Algorithm FactorialRec (int n) | #Primops

if $(n \le 1)$ return n — or 3 / 2

else

return $n *$ FactorialRec$(n-1)$ — $3 + T(n-1)$

$$\underbrace{\text{return}}_{1} \quad \underbrace{n *}_{1} \text{FactorialRec}\underbrace{(n-1)}_{1}$$

$T(n) =$ # of prim op. performed by FactorialRec(n)

$$T(n) = \begin{cases} 3 & \text{if } n \le 1 \\ 5 + T(n-1) & \text{if } n > 1 \end{cases}$$ Recursive formula for $T(n)$

n	0	1	2	3	4	5
$T(n)$	3	3	5+3=8	5+8=13	18	23 . . . . .
explicit	3	3	8	13	18	23

Goal: Find an explicit formula for $T(n)$

→ no $T(\cdot)$ on RHS $\quad T(n) = 3n+2$

$T(n) = n\log(n) + 17$

~~Substitut~~ Substitution approach

$$\begin{aligned}
T(n) &= 5 + T(n-1) && (1)\\
&= 5 + (5 + T((n-1)-1)) = 10 + T(n-2) && (2)\\
&= 10 + (5 + T(n-2-1)) = 15 + T(n-3) && (3)\\
&= 15 + (5 + T(n-3-1)) = 20 + T(n-4) && (4)\\
&\ \ \vdots \\
&= 5 \cdot k + T(n-k) && (k)
\end{aligned}$$

Recursion stops when $n-k=1 \equiv k=n-1$

$T(n) = 5 \cdot (n-1) + T(1) = 5n - 5 + 3 = 5n - 2$

$\Rightarrow$ Explicit formula: $T(n) = \begin{cases} 5n-2 & \text{if } n \ge 1 \\ 3 & \text{if } n = 0 \end{cases}$

Check $\Rightarrow T(n)$ is $O(n)$

Running time of recursive algos.

```
BinarySearch(A, start, stop, key)
if(start==stop) {
    if(A[start]=key) return start
    else return -1
else
    mid = ⌊(start+stop)/2⌋
    if(A[mid] ≤ key) r = BinarySearch(A, start, mid, key)
    else r = BinarySearch(A, mid+1, stop)
return r
```

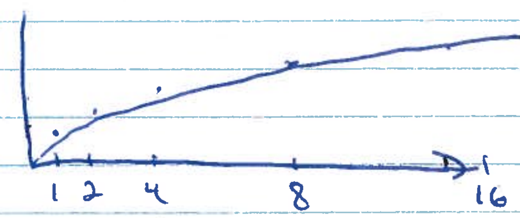
BinarySearch (recursive)

$T(n) =$ # of prim ops on array of size n

$$T(n) = \begin{cases} \underline{T\left(\frac{n}{2}\right)} + \cancel{7}\ 11 & \text{if } n>1 \\ \cancel{2}\ 6 & \text{if } n=1 \quad \text{Base case} \end{cases} \quad \text{recurrence}$$

Goal: Find explicit formula for $T(n)$: $3n^2+2n+5$, $n \cdot \log(n) + 7$ } No $T(\cdot)$ on RHS

n	1	2	4	8	16
$T(n)$	2	$T(1)+7 = 9$	$T(2)+7 = 16$	$T(4)+7$ 23	30



Substitution method

$$T(n) = T(n/2) + 7 \quad (1)$$

$$= \left[T\left(\frac{n/2}{2}\right) + 7\right] + 7 = T\left(\frac{n}{4}\right) + 14 \quad (2)$$

$$= \left[T\left(\frac{n/4}{2}\right) + 7\right] + 14 = T\left(\frac{n}{8}\right) + 21 \quad (3)$$

$$= \left[T\left(\frac{n/8}{2}\right) + 7\right] + 21 = T\left(\frac{n}{16}\right) + 28 \quad (4)$$

$$\vdots$$

$$= T\left(\frac{n}{2^k}\right) + 7k \quad (k)$$

We hit base case of recurrence when $\frac{n}{2^k} = 1 \Leftrightarrow n = 2^k \Leftrightarrow k = \log_2 n$

$$= T(1) + 7 \cdot \log_2 n$$

$$\boxed{T(n) = 2 + 7 \cdot \log_2 n}$$

! explicit formula

n	1	2	4	8	16
$T(n)$	2	9	16	23	30

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

Algorithm MergeSort(A, start, stop)

if(start = stop) return	3 or 2
else	
mid $\leftarrow \lfloor (start+stop)/2 \rfloor$	4
MergeSort(A, start, mid)	$1+T(n/2)$
MergeSort(A, mid+1, stop)	$1+T(n/2)+1$
Merge(A, start, mid, stop)	$1+9n+3$ (say)

