```
int fun(int n)
{
    if(n==0)
        return 0;
    return n + fun(n-1);
}
```

$\rightarrow O(1)$

$$\Rightarrow \quad fun(n) = \begin{cases} 0, & n = 0 \leftarrow \\ n + fun(n-1), & \text{otherwise} \end{cases}$$

$$f(n) = n + f(n-1)$$

$$\Rightarrow \boxed{T(n) = n + T(n-1)} \leftarrow \text{Recurrence Relation}$$

```
void fun(n)
{
  if (n>0)
  {
      fun(n-1);
      s.o.p(n);
  }
}
```

$$fun(n) = \begin{cases} \phi, & n=0 \\ fun(n-1) + s.o.p(n) \end{cases}$$

$$\boxed{T(n) = T(n-1) + 1}$$

← Recurrence Relation.

```
int fact(n)
{
    if(n==0)
        return 1;
    return n* fact(n-1);
}
```

$$fact(n) = \begin{cases} 1 & , n = 0 \\ n * fact(n-1) & , n > 0 \end{cases}$$

$$f(n) = n * f(n-1)$$

$$\boxed{T(n) = n * T(n-1)}$$

(1) $T(n) = T(n-1) + 1 \rightarrow$

(2) $T(n) = n + T(n-1) \rightarrow$

(3) $T(n) = n * T(n-1) \rightarrow$

$$T(n) = T(n-1) + 1$$

$$\boxed{T(n-1) = T(n-2) + 1}$$

$$T(n) = [T(n-2) + 1] + 1$$

$$T(n) = T(n-2) + 2$$

$$= [T(n-3) + 1] + 2$$

$$T(n) = T(n-3) + 3$$

$$\vdots$$

$$\boxed{T(n) = T(n-k) + k}$$

$T(0) \Rightarrow 1$

At least step $\Rightarrow$ $T(n-k) = T(0)$

$\Rightarrow n-k = 0 \Rightarrow \boxed{k = n}$

$\Rightarrow T(n) = T(n-n) + n$

$$\boxed{T(n) = O(n)}$$

$$T(n) = T(n-1) + n$$

$$= \left[T(n-2) + (n-1)\right] + n$$

$$= T(n-2) + (n-1) + n$$

$$= \left[T(n-3) + (n-2)\right] + (n-1) + n$$

$$T(n) = T(n-3) + (n-2) + (n-1) + n$$

$$\vdots$$

$$= T(n-k) + (n-k+1) + \dots + (n-2) + (n-1) + n$$

$T(n-k) = T(0)$

$n-k=0$

$k=n$

$$= T(n-n) + (n-n+1) + \dots + (n-2) + (n-1) + n$$

$$= 1 + \dots + (n-2) + (n-1) + n$$

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

$$\Rightarrow \boxed{T(n) = O(n^2)}$$

$$T(n) = n * T(n-1)$$

$$\Rightarrow T(n) = n * [(n-1) * T(n-2)]$$

$$= n * (n-1) * T(n-2)$$

$$= n * (n-1) * [(n-2) * T(n-3)]$$

$$= n * (n-1) * (n-2) * T(n-3)$$

$$\vdots$$

$$= n * (n-1) * (n-2) * \dots * (n-k+1) * T(n-k)$$

$$= n * (n-1) * (n-2) * \dots * (n-n+1) * \cancel{T(n-n)}$$

$$= n * (n-1) * (n-2) * \dots * 1 \Rightarrow n! \Rightarrow \boxed{T(n) = O(n!)}$$

$$\boxed{T(n) = T(n/2) + 1}$$

$$= [T(n/4) + 1] + 1$$

$$= T(n/4) + 2$$

$$= [T(n/8) + 1] + 2$$

$$= T(n/8) + 3$$

$$\vdots$$

$$\boxed{T(n) = T(n/2^k) + k}$$

At last step $\Rightarrow T(n/2^k) = T(1)$

$n/2^k = 1 \Rightarrow 2^k = n$

$\Rightarrow \boxed{k = \log(n)}$

$T(n) = T(\cancel{n/n}) + \log(n)$

$$\boxed{T(n) = \log(n)}$$

$$T(n) = 2T(n/2) + n$$

$$\Rightarrow \boxed{T(n/2) = 2T(n/4) + n/2}$$

$$= 2\left[2T(n/4) + \frac{n}{2}\right] + n$$

$$= 4T(n/4) + \frac{n}{2} + n$$

$$= 4\left[2T(n/8) + n/4\right] + \frac{n}{2} + n$$

$$= 8T(n/8) + \frac{n}{4} + \frac{n}{2} + n$$

$$\vdots$$

$$= 2^k T(n/2^k) + \frac{n}{2^{k-1}} + \cdots + \frac{n}{4} + \frac{n}{2} + n$$

$$\frac{n}{2^k} = 1$$

$$\Rightarrow 2^k = n \Rightarrow k = \log n$$

$$T(n) = 2^{\log n} T(n/n) + \frac{n}{2^{\log n - 1}} + \cdots \frac{n}{4} + \frac{n}{2} + n$$

