Time Complexity

Constant time O(1)

1. print statement O(1)
2. if else statement
3. Arithmetic operations O(1)
4. Switch statement.
5. return → constant
6. Break.

Question Pairing

arr → | 1 | 2 | 3 | 4 | 5 | 6 |

0 1 2 3 4 5

i i+1 i+2 i

12 23 34 45 56
13 24 35 46
14 25 36
15 26
16

n = 6

```
for(int i=0; i<arr.length; i++){
    for(int j=i+1; j<arr.length; j++){
        System.out.println(arr[i] + " " + arr[j]);
    }
}
```

(operation)

i	j	
0	5	opn
1	4	opn
2	3	opn
3	2	opn
4	1	opn
5	0	opn

System.out.println(arr[i] + " " + arr[j]); → O(1)

(n-1)

5 + 4 + 3 + 2 + 1 + 0

(n-1) + (n-2) + (n-3) - - - - 1 + 0

Sum of natural no.

$\frac{n(n+1)}{2}$

$\frac{(n-1)(n)}{2} \Rightarrow \frac{1}{2}(n^2 - n) = \frac{n^2}{2} - \frac{n}{2}$

$= O(n^2)$

12 34
13 35
14 36
15
16
23
24
25

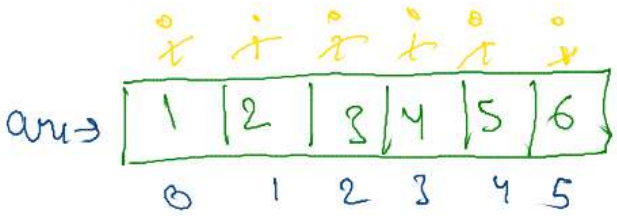
Question

Sum

arr →

1	2	3	4	5	6
0	1	2	3	4	5

i i i i i i i



ans = 21

ans = 0;

for (int i = 0; i < arr.length; i++) {

ans += arr[i] O(1)

Syso (ans); O(1)

4

1 + 1 + 1 + 1 + 1 + 1

⇓

6

⇓

O(n)

ans = ~~0~~ + ~~3~~

~~6~~ ~~10~~

~~15~~

21

Qu

i = 1 ; i < n ; i = i*2) {

Syso(i);

n = 32

TC = ∞ when i = 0

}

1+1+1+1+1

i = 1*2
⇓
2*2
⇓
4*2 = 8*2
= 16*2
= 32

# of op^n

	n
1	⇓
2	5
4	
8	32
16	⇓
	5

5

n = 32

$= 2^5 \Rightarrow$

$\log_2(n) = \log_2(2^5)$

$\log n = 5$

O(log n)

↑ log n

32	5

123 - - - 32

O(n)

for (int i=1 ; i < n ; i++) {

if (i*2 = pow(2,i)) {

Syso(i)

32

}

}

i*3

⇓

$\log_3 n$

$\log_2(n)$

$\log n < n$

$\log 32 < 32$

↓

$\boxed{5 < 32}$

$\log a^b$

multiply both side by 3

$3 \times a = b * 3$

$32 = 2^5$ multiply $\log_2$ both side

$\log_2(32) = \log_2(2^5)$

$\log_2(32) = 5(\log_2 2)$

$\log(32) = 5*1$

↓

$\Rightarrow \boxed{\log n}$

linear Search = O(n) n=15

```
for(i=0 ; i< n ; i++){
    i = i*2 ;
}
```

i = ~~0~~ ~~2~~ ~~0~~ 1*2 ~~3~~ ~~6~~ ~~7~~ ~~14~~ 15

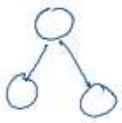
1 + 1 + 1 + 1

= TC ⇒ log(n) (+1) → α

↳ O(log n) ?

0 1 3 7 14

*2 *2 *2 *2



```
for(i=1 ; i<n ; i=i*2){
    sysout
}
```

1
2
4
8
16
32
64

= 6

$64 = 2^6$

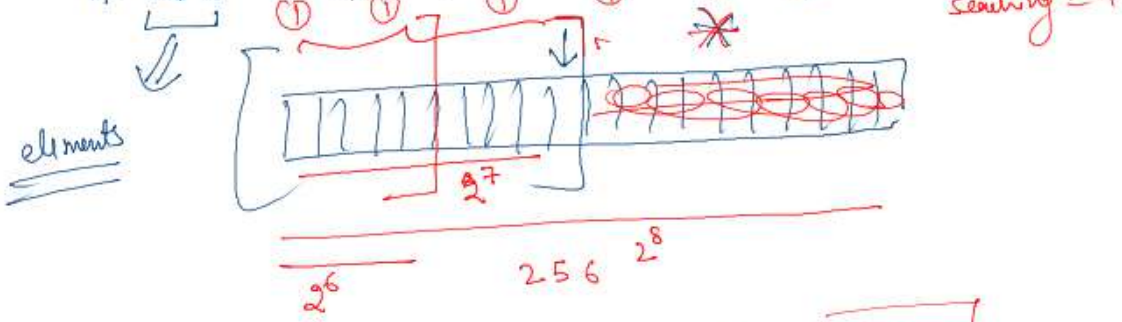
$\log(64) = \log 2^6$

$\log(n) = 6$

↳

$n = 256 = 2^8 \Rightarrow 2^7 \Rightarrow 2^6 \Rightarrow 2^5 \Rightarrow 2^4 \Rightarrow 2^3 - 2^2 - 2^1 \rightarrow 2^0$

searching = 9



$2^8 \rightarrow 2^7 \rightarrow 2^6 - - - - 2^0$ $\quad \log_2 n$

$256 \simeq 9$