Let $n = \cancel{start}$ stop - start+1 = # elements to be sorted

$T(n)$ = # of prim ops performed by MergeSort when sorting n elements

$$T(n) = \begin{cases} 3 & \text{if } n=1 \\ 2+4+T(n/2)+T(n/2)+1+9n+3 = 2T(n/2)+9n+13 & \text{if } n>1 \end{cases}$$

$$T(n) = \begin{cases} 3 & \text{if } n=1 \\ 2T(n/2)+9n+13 & \text{if } n>1 \end{cases}$$

Assume $$T(n) = \begin{cases} 3 & \text{if } n=1 \\ 2T(n/2)+n+1 & \text{if } n>1 \end{cases}$$

n	1	2	~~3~~	4	5
$T(n)$	3	$2T(2/2)+2+1 = 2\cdot 3+2+1 = 9$		$2\cdot T(4/2)+4+1 = 2\cdot T(2)+4+1 = 23$	

Goal: Get an explicit formula for $T(n)$
Use subst. method

MergeSort

MergeSort(...)
if (start = stop) return
else mid = [(start+stop)/2]
MergeSort(A, start, mid)
MS (A, mid, stop)
merge

4
$T(n/2)$
$T(n/2)+1$
$9n+3$ (say)

$T(n) = \begin{cases} 3 & \ldots \\ 10 + 9n + 2\ldots \end{cases}$

$$T(n) = \begin{cases} 2\cdot T\left(\frac{n}{2}\right) + n + 1 & \text{if } n > 1 \\ 1 & \text{if } n = 1 \end{cases}$$

n	1	2	4	8	16
$T(n)$	1	$2\cdot T(1) + 2 + 1 = 5$	$2\cdot T(2) + 4 + 1 = 15$	.	.

Subst. method

$$T(n) = 2\,\underbrace{T\left(\tfrac{n}{2}\right)} + n + 1 \quad (1)$$

$$= 2\left[2\cdot T\left(\tfrac{n/2}{2}\right) + \tfrac{n}{2} + 1\right] + n + 1 = 4\,\underbrace{T\left(\tfrac{n}{4}\right)} + 2n + 2 + 1 \quad (2)$$

$$= 4\left[2\cdot T\left(\tfrac{n/4}{2}\right) + \tfrac{n}{4} + 1\right] + 2n + 2 + 1 = 8\,\underbrace{T\left(\tfrac{n}{8}\right)} + 3n + 4 + 2 + 1 \quad (3)$$

$$= 8\left[2\cdot T\left(\tfrac{n/8}{2}\right) + \tfrac{n}{8} + 1\right] + 3n + 4 + 2 + 1 = 16\cdot T\left(\tfrac{n}{16}\right) + 4n + 8 + 4 + 2 + 1 \quad (4)$$

$$\vdots$$

$$= 2^k\cdot T\left(\frac{n}{2^k}\right) + k\cdot n + \sum_{i=0}^{k-1} 2^i \quad (k)$$

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

$$\boxed{\sum_{i=0}^{n} a^i = \frac{a^{n+1} - 1}{a - 1}}$$

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

We hit base case when $\frac{n}{2^k} = 1 \Leftrightarrow 2^k = n \Leftrightarrow k = \log_2 n$

$$= n\cdot T(1) + (\log_2 n)\cdot n + (n-1)$$

$$= n + (\log_2 n)n + n - 1$$

$$= \boxed{n\cdot\log_2 n + 2n - 1} \Leftarrow \text{explicit formula}$$

Check $n = 4$ $\quad T(4) = 4\cdot\log_2 4 + 2\cdot 4 - 1 = 4\cdot 2 + 2\cdot 4 - 1 = 15$

$\Rightarrow T(n)$ is $O(n\log n)$