

ESTRUCTURAS DE DATOS Y ALGORITMOS
1^{er} SEMESTRE 2019
CONTROL 1

1. ¿Qué es la *Random Access Machine*? **(0.5 puntos)** ¿Para qué se usa en el análisis de algoritmos? **(0.5 puntos)**

Es una máquina (computador) genérico con características bien definidas como: un solo procesador, un conjunto estándar de instrucciones, tiempo constante de ejecución de instrucciones y una memoria única (sin jerarquía de memoria).

En el análisis de algoritmos se usa para analizar el tiempo que un algoritmo toma en ejecutarse, de modo que la plataforma de hardware sea la misma para todos los algoritmos.

2. Aparte del tiempo de ejecución, nombra al menos otros 2 aspectos de un algoritmo que pueden medirse para determinar su calidad **(1 punto)**

*Algunos otros aspectos **propios del algoritmo** que pueden medirse son:*

Uso de memoria

Uso de ancho de banda

Gasto de energía

Legibilidad

Nota: Los datos de entrada no son propios del algoritmo. Lo que uno hace es medir memoria, tiempo, energía etc. para ciertos datos de entrada. Pero los datos de entrada no se miden. Se mide rendimiento (uso memoria, tiempo, etc.) en función de los datos de entrada.

3. El análisis asintótico del tiempo de ejecución del algoritmo A dio por resultado que es $O(N^2)$. El mismo análisis para el algoritmo B dio por resultado que es $O(\log_2 N)$. En ambos casos, las funciones indican límites superiores ajustados. Indica si las siguientes afirmaciones son verdaderas o falsas y justifica.
- a. B es siempre mejor que A **(0.5 puntos)** *FALSO. Dependiendo del valor de N y de las constantes, podría ocurrir que A sea mejor que B en casos con N pequeños.*
 - b. A es también $O(N^3)$ **(0.5 puntos)**. *VERDADERO. Ya que $O(N^2)$ es un límite superior ajustado para A, entonces cualquier función que crezca a mayor velocidad que N^2 es también un límite superior para A.*
 - c. B es también $\Omega(1)$ **(0.5 puntos)**. *VERDADERO. Ya que $O(\log_2 N)$ es un límite superior ajustado para B, entonces cualquier función que crezca a menor velocidad que $\log_2 N$ es también un límite inferior para B.*
 - d. A es también $\Omega(\log_2 N)$ **(0.5 puntos)** *VERDADERO. Ya que $O(N^2)$ es un límite superior ajustado para A, entonces cualquier función que crezca a menor velocidad que N^2 es también un límite inferior para A.*

- e. B es también $O(N^2)$ **(0.5 puntos)**. VERDADERO. Ya que $O(\log_2 N)$ es un límite superior ajustado para B, entonces cualquier función que crezca a mayor velocidad que $\log_2 N$ es también un límite superior para B.

4. Realiza un análisis asintótico del siguiente algoritmo C, que recibe como argumento de entrada un arreglo A de tamaño N. **(1.5 puntos)**

```
function C(A)
  for 0 ≤ k < N
    if (A[k]%2==0)
      A[k]=A[k]+1
    end if
  end for
end function
```

Análisis de peor caso: todos los números del arreglo son pares, se ejecutan todas las instrucciones el máximo número de veces posible

	Seudo código	Costo	Veces
1.	function C(A)	---	
2.	for 0 ≤ k < N	C1	N+1
3.	if (A[k]%2==0)	C2	N
4.	A[k]=A[k]+1	C3	N
5.	end if	---	
6.	end for	---	
7.	end function	---	

$$T(N) = (C1 + C2 + C3) * N + C1 \rightarrow T(N) \text{ es } O(N)$$

Análisis de mejor caso: todos los números del arreglo son impares, no se ejecutan las instrucciones dentro del if

	Seudo código	Costo	Veces
1.	function C(A)	---	
2.	for 0 ≤ k < N	C1	N+1
3.	if (A[k]%2==0)	C2	N
4.	A[k]=A[k]+1	C3	0
5.	end if	---	
6.	end for	---	
7.	end function	---	

$$T(N) = (C1 + C2) * N + C1 \rightarrow T(N) \text{ es } \Omega(N)$$

Dado que $T(N)$ es $O(N)$ y $\Omega(N) \rightarrow T(N)$ es también $\Theta(N)$