

1. 1000
2. 500
3. 250
4. 125
5. 64
6. 32
7. 16
8. 8
9. 4
10. 2
11. ①

the worst case is 10

$$\log_2 1000 = 10$$

$$\log 1000 = 10$$

sort is only $n \cdot \log n$

data set 100
100 checks

1000 ds
1000 ops

data size	$\log N$
1	1
10/16	4
100/128	7
1000/1024	10
1000000	20

conventional	$100 \times 100 = 10000$
binary search	sort = 700
	check $100 \times 7 = 700$
	<u>1400</u>
	6x

<u>1,000,000</u>
10000
10,000
<u>20,000</u>
20K v.

data = 100 elements

request = 100

1) Linear search $\Rightarrow 100 \times 100 = 10,000$

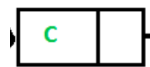
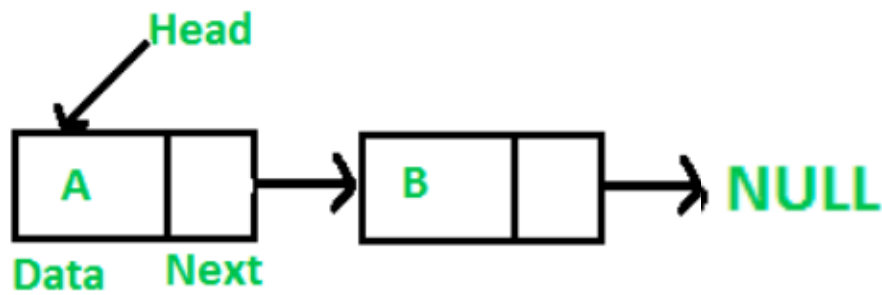
2.1) Merge sort $\Rightarrow 100 \times 7 = 700$

2.2) Binary search $\Rightarrow 7$

2) Binary search $\Rightarrow 700 + 100 \times 7 = 1,400$

1) Linear search $\Rightarrow 100 \times 1,000,000 = 100,000,000$

2) Binary search $\Rightarrow 700 + 1,000,000 \times 7 = 7,000,700$



Head

```
void append(Node nextNode) {
    if (head == null) head = nextNode;
    else {
        Node temp = head;
        while (temp.next != null) temp = temp.next;
        temp.next = nextNode;
    }
}
```

