

2014

May

SATURDAY

24

## Binary Search

Given a sorted array  $[ ]$  of  $n$  elements

Search a sorted array by repeated dividing the search interval in half. Begin with an interval covering the whole array. If the value of the search key is less than item in the middle of the interval, narrow the interval to the lower half otherwise, narrow it to the upper half.

## Binary Search

Search 23

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

25

23 > 16  
take 2nd half

$L=0$   $M=4$   $H=9$

2	5	8	12	16	23	38	56	72	91
---	---	---	----	----	----	----	----	----	----

23 > 56  
take 1st half

$L=5$   $M=7$   $H=9$

2	5	8	12	16	23	38	56	72	91
---	---	---	----	----	----	----	----	----	----

Notes

Round 23,

Return 5

Nobody talks of entrepreneurship as survival, but that's exactly what it is.

July	August	September	October	November	December
S M T W T F S	S M T W T F S	S M T W T F S	S M T W T F S	S M T W T F S	S M T W T F S
1 2 3 4 5 6	1 2 3 4 5 6	1 2 3 4 5 6	1 2 3 4 5 6	1 2 3 4 5 6	1 2 3 4 5 6
7 8 9 10 11 12	7 8 9 10 11 12	7 8 9 10 11 12	7 8 9 10 11 12	7 8 9 10 11 12	7 8 9 10 11 12
13 14 15 16 17 18	13 14 15 16 17 18	13 14 15 16 17 18	13 14 15 16 17 18	13 14 15 16 17 18	13 14 15 16 17 18
19 20 21 22 23 24	19 20 21 22 23 24	19 20 21 22 23 24	19 20 21 22 23 24	19 20 21 22 23 24	19 20 21 22 23 24
25 26 27 28 29 30	25 26 27 28 29 30	25 26 27 28 29 30	25 26 27 28 29 30	25 26 27 28 29 30	25 26 27 28 29 30
31					31



26 MONDAY

2014  
May

Sorting's in Java

1) Arrays.sort()

which can be of primitive data type also

2) Collection.sort()

works for objects collections  
like ArrayList and LinkedList

It is by default - sorts in  
ascending order.

It can be done with the help of  
Collections.reverseOrder().

Collections.reverseOrder().

Notes

Logic will get you from A to B. Imagination will take you everywhere.

January	February	March	April	May	June
S M T W T F S	S M T W T F S	S M T W T F S	S M T W T F S	S M T W T F S	S M T W T F S
1 2 3 4	1	30 31	1 2 3 4 5	1 2 3	1 2 3 4 5 6 7
5 6 7 8 9 10 11	2 3 4 5 6 7 8	2 3 4 5 6 7 8	6 7 8 9 10 11 12	4 5 6 7 8 9 10	8 9 10 11 12 13 14
12 13 14 15 16 17 18	9 10 11 12 13 14 15	9 10 11 12 13 14 15	13 14 15 16 17 18 19	11 12 13 14 15 16 17	15 16 17 18 19 20 21
19 20 21 22 23 24 25	16 17 18 19 20 21 22	16 17 18 19 20 21 22	20 21 22 23 24 25 26	18 19 20 21 22 23 24	22 23 24 25 26 27 28
26 27 28 29 30 31	23 24 25 26 27 28	23 24 25 26 27 28 29	27 28 29 30	25 26 27 28 29 30 31	29 30