IT 202 Assignment-1

Name: Suyash Chintawar
Roll no: 191IT109

Page No.:

Q. Analysing time complexity for Pascal Triangle

⇒ 1] Time complexity for pascEle (r, c) function :-

Here, $T(n,m)$ is given by,

$$T(n,m) = T(n-1,m) + T(n-1,m-1) + O(1)$$

where $r = n$ (row), $c = m$ (column)

Assuming $T(n-1,m) = T(n-1,m-1)$ — (1) (given).

$\therefore T(n,m) = 2\,T(n-1,m) + O(1)$ — (2)

$T(n-1,m) = 2\,T(n-2,m) + O(1)$

$T(n-2,m) = 2\,T(n-3,m) + O(1)$

⋮ ⋮ ⋮

$T(k,m) = 2\,T(k-1,m) + O(1)$

Using back substitution in eqⁿ (2),

$$\begin{aligned} T(n,m) &= 2\,T(n-1,m) + O(1) \\ &= 2(2\,T(n-2,m) + O(1)) + O(1) \\ &= 4(2\,T(n-3,m) + O(1)) + 3\,O(1) \\ &= 8(2\,T(n-4,m) + O(1)) + 7\,O(1) \\ &= 16(2\,T(n-5,m) + O(1)) + 15\,O(1) \end{aligned}$$

Hence, we see that,

$$T(n,m) = 2^k\,T(n-k,m) + (2^k - 1)\,O(1)$$

if $n-k=0 \Rightarrow n=k$,

$\therefore T(n,m) = 2^n\,T(0,m) + (2^n-1)\,O(1)$

$$\therefore \boxed{T(n,m) = 2^n + (2^n-1)\,O(1)}$$

Hence, time complexity for recursive function is $\boxed{O(2^n)}$.

As time complexity for pascEle(r,c) is $O(2^n)$ we can assume its cost to be $c_6 = 2^n$. ——— (3)

2] Time complexity of main function :-

	cost	times
input n	c_1	1
for i ← 0 to n-1 do	c_2	n+1
for j ← 0 to n-i-1 do	c_3	$\sum_{t=0}^{n-1} t+1$
print " "	c_4	$\sum_{t=0}^{n-1} t$
for j ← 0 to i do	c_5	$\sum_{t=0}^{n-1} t+1$
print pascEle(i,j)	c_6	$\sum_{t=0}^{n-1} t$
print "\n"	c_7	n.

so,

$$\text{execution time}(T(n)) = c_1 + c_2(n+1) + c_3\left(\sum_{t=0}^{n-1} t+1\right) + c_4\left(\sum_{t=0}^{n-1} t\right) + c_5\left(\sum_{t=0}^{n-1} t+1\right) + c_6\left(\sum_{t=0}^{n-1} t\right) + c_7 n$$

$$\therefore T(n) = c_1 + c_2(n+1) + c_3\left(\frac{n(n-1)}{2} + n\right) + c_4\left(\frac{n(n-1)}{2}\right) + c_5\left(\frac{n(n-1)}{2} + n\right) + c_6\left(\frac{n(n-1)}{2}\right) + c_7 n.$$

From eqn (3), substituting $c_6 = 2^n$,

$$T(n) = c_1 + c_2(n+1) + c_3\left(\frac{n(n-1)}{2} + n\right) + c_4\left(\frac{n(n-1)}{2}\right) + c_5\left(\frac{n(n-1)}{2} + n\right) + 2^n\left(\frac{n(n-1)}{2}\right) + c_7 n.$$

Highest order in the above eqn is $2^n \cdot n^2$.

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

Hence, for main function (whole program), big-O is $O(2^n \cdot n^2)$