

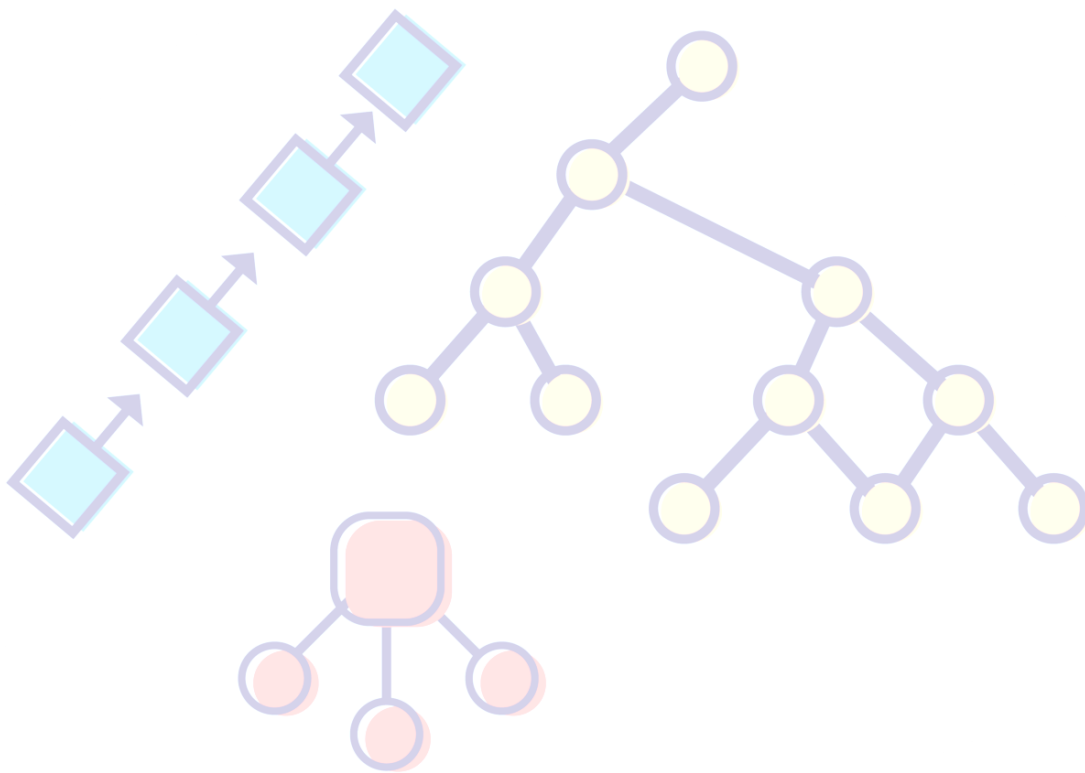
Data Structures and Algorithm

Chapter - 7.1

Binary Search

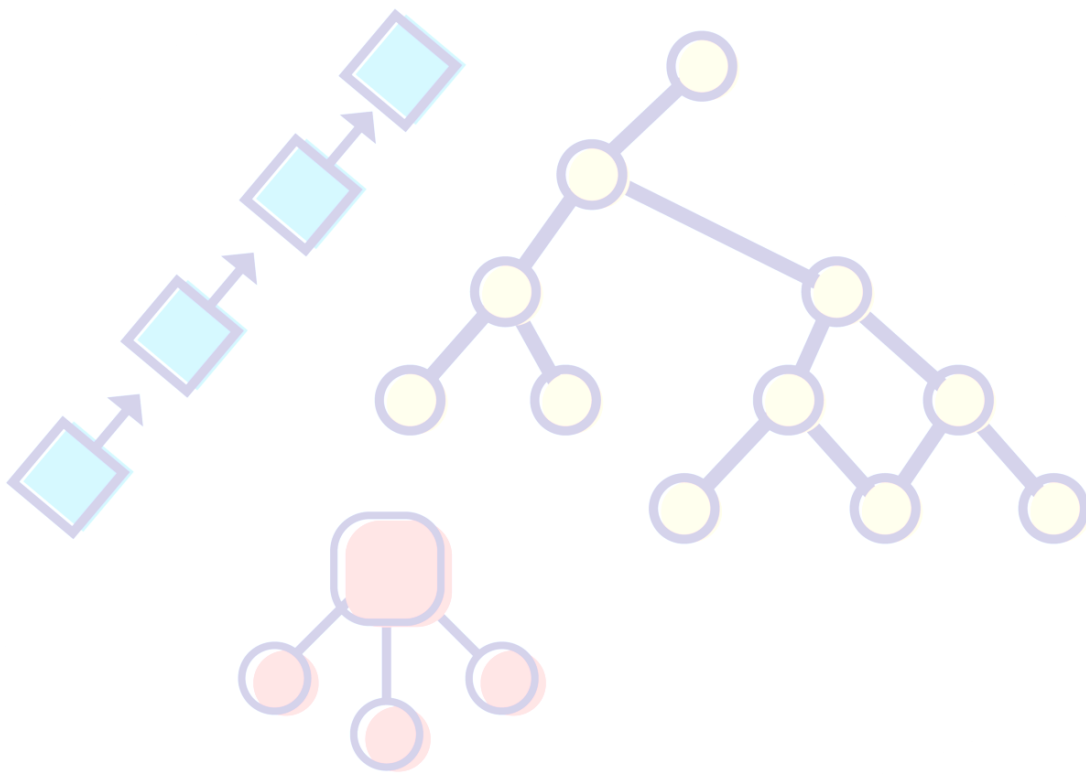
[Rakin Mohammad Sifullah](#)

