Name: Rahul Goel
Reg No: RA1911030010094
Batch: CSE - 02

①

Algorithm	Frequency	Cost
Algorithm Fibonacci (n)	—	—
{ if (n ≤ 1) then	1	1
write n;	1	1
else		
{ fnm 2 = 0; fnm := 1;	1	1
for (i := 2 to n) do	1	n
{		
fn : fnm 1 + fnm 2;	n	1
fnm 2 : = fnm 1;	n	1
fnm 1 := fn;		
}		
write (fn);		
}		

Time Complexity $= 1 + 1 + 1 + (n-1) + (n(1) + n(1)$

$\qquad = 3 + n - 1 + n + n = 2 + 3n = \boxed{O(n)}$

(2) (a)

Algorithm	frequency	Cost
for i := 1 to n do	1	n+1
for j := 1 to i do	n	n+1
for k := 1 to j do	n×n	n+1
x := x+1	n×n×n	

Time Complexity $= n + 1 + n^2 + n + n^3$

$\qquad = n^3 + n^2 + 2n + 1$

$$\boxed{O(n^3)}$$

(b)

Algorithm	Frequency	Cost
i := 1	1	1
while (i ≤ n) do	n+1	1
{ x := x+1	n	1
i := j+	n	1
}		

$f(n) = n + n + n + 1 + 1 = 3n + 2$

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

Algorithm	Frequency	Cost
Algorithm Transpose (a,n)		
for (i=1 to n-1) do	n+1	1
for (j=i+1 to n) do	n(n+1)	1
{		
t= a [i,j];		
a [i,j]= a [j,i];		
a [j,i]= t;	n×n	1
}		

$$f(n) = (n+1) + (n^2+n) + n^2$$

$$= 2n^2 + 2n + 1$$

Time Complexity $= O(n^2)$