Time complexity 2

```
n=4
for(int i=0; i<=n; i++){
    syso(i); ✓ O(1)
}
```

n+1

O(n)

i=0	O(1)
i=1	1 O(1)
i=2	2 O(1)
:	
i=n	n O(1)

O(1) + O(1) + O(1) ---- n time

n * O(1) = O(n)

(n→0) (n+1)

n = 5

②
```
int count=0;
for(int i=n; i>=0; i--){
    count++; → O(1)
}
syso(count); → O(1)
```

n → 1 → n

n → 0 → n+1

→ O(n)

n	
5 →	1
4 →	2
3 →	3
2 →	4
1 →	5
0 →	6

5<=0

n=5

③
```
for(int i=n; i<=0; i--){
    syso(i);
}
```
O(1) ✓

④
```
for(int i=0; i<n; i+=2){
    syso(i) → O(1)
}
```
n=8

0	0
2	2
4	4
6	6
8	

O(n/2)

1/2 * O(n)

O(n)

(5) `for(int i=0; i<n; i+=3){` → n/3
`Syso(i);` = O(n)
}

n = 4

(6)
```
for (int i = 0; i<=n ; i++){
    for (int j=0; j<=n; j++){
        Syso( j);
    }
}
```

i = 0	0,1,2,3,4 (n)
i = 1	0,1,2,3,4 (n)
i = 2	0,1,2,3,4 (n)
i = 3	0,1,2,3,4 (n)
i = 4	0,1,2,3,4 (n)

n + n + n + n - - - - n times.

n(1 + 1 + 1 + 1 - - - n time);

n(1 * n) = n^2

(7)
```
for (int i = 0; i<=n; i++){
    for( int j = i ; j<=n; j++){
        Syso(j);
    }
}
```

i	n = 4
0	(0,1,2,3,4) n
1	(1 2 3 4) n-1
2	(2 3 4) n-2
3	3 4 : 2
4	4 : 1

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

$\frac{n*(n+1)}{2} = \frac{1}{2}(n^2+n) = n^2+n = O(n^2)$

(8) for (int i=0; i<n; i++) {
    for (int j=0; j<=i; j++)
        Syso(j);
}

$O(n^2)$

i	
0	(0) 1
1	(0,1) 2
2	(0,1,2) 3
3	(0,1,2,3) 4

n (0 1 ---- n) n

$1+2+3+4+\cdots+n = n(n+1)/2 = O(n^2)$

(9) for (int i=1; i*i<=n; i++) {
    Syso(i);
}

$\sqrt{n}=1$

n = 100 → 32 → 5 → 6

i	
1	1
2	2
3	3
4	4
5	5

$\sqrt{16} \to 4$

$\sqrt{32} \to 5$

$\sqrt{36} \to 6$

$O(\sqrt{n})$

int q = (int) Math.sqrt(n);

(10) for (int i = 1; i <= q; i++) {

System.out.println(i);

}

n = √16 = 4

i
1
2
3
4
5

√n

O(√n)

O(n)
O(n²)
O(√n)

(int)