$$= n + n\left[\frac{1}{n} + \cdots + \frac{1}{8} + \frac{1}{4} + \frac{1}{2} + 1\right]$$

$$= n + n\left[1 + \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \cdots + \frac{1}{n}\right]$$

$$a\left(\frac{1-r^n}{1-r}\right) \Rightarrow \left[\frac{1-\left(\frac{1}{2}\right)^n}{1-\frac{1}{2}}\right]$$

$$T(n) = 2\,T(n/2) + n \leftarrow \text{Quick Sort}$$

$$= 2\left[2\,T(n/4) + n/2\right] + n$$

$$= 4\,T(n/4) + n + n$$

$$= 4\left[2 * T(n/8) + \frac{n}{4}\right] + n + n$$

$$= 8 * T(n/8) + n + n + n$$

$$\vdots$$

$$= 2^k\,T(n/2^k) + n + \cdots + n + n$$

$$= n + n + \cdots + n + n = O(n^2)$$

$$\frac{n}{2^k} = 1$$

$$1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} \cdots \frac{1}{n}$$

$$\sum_{k=1}^{n} \frac{1}{k} = \log n$$

$$\boxed{fib(n) = fib(n-1) + fib(n-2)}$$

fib(5) ← 1

fib(4) + fib(3) ← 2

fib(3) + fib(2)   fib(2) + fib(1) ← 4

fib(2) + fib(1)   fib(1) + fib(0)   fib(1) + fib(0) ← 8

2^k ← 2^n ←

$$T(n) = T(n/2) + n$$

$$= T(n/4) + \frac{n}{2} + n$$

$$= T(n/8) + \frac{n}{4} + \frac{n}{2} + n$$

$$\vdots$$

$$= \cancel{T(n/2^k)} + \cancel{\frac{n}{2^{k-1}}} + \dots + \frac{n}{4} + \frac{n}{2} + n$$

$$\frac{1}{2} \quad \frac{1}{4}$$

$$n\left(\frac{1}{2^{k-1}} + \dots + \frac{1}{4} + \frac{1}{2} + 1\right) \Rightarrow n\left(\frac{1}{2^{\log n - 1}} + \dots \frac{1}{4} + \frac{1}{2} + 1\right)$$

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

$$= 2^{\log n} - 1$$

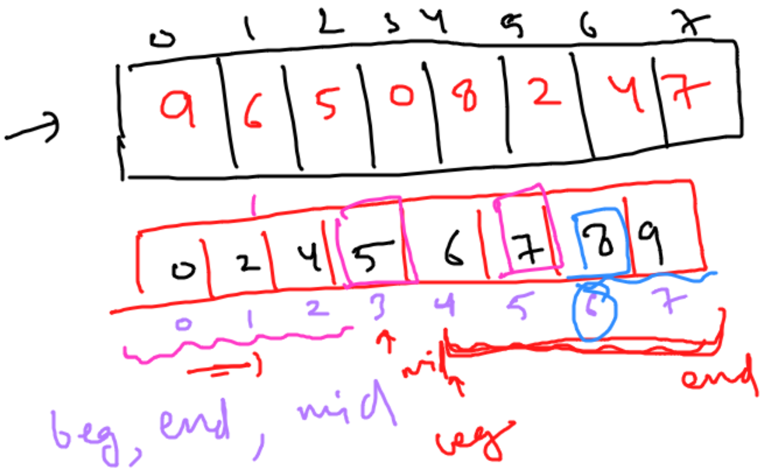
$$n$$

$$\Rightarrow n^2$$

10

~~8~~

x



beg, end, mid

2

x

beg = 0
end = mid-1

beg = ~~0~~ ~~4~~ ~~6~~ ~~7~~ 8

end = n-1 7

$\frac{0+7}{2} = 3$

$\frac{4+7}{2} = 5$

$\frac{6+7}{2} = 6$

beg = 0
end = 2

```
while ( beg <= end)
{
    mid = (beg + end) / 2
    if( x == arr[mid])
        return mid;
    else if( x > arr[mid])
        beg = mid+1;
    else
        end = mid-1;
}
return -1;
```