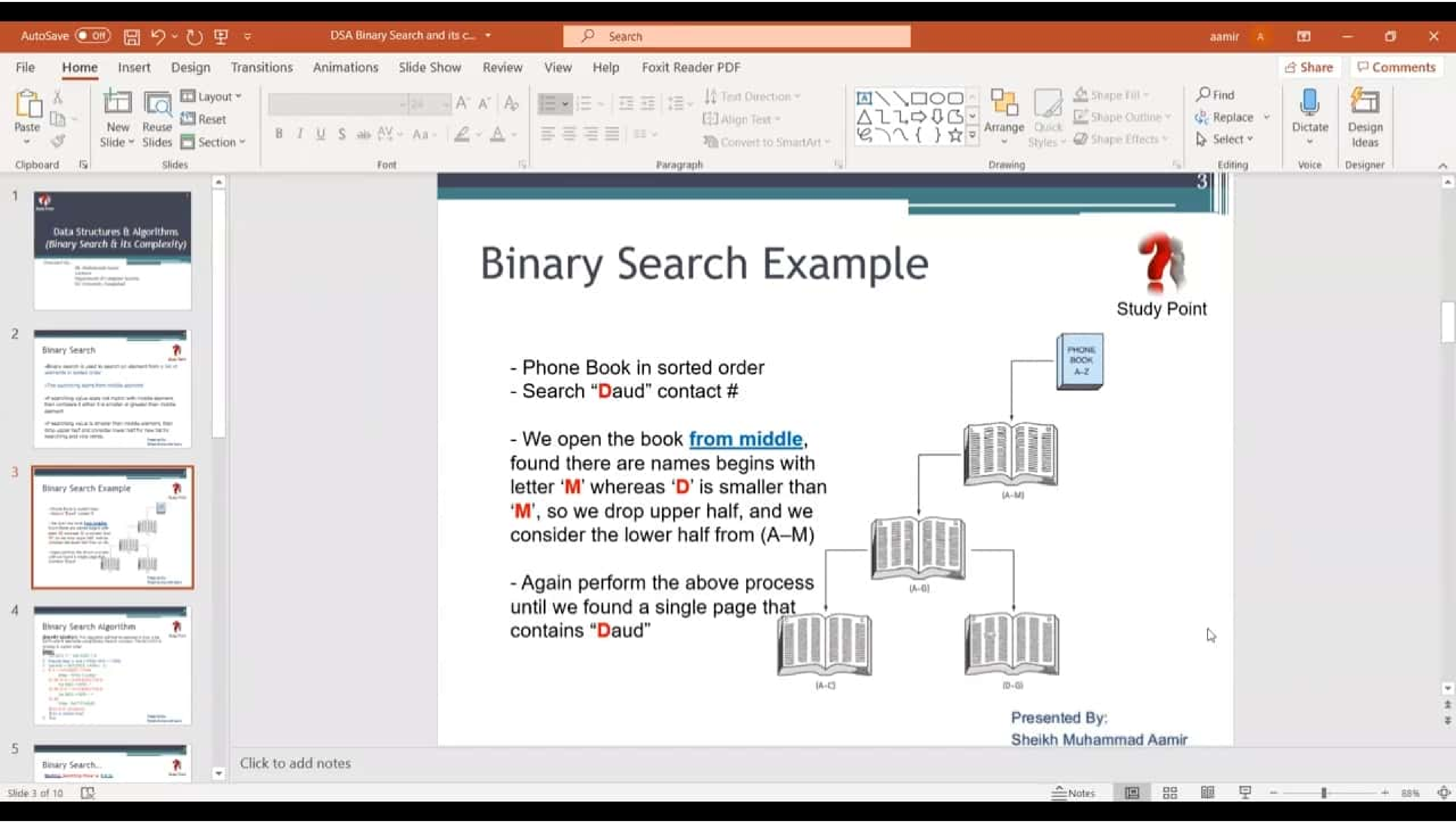
- Binary search is used to search an element from a list of elements in sorted order

- The searching starts from middle element

- If searching value does not match with middle element, then compare it either it is smaller or greater than middle element

- If searching value is smaller than middle element, then drop upper half and consider lower half for new list for searching and vice versa.

Presented By:
Sheikh Muhammad Aamir


DSA Binary Search and its c...
Binary Search Example
Study Point
- Phone Book in sorted order
- Search "Daud" contact #
- We open the book from middle, found there are names begins with letter 'M' whereas 'D' is smaller than 'M', so we drop upper half, and we consider the lower half from (A–M)
- Again perform the above process until we found a single page that contains "Daud"
PHONE BOOK A-Z
(A-M)
(A-G)
(A-C)
(D-G)
Presented By:
Sheikh Muhammad Aamir
Click to add notes
Slide 3 of 10

Binary Search...