for (int i = 1; i <= Math.sqrt(n); i++) {
System.out.println(i);
}

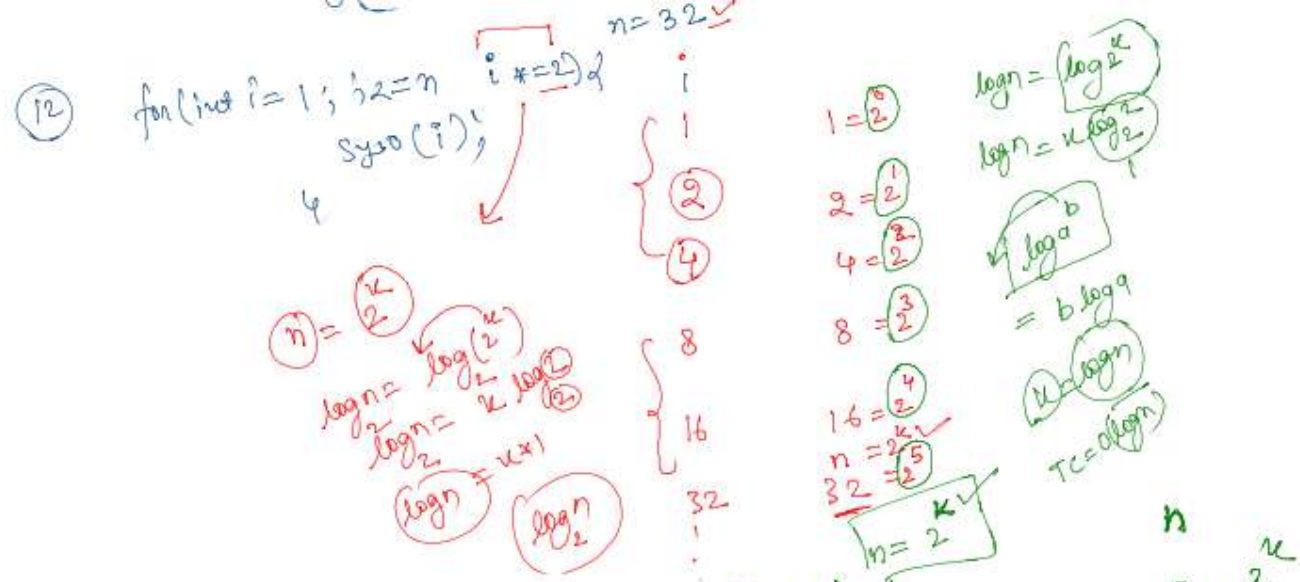
(11)

```
for (int i = 0; i <= n; i++) {
    for (int j = 1; j <= √n; j++) {
        System.out.println(i);
    }
}
```

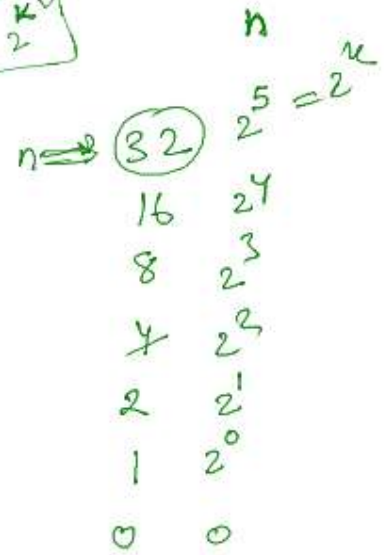
n × √n

O(n * √n)

(12) `for(int i=1; i<=n; i*=2){ sysо(i); }`



(13) `for (int i=n; i>=1; i/=2){ sysо(i); }`



$n = 2^k$

$k = O(\log n)$

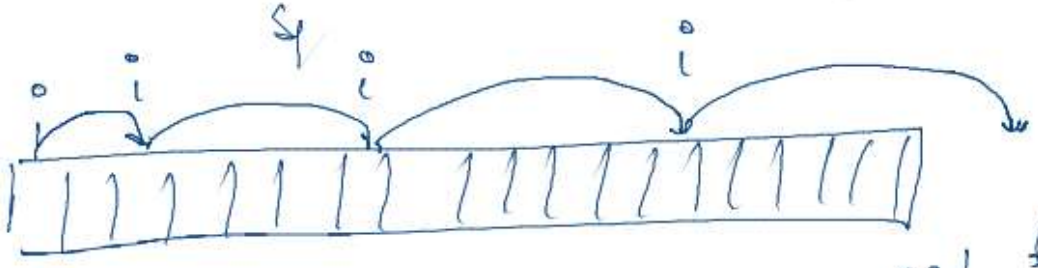
$= O(\log n)$

(14) `for(int i=0; i<=n; i++){ for(int j=1; j<=n; j*=2){ sysо(j); } }`

$O(n)$ × $(\log n)$

$= n \log n$

(15) `for (int i=0; i< arr.length; i*=2) {`
`sop (arr[i]);`

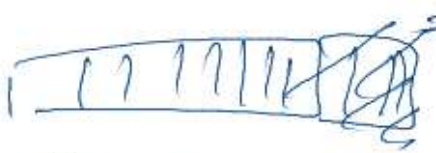


$\log(n)$

$= \log_2 n$

`int n = arr.length;`

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



n
$n/2$
$n/4$
$n/8 \rightarrow n/16 \ldots$ (1)

$32/2$
$16/2$
$8/2$
$4/2$
$2/2$
1

$(\log_2 n)$

(17) `n=str.length()`

```
for (int i=0; i< str.length(); i*=2) {
    sop (str.charAt(i));
}
```

$= O(n)$

$= O(n)$

$\log n$

0
2
4
6
8
10

education
0 1 2 3 4 5 6 7 8