Time Complexity

1. for (i=0, i<n, i++)
   { stmt — n
   } → O(n)

2. for (i=n, i>0; i--)
   { stmt } — n → O(n)

3. for (i=1; i<n; i=i+2)
   { stmt — n/2
   } f(n)=n/2 → O(n)

4. for (i=0, i<n, i++) — n+1
   { for (j=0, j<n, j++) — n(n+1)
   { stmt — n×n
   } } → O(n²)

5. for (i=0; i<n, i++)
   { for (j=0, j<i, j++)
   { stmt
   } } → O(n²)

6. p=0
   for (i=1, p<=n, i++)
   { p = p+i
   }

   p>n

   ∵ $p = \frac{K(K+1)}{2} > n$

   $K^2 > n$

   $K > \sqrt{n} \rightarrow O(\sqrt{n})$

7. for(i=1, i<n, i=i*2)
{ stmt
}
→ O(log₂ n)

Assume, i>n
∴ i = 2^K
∴ K = log₂ n

1×2 = 2
2×2 = 4
2²×2 = 2³
⋮
2^K

8. for(i=n, i>=1, i=i/2)
{ stmt
}
→ O(log₂ n)

Assume i<1
then i/n
∴ $\frac{n}{2^K} \leq 1$
⇒ K = log₂ n

9. for(i=0, i*i<n, i++)
{ stmt
}
→ O(√n)

i*i < n
i*i >= n
i² = n
i = √n

10. for(i=0, i<n, i++)
{ stmt ——— n
}
for(j=0, j<n, j++)
{ stmt
} ——— $\frac{n}{2n}$
→ O(n)

11. P=0
for(i=1, i<n, i=i*2)
{ P++ } P = log n
for(j=1, j<P, j=j*2)
{ stmt — log P
}
O(log(log n))

12. for (i=0, i<n, i++) —— n

{ for (j=1, j<n, j=j*2)

{ stmt —— n x logn

} O(n logn)

Summary

for (i=0, i<n, i++) — O(n)

for (i=0, i<n, i=i+2) — n/2 - O(n)

for (i=n, i>1, i--) — O(n)

for (i=1, i<n, i=i*2) — O($\log_2 n$)

for (i=1; i<n, i=i*3) — O($\log_3 n$)

for (i=n, i>1, i=i/2) — O($\log_2 n$)

13. i=0

while(i<n) — n+1

{ stmt — n

i++ — n

} f(n) = 3n+2

→ Θ(n)

for (i=0, i<n, i++)

{ stmt

}

→ O(n)

14. a=1

while(a<b)

{ stmt;

a=a*2 }

→ O(logn)

a
1 x 2 = 2
2 x 2 = 2^2
2^2 x 2 = 2^3
⋮
2^k

$a > b$

$\therefore a = 2^k$

$2^k \geq b$

$\Rightarrow k = \log_2 b$

for (a=1, a<b, a=a*2)

{ stmt;

~~i=i/2~~ }

O(logn)

15.
```
i = n;
while (i > 1):
{ stmt
    i = i/2 }
```
→ O(log n)

16.
```
i = 1
K = 1
While (K < n)
{ stmt
    K = K + i
    i++ }
```
→ $O(\sqrt{n})$

i	K
1	1
2	1+1 = 2
3	2+2 = 4
4	2+2+3 =
⋮	
m	1+2+3+4 $\frac{m(m+1)}{2}$

$K \geq n$

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

$m \geq \sqrt{n}$

```
for (K=1, i=1, K<n, i++)
{ stmt
    K = K+i }
```

17.
```
While (m != n)
{ if (m > n) :
    m = m - n
  else: n = n - m }
```
min O(1) , O(n)

m = 16	n = 2
14	2
12	2
10	2
⋮	
2	2

18.
```
Algorithm test (n)
{ if (n < 5)
    printf ("%d", n) }—— 1
  else: { for (i=0, i<n, i++)
            { printf ("%d", i) — n
            }}
```
best O(1)

Worst O(n)