Computer Science & Engineer



Contents

1. What is Binary Search
2. Algorithm of Binary Search
3. How does Binary Search work?

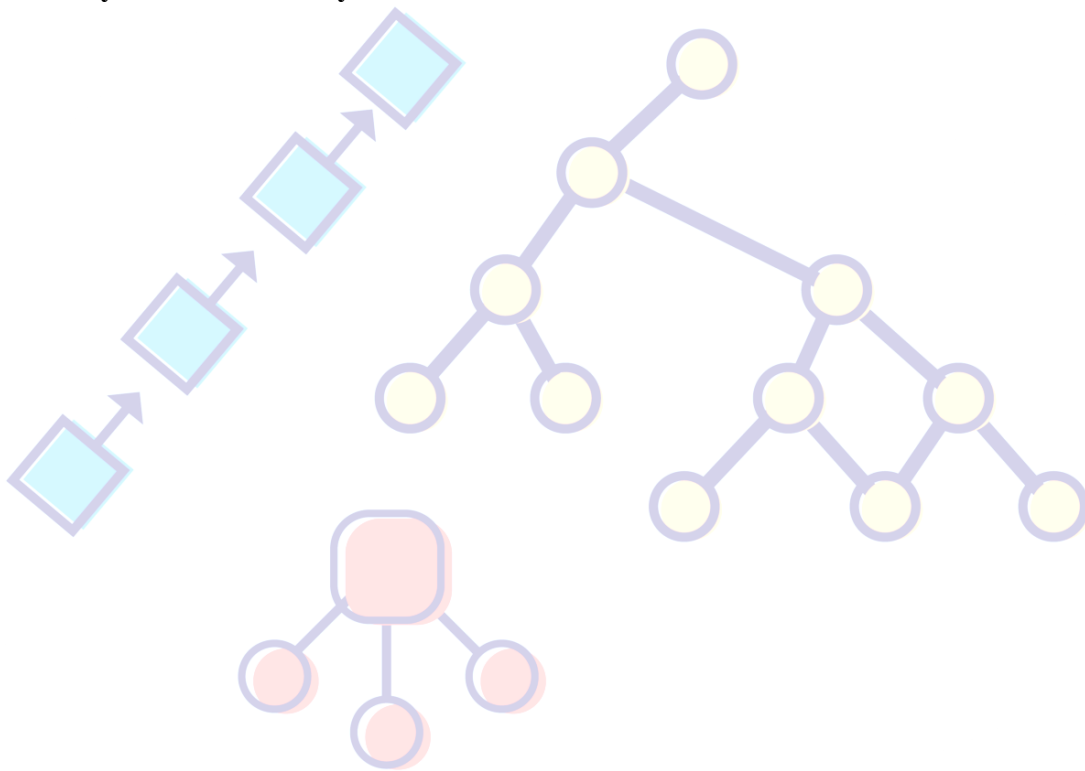


Binary Search

Binary search looks for a particular item by comparing the middlemost item of the collection. If a match occurs, then the index of the item is returned. If the middle item is greater than the item, then the item is searched in the sub-array to the left of the middle item. Otherwise, the item is searched for in the sub-array to the right of the middle item. This process continues on the sub-array as well until the size of the subarray reduces to zero.

