Alumno: Francisco Javier Bolívar Expósito

***1.** Usando la notación O, determinar la eficiencia de los siguientes segmentos de código:*

```
int n, j; int i=1; int x=0;                                  O(1)          |
do{                                                                        |
        j=1;                                        O(1)     |            |Bucle y producto
        while (j <=n){                              |Bucle y prod. |      |
                j=j*2;  O(1)    |Suma               |O(log2n)  |Suma        |O(n * log2n)
                x++;    O(1)    |O(1)               |          |O(log2n)    |
        }                                                      |            |
        i++;                                        O(1)     |            |
}while (i<=n);                                                             |
```

Eficiencia → **O(n log$_2$n)**

```
int n, j; int i=2; int x=0;
do{
        j=1;
        while (j <= i){
                j=j*2;
                x++;
        }
        i++;
}while (i<=n);
```

Aunque el orden de ejecución cambie, el bucle while (j <= i) se ejecuta el mismo número de veces que (j <= n) ya que la variable **i** aumenta progresivamente hasta **n**. El resto del código es igual que el segmento de código anterior y por lo tanto la eficiencia es igual también. Eficiencia → **O(n log$_2$n)**

***2.** Para cada función f(n) y cada tiempo t de la tabla siguiente, determinar el mayor tamaño de un problema que puede ser resuelto en un tiempo t (suponiendo que el algoritmo para resolver el problema tarda f(n) microsegundos, es decir, f(n) x 10^{-6} sg.)*

f(n)	***t***				
	1 sg.	1 h.	1 semana	1 año	1000 años
log$_2$n	10^{300000}	$2^{3600000000}$	$2^{(6x10^{\wedge}11)}$	$2^{(3,15x10^{\wedge}13)}$	$2^{(3,15x10^{\wedge}16)}$
n	1000000	3600000000	6,048 x 10^{11}	3,15 x 10^{13}	3,15 x 10^{16}
nlog$_2$n	6,27 x 10^4	1,33 x 10^8	1,77 x 10^{10}	7,9 x 10^{11}	6,4 x 10^{14}
n^3	100	1532	8456	31581	315817
2^n	19	31	39	44	54
n!	9	12	14	16	18