Study Point

Working: Searching Value i.e. K = 40

	BEG				MID				END
	1	2	3	4	5	6	7	8	9	10
	5	10	15	20	25	30	35	40	45	50

MID = (1+10) / 2 = 5.5

1. Here K ≠ DATA [MID] i.e. 40 ≠ 25 But K > DATA[MID] so BEG = MID+1 i.e. BEG = 6

					BEG		MID		END
1	2	3	4	5	6	7	8	9	10
5	10	15	20	25	30	35	40	45	50

MID = (6+10) / 2 = 8

2. Here K = DATA [MID] i.e. 40 = 40, So searching value found at MID location

Presented By:
Sheikh Muhammad Aamir

Binary Search...

Study Point

Working: Searching Value i.e. K = 50

BEG (1) · MID (5) · END (10)

1	2	3	4	5	6	7	8	9	10
5	10	15	20	25	30	35	40	45	50

MID = (1+10) / 2 = 5.5

1. Here K ≠ DATA [MID] i.e. 50 ≠ 25, But K > DATA[MID] so BEG = MID+1 i.e. BEG = 6

BEG (6) · MID (8) · END (10)

1	2	3	4	5	6	7	8	9	10
5	10	15	20	25	30	35	40	45	50

MID = (6+10) / 2 = 8

2. Here K ≠ DATA [MID] i.e. 50 ≠ 40, Also K > DATA[MID] so BEG = MID+1 i.e. BEG = 9

MID, BEG (9) · END (10)

1	2	3	4	5	6	7	8	9	10
5	10	15	20	25	30	35	40	45	50

MID = (9+10) / 2 = 9.5

Presented By:

Shaikh Muhammad Aamir

Click to add notes

Binary Search...

Study Point

Working: **Searching Value i.e. K = 50**

MID
BEG
END

1	2	3	4	5	6	7	8	9	10
5	10	15	20	25	30	35	40	45	50

MID = (9+10) / 2
= 9.5

3. Here K ≠ DATA [MID] i.e. 50 ≠ 45, Also K > DATA[MID] so BEG = MID+1 i.e. BEG = 10

MID
BEG
END

1	2	3	4	5	6	7	8	9	10
5	10	15	20	25	30	35	40	45	50

MID = (10+10) / 2
= 10

4. Here K = DATA [MID] i.e. 50 = 50, So, searching element is found at location MID

Presented By:
Sheikh Muhammad Aamir

Binary Search Algorithm

Study Point

(BINARY SEARCH) This algorithm will find an element K from a list DATA with N elements using Binary Search concept. The list DATA is already in sorted order.

Steps:

1. Set BEG := 1 and END := N
2. Repeat Step 3, and 4 While BEG <= END
3. Set MID := INT((BEG + END) / 2)
4. IF K = DATA[MID] THEN
   Write : 'ITEM FOUND'
   ELSE IF K < DATA[MID] THEN
   Set END := MID - 1
   ELSE IF K > DATA[MID] THEN
   Set BEG := MID + 1
   ELSE
   Write : 'NOT FOUND'
   [End of IF Structure]
   [End of repeat loop]
5. Exit

Presented By:
Sheikh Muhammad Aamir

Complexity of Binary Search

- Let say the iteration in Binary Search terminates after **k** iterations. In the above example, it terminates after 3 iterations, so **here k = 3**
- At each iteration, the array is divided by half. So let's say the length of array at any iteration is **n**
- At **Iteration 1**,

  ```
  Length of array = n
  ```

- At **Iteration 2**,

  ```
  Length of array = n/2
  ```

- At **Iteration 3**,

  ```
  Length of array = (n/2)/2 = n/2^2
  ```

- Therefore, after **Iteration k**,

  ```
  Length of array = n/2^k
  ```

- Also, we know that after

  ```
  After k divisions, the length of array becomes 1
  ```

- Therefore

  ```
  Length of array = n/2^k = 1
  => n = 2^k
  ```

- Applying log function on both sides:

  ```
  => log2 (n) = log2 (2^k)
  => log2 (n) = k log2 (2)
  ```

- As **($\log_a (a) = 1$)**
  Therefore,

  ```
  => k = log2 (n)
  ```

Hence, the time complexity of Binary Search is

$\log_2 (n)$

Study Point

Presented By:
Sheikh Muhammad Aamir

Complexity of Binary Search

Study Point

- The complexity is measured by the number f(n) of comparisons to locate the ITEM in DATA where DATA contains n elements.
- Observe that each comparison reduces the sample size in half.
- So,

$$f(n) = \log_2 (n) = O(\log_2 n)$$

Presented By:
Sheikh Muhammad Aamir

Click to add notes