

Nombre: Laura Isabel Olivero
Código: 897853
Fecha: 15/ 02/ 2022

Nombre: Nikolas Jimenez Sanchez
Código:8972212

Punto 1	
void algoritmo1(int n){	
int i, j = 1;	1
for(i = n * n; i > 0; i = i / 2){	$B = \log_2(n^2) + 2$
int suma = i + j;	B - 1
printf("Suma %d\n", suma);	B - 1
++j;	B - 1
}	
}	Complejidad: $O(\log_2(n^2))$

Cuando se le da 8 como valor a n el ciclo for se ejecuta n veces dando como resultado:

Suma 33

Suma 18

Suma 11

Suma 8

Suma 7

Suma 7

Suma 7

Ya que inicialmente $i = 64$ y $j = 1$

Después de la primera iteración será $i = 32$, $j = 2$ y suma = 33

Después de la segunda iteración será $i = 16$, $j = 3$ y suma = 18

Después de la tercera iteración será $i = 8$, $j = 4$ y suma = 11

Después de la cuarta iteración será $i = 4$, $j = 5$ y suma = 8

Después de la quinta iteración será $i = 2$, $j = 6$ y suma = 7

Después de la sexta iteración será $i = 1$, $j = 7$ y suma = 7

Después de la séptima iteración será $i = 0$, $j = 8$ y suma = 7

En la octava iteración verificara que la condición no se cumple y finaliza el ciclo

Nombre: Laura Isabel Olivero
 Código: 897853
 Fecha: 15/ 02/ 2022

Nombre: Nikolas Jimenez Sanchez
 Código:8972212

Punto 2	
int algoritmo2(int n){	
int res = 1, i, j;	3
for(i = 1; i <= 2 * n; i += 4)	$B = ((n+2)*2)/4$
for(j = 1; j * j <= n; j++)	$B^2 - 1$
res += 2;	$(n*2)/4$
return res;	1
	Complejidad: $O(((n+2)*2)/4)^2$

El resultado de algoritmo2(8) es 21 ya que:	
i = 1	J = 1, res = 3 J = 2, res = 5
i = 5	J = 1, res = 7 J = 2, res = 9
i = 9	J = 1, res = 11 J = 2, res = 13
i = 13	J = 1, res = 15 J = 2, res = 17
i = 17	J = 1, res = 19 J = 2, res = 21

Nombre: Laura Isabel Olivero
Código: 897853
Fecha: 15/ 02/ 2022

Nombre: Nikolas Jimenez Sanchez
Código:8972212

Punto 3	
void algoritmo3(int n){	
int i, j, k;	3
for(i = n; i > 1; i--)	n
for(j = 1; j <= n; j++)	n - 1
for(k = 1; k <= i; k++)	n - 2
printf("Vida cruel!!\n");	n/2
}	
	Complejidad: O(3.5n)

Nombre: Laura Isabel Olivero
Código: 897853
Fecha: 15/ 02/ 2022

Nombre: Nikolas Jimenez Sanchez
Código:8972212

Punto 5	
void algoritmo5(int n){	
int i = 0;	1
while(i <= n){	$n + 7 - n$
printf("%d\n", i);	$n + 6 - n$
i += n / 5;	$n + 6 - n$
}	
}	Complejidad: $O(7)$

Nombre: Laura Isabel Olivero
Código: 897853
Fecha: 15/ 02/ 2022

Nombre: Nikolas Jimenez Sanchez
Código:8972212

Punto 6

```
def fib(n):  
    if n < 2:  
        ans = n  
    else:  
        ans = fib(n-1) + fib(n-2)
```

Tamaño Entrada	Tiempo	Tamaño Entrada	Tiempo
5	0,044s	35	5,603s
10	0,069s	40	1m0,937s
15	0,086s	45	11m14,520s
20	0,066s	50	125m5,380s
25	0,097s	60	No calculado
30	0,566s	100	No calculado

Nombre: Laura Isabel Olivero
Código: 897853
Fecha: 15/ 02/ 2022

Nombre: Nikolas Jimenez Sanchez
Código:8972212

Punto 7

```
def fiposterioroCiclo(n):  
    anterior = 0  
    posterior = 1  
    for k in range(n-1):  
        sumador = anterior + posterior  
        anterior = posterior  
        posterior = sumador  
    return posterior
```

Tamaño entrada	time	Tamaño entrada	time
5	0.097s	45	0,045s
10	0.058s	50	0,082s
15	0,045s	100	0,088s
20	0,063s	200	0,054s
25	0,082s	500	0,045s
30	0,045s	1000	0,047s
35	0,046s	5000	0,070s
40	0,045s	10000	0,051s

Nombre: Laura Isabel Olivero
Código: 897853
Fecha: 15/ 02/ 2022

Nombre: Nikolas Jimenez Sanchez
Código:8972212

Punto 8

Tamaño entrada	Tiempo de Solución Propia	Tiempo de solución Profesores
100	0,068s	0,090s
1000	0,064s	0,060s
5000	0,120s	0,108s
10000	0,177s	0,149s
50000	0,843s	0,420s
100000	2,500s	0,862s
200000	5,894s	2,010s

Tiempos programa primos

en ambos casos la operación es tiene una complejidad de $O(n)$

los tiempos de ejecución son similares hasta cuando se le dé una entrada de 50000 con casi siempre dejando en claro que la solución de los profesores es más eficiente tal vez siendo incluso detenida por el propio comienzo de la operación en el computador, esto se soluciona a partir de la entrada anteriormente mencionada, desde allí se nota una tendencia a que la solución del profesor genere la respuesta en menos de la mitad del tiempo que la propuesta por nosotros