4.

Ambos algoritmos tienen como entrada un arreglo y calculan y guardan en otro arreglo la suma de los elementos hasta el “i”. Es decir, dado el arreglo A, el elemento i del nuevo arreglo es la suma de los elementos A[i]+A[i-1]+...+A[1]+A[0]. El primer algoritmo usa 2 ciclos for, uno dentro del otro, para calcular mientras que el segundo usa solo uno.

5.

Algoritmo 1

	Numero de operaciones
int A[n],sum[n]	2n
for(int i = 0;i < n; i++) scanf(“%d”,&A[i]);	n
for(int i = 0;i < n; i++){	
int aux=0;	n
for(int j = 0;j <= i;j++){	
aux+=A[j];	1+2+...+(n-1)+n=n(n+1)/2
}	
sum[i]=aux;	n
}	

Luego, este algoritmo es de complejidad $O(n^2)$

Algoritmo 2

	Numero de operaciones
int A[n],sum[n]	2n
for(int i = 0;i < n; i++) scanf(“%d”,&A[i]);	n
sum[0]=A[0];	1
for(int i = 0;i < n; i++){	
sum[i]=sum[i-1]+A[i];	n
}	

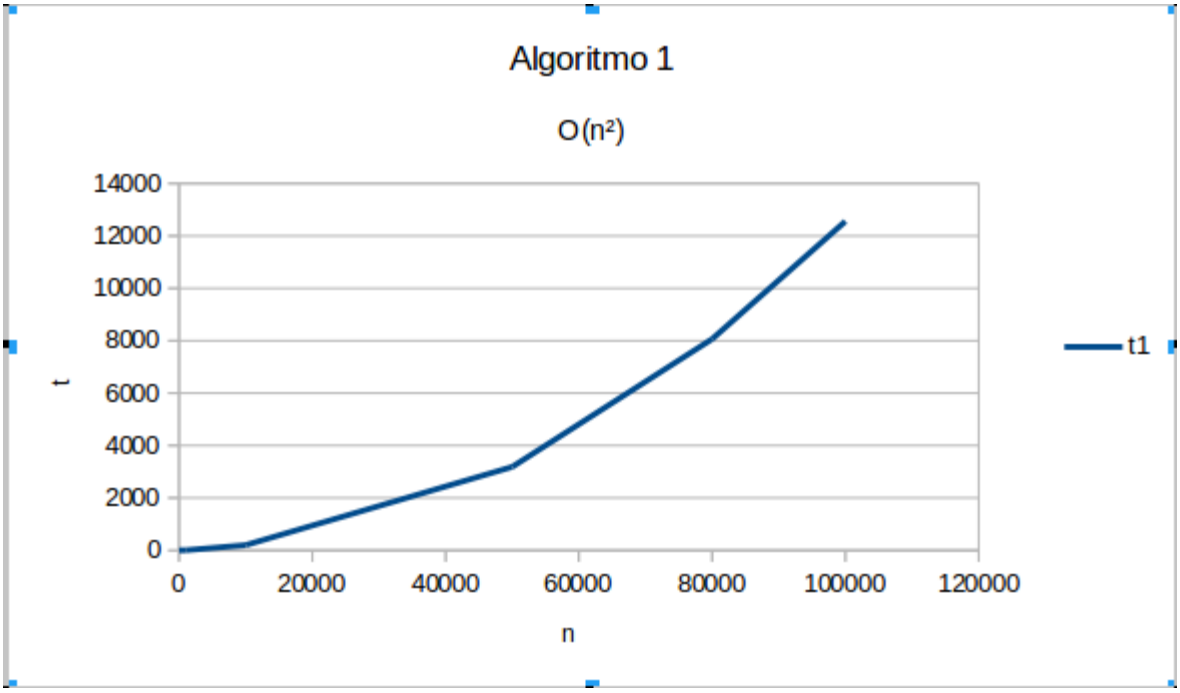
Este algoritmo es de complejidad O(n)

8.

Algoritmo 1

Datos

n	t1
10	0.004
100	0.048
1000	4.227
10000	200.091
50000	3186.31
80000	8063.57
100000	12556.4



Algoritmo 2

Datos

n	t2
10	0.001
100	0.002
1000	0.012
10000	0.035
50000	0.156
80000	0.252
100000	0.312