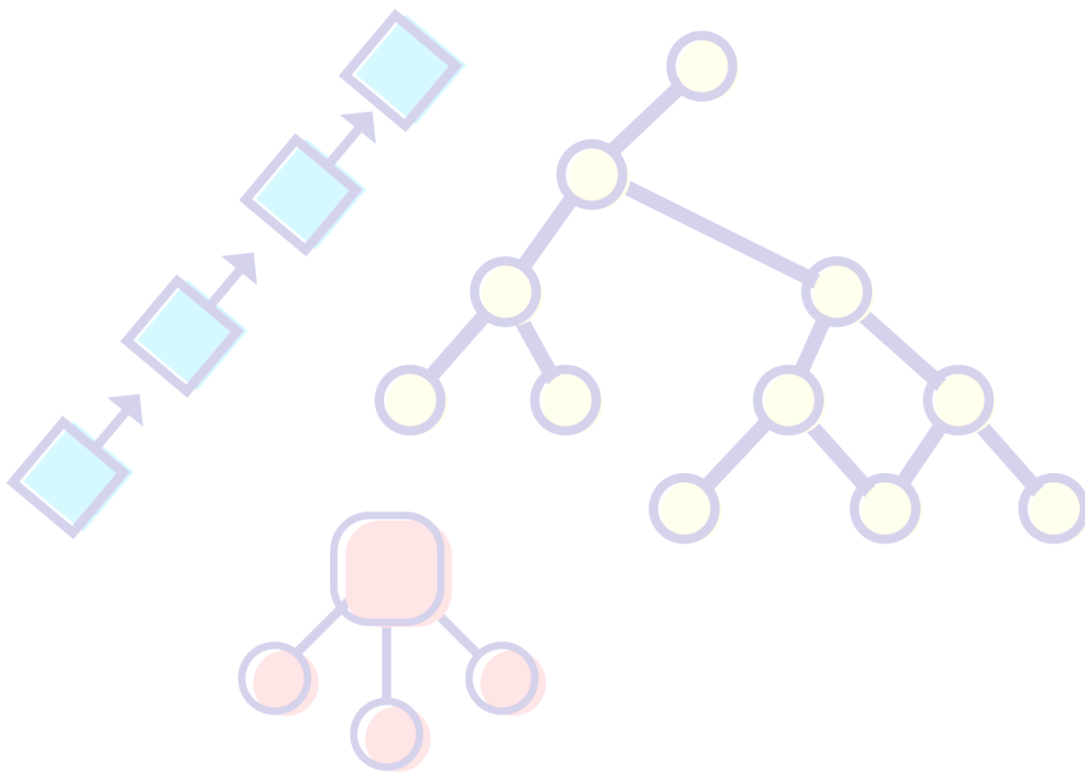
Binary Search is a searching algorithm used in a sorted array by repeatedly dividing the search interval in half.

So, in **Binary Search the array must be sorted**.



Algorithm of Linear Search

1. Find the Middle (M) point of the array and compare it with the expected value.
2. If the expected value is equal to the middle value then return the middle-value index.
3. If the expected value is not matched with the middle value and the expected value is less than the item in the middle of the interval, narrow the interval to the lower half.
4. Otherwise, narrow it to the upper half.
5. Repeatedly check from the second point until the value is found or the interval is empty.



How does Binary Search work?

- Let's assume we have a sorted array = {2,5,8,12,16,23,38,56,72,91} and we have to find 23.
- Here in the array middle point is 16.
- Compare 23 with the middle value 16.
- As $23 > 16$ so pick the upper half of the array.
- Again, find the middle point of the upper half of the array which is 56.
- Compare 23 with the new middle value 56.
- As $23 < 56$ so pick the lower half.
- In the lower half find the middle point which is 23.
- Now compare expected value with middle point and it is a match.

Search 23

2	5	8	12	16	23	38	56	72	91
0	1	2	3	4	5	6	7	8	9

$23 > 16$; pick upper half

2	5	8	12	16	23	38	56	72	91
L=0	1	2	3	M=4	5	6	7	8	H=9

$23 < 56$; pick lower half

					23	38	56	72	91
0	1	2	3	4	L=5	6	M=7	8	H=9

found 23, return 5

					23	38			
0	1	2	3	4	L=5, M=5	H=6	7	8	9
