Algorithm one analysis

$\sum_{i=2}^{n} 1$ = n-2+1 $\approx O(n)$

Best case	Average case	Worst case
$\Omega(1)$	$\Theta(n)$	$O(n)$
When number of steps = 0 or 1	When number of steps > 1 to n	When number of steps > 1 to n

Algorithms two analysis

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

T(1) = 1, T(2) = 2

= 2T(n - 2) + b

= 2[T(n - 3) + T(n - 4) + b] + b by substituting T(n - 2) in (2)

$\geq$ 2[T(n - 4) + T(n - 4) + b] + b

= 2^2T(n - 4) + 2b + b

= 2^2[T(n - 5) + T(n - 6) + b] + 2b + b

by substituting T(n - 4) in (2)

$\geq 2^3T(n - 6) + (2^2 + 2^1 + 2^0)b$

. . .

$\geq 2^kT(n - 2k) + (2^{k-1} + 2^{k-2} + \ldots + 2^1 + 2^0)b$

$= 2^kT(n - 2k) + (2^k - 1)b$

Hence $T(n) \geq 2^{(n-2)/2}T(2) + [2^{(n-2)/2} - 1]b$

$= (b + c)2^{(n-2)/2} - b$

= [(b + c) / 2]*(2)$^{n/2}$ – b $\approx$ $\boldsymbol{O(2^n)}$

Best case	Average case	Worst case
$\Omega(1)$	$\Theta(2^n)$	$O(2^n)$
When number of steps = 1 or 2	When number of steps > 2	When number of steps > 2

Comparison between algorithm one and two

	Algorithm one	Algorithm two
Best case	$\Omega(1)$	$\Omega(1)$
Average case	$\Theta(\mathrm{n})$	$\Theta(2^n)$
Worst case	$\mathrm{O}(n)$	$O(2^n)$