nsp	nst	
~~3~~ 2	~~1~~ 3	1
~~2~~ 1	~~3~~ 5	2
~~1~~ 0	~~5~~ 7	3
~~0~~ 1	~~7~~ 5	4
~~1~~ 2	~~5~~ 3	5
~~2~~ 3	~~3~~ 1	6
3 4	~~1~~ -1	7

nst += 2, nsp -= 1;

```
      x
    x x x
  x x x x x
x x x x x x x
  x x x x x
    x x x
      x
```

nst -= 2
nsp += 1

n/2 = 3

row = 7 (odd number)
≠ 8 (ceil, 9)

n	nst	nsp
7	1	3
9	1	4
11	1	5
n	1	(n/2)

r
1 → 1
2 → 2 3 2
3 → 3 4 5 4 3
4 → 4 5 6 7 6 5 4
5 → 5 6 7 8 9 8 7 6 5
(6→4) → 4 5 6 7 6 5 4
(7→3) → 3 4 5 4 3
(8→2) → 2 3 2
(9→1) → 1

1 2 3 4 5 6 7 8 9

n	starting
1	1
2	2
3	3
4	4
5	5
6	4 (9-6+1)
7	3 (9-7+1)
8	2 (9-8+1)
9	1 (9-9+1)

val = n

val = (N-n+1)

N = 9

if r ≤ n/2 +1
  val = r;
else
  val = N-r+1

(*)

```
x x x x x x x
  x x x x x
    x x x
      x
    x x x
  x x x x x
x x x x x x x
```

	nst	nsp
1	7	0
2	5	1
3	3	2
4	1	3
5	3	2
6	5	1
7	7	0

→ nst -= 2, nsp += 1

→ nst += 2, nsp -= 1

List

Array

list = [ ]

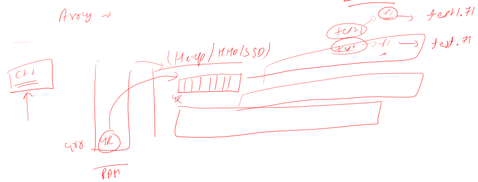
["ab", "cd", 1, 2, 3]

[

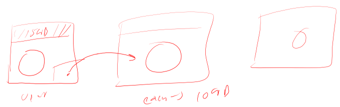
int → 4byte

Linkedlist →   int a(100000)   4,00,000

Array →

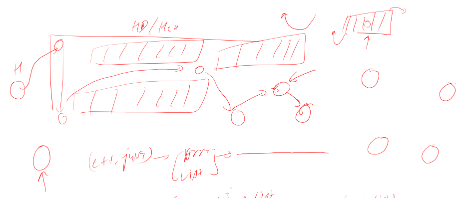


15GB   10000 GB



each → 10GB

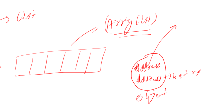
1000 user → 10,000GB



(C++, java) → [Array, List] →

python → [linkedlist] → list

→ numpy Array →



list → (,)

→ linkedlist

10^9 operation → 1s

10^{12} operation → $\frac{1}{10^9} \times 10^{12}$

$= 10^3$ s

= 16 min

-23	-15	0	10	11	13	19	65	79	80	82	93	95	102	111
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

si m ei

80% (How/what) thinking

10% (where)

10% (code)

tc : -23

: 100

: 102

: 11

T: O(??)

mathematical proof

```
si = 0, ei = n
while si < ei
    mid = (si+ei)//2
    if arr[mid] == data:
        return mid
    elif arr[mid] < data:
        si = mid+1
    else
        ei = mid-1
```

```
def binarySearch(arr, data):
    l = len(arr)
    si = 0
    li = l - 1

    while si <= ei:
        mid = (si + ei) // 2
        if mid == data:
            return mid
        elif arr[mid] < data:
            si = mid + 1
        else:
            ei = mid - 1

    return -1
```

$\rightarrow n \quad \frac{n}{2} \quad \frac{n}{4} \quad \frac{n}{8} \quad \frac{n}{16} \; \ldots\ldots \; 8, 4, 2, 1$

$a_0 = n$

$a_k = 1$

$k = ?$

$r = (1/2)$

$a_k = a_0 r^{k-1}$

$1 = n\left(\frac{1}{2}\right)^{k-1}$

$2^{k-1} = n$

$\log_2 2^{k-1} = \log_2(n)$

$(k-1)\cancel{\log_2(2)} = \log_2(n)$

$\boxed{K = \log_2(n) + 1}$

$T: O(K)$

$T: O(\log_2(n))$

$10^{12} \rightarrow 16\text{mil}$

$\log(10^{12}) = 12\log_2(10)$

$= 12\frac{\log_{10}(10)}{\log_{10}(2)}$

$= \frac{12}{0.3}$

$= 40$

$10^9 \rightarrow 1s$

$60 \rightarrow \left(\frac{1}{10^9} \times 60\right) s$

$N = 10^9 \rightarrow$

$N = 10^8 \rightarrow$

$N = 10^7 \rightarrow$

$N = 10^5 \rightarrow$

$N = 10^2$

$\leq O(N)$

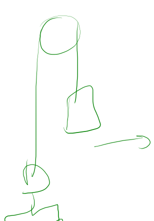
$\leq \log(N)$

$\leq (N^2)$

(N^3)

(N^4)

$10^9 \; 0-1s$



FBD