Ejercicio 1:

Obtenga la complejidad algorítmica temporal de foo1 en notación Gran O.
Asuma que inicio y fin son indices validos de arreglo.

```
void foo1(int* arreglo, int inicio, int fin) // ← T(n)
{
        if(inicio < fin) // ← 1
        {
                int mitad = (inicio + fin) / 2;   // ← 3
                foo1(arreglo, inicio, mitad);   // ← T(n/2)
                foo1(arreglo,  mitad + 1, fin); // ← T(n/2)
                invertir(arreglo, inicio,mitad, fin); // 14n + 4
        }
}

void invertir(int* &arreglo,int inicio, int mitad, int fin); // ← 14n + 4
{
        int x.y,z;
        for(x = inicio; x <= mitad; x++) // ← 3(n/2) + 2
        {
                y = mitad – x; // ← 2(n/2)
                arreglo[z] = arreglo[y]; // ← 3(n/2)
                arreglo[y] = arreglo[x]; // ← 3(n/2)
                arreglo[x] = arreglo[y]; // ← 3(n/2)
        }
        for(x = mitad+1; x < fin; x++) // ← 3(n/2) + 2
        {
                y = mitad – x; // ← 2(n/2)
                arreglo[z] = arreglo[y]: // ← 3(n/2)
                arreglo[y] = arreglo[x]; // ← 3(n/2)
                arreglo[x] = arreglo[y]; // ← 3(n/2)
        }
}
```

Ecuación de recurrencia:

$$T(1)=1$$

$$T(n)=2\mathrm{T}(\frac{n}{2})+14\mathrm{n}+8$$

$$T(\frac{n}{2})=2\mathrm{T}(\frac{n}{4})+14(\frac{n}{2})+8$$

$$T(n)=2*[2\mathrm{T}(\frac{n}{4})+14(\frac{n}{2})+8]+14\mathrm{n}+8$$

$$T(n)=2*2\mathrm{T}(\frac{n}{4})+2*14(\frac{n}{2})+14\mathrm{n}+2*8+8$$

$$T(n)=2^k T(\frac{n}{2^k})+\sum_{i=0}^{k-1}(2^i*14(\frac{n}{2^i})+2^i*8)$$

$T(n)=2^k T(\frac{n}{2^k})+14\text{nk}+8(2^k-1)$

$T(n)=2^k T(\frac{n}{2^k})+14\text{nk}+8*2^k-8$

$Si\ 2^k=n \rightarrow k=\log_2(n)$

$T(n)=n\,T(1)+14\text{n}\log_2(n)+8\text{n}-8$

$T(n)=14\text{n}\log_2(n)+9\text{n}-8 \in O(nlog_2(n))$

Ejercicio 2:

Obtenga la complejidad algorítmica temporal de foo2 en notación Gran O.
Asuma que arreglo tiene largo N.

```
double foo2(double* &arreglo, int N ) // ← T(n)
{
	if(N > 0) // ← 1
	{
		int x;
		double y;
		for(int x = 0; x < N/2; x++) // ← 4(n/2) + 3
		{
			arreglo[x] /= 2; // ← 3(n/2)
			y *= arreglo[x]; // ← 3(n/2)
		}
		for(int x = (N/2); x < N; x++) // ← 3(n/2) + 3
		{
			arreglo[x] *= 2; // ← 3(n/2)
			y /= arreglo[x]; // ← 3(n/2)
		}
		return y *  foo2(arreglo,N/4)  + y / foo2(arreglo,N/4) ; // ← 5 + 2T(n/4)
	}
	else
		return 1; // ← Descartado
}
```

Ecuacion de recurrencia:

$T(1)=1$

$T(n)=2\text{T}(\frac{n}{4})+\frac{19}{2}n+12$

$T(\frac{n}{4})=2\text{T}(\frac{n}{16})+\frac{19}{2}\frac{n}{4}+12$

$T(n)=2[2\text{T}(\frac{n}{16})+\frac{19}{2}\frac{n}{4}+12]+\frac{19}{2}n+12$

$T(n)=2*2\text{T}(\frac{n}{16})+2*\frac{19}{2}\frac{n}{4}+\frac{19}{2}n+2*12+12$

$$T(n)=2^k T(\frac{n}{4^k})+\sum_{i=0}^{k-1}(2^i \frac{19}{2}\frac{n}{4^i}+2^i*12)$$

$$T(n)=2^k T(\frac{n}{4^k})+\sum_{i=0}^{k-1}(2^i \frac{19}{2}\frac{n}{4^i})+12(2^k-1)$$

$$T(n)=2^k T(\frac{n}{4^k})+\frac{19\mathrm{n}}{2}\sum_{i=0}^{k-1}(\frac{2^i}{4^i})+12(2^k-1)$$

$$T(n)=2^k T(\frac{n}{4^k})+\frac{19\mathrm{n}}{2}\sum_{i=0}^{k-1}((\frac{2}{4})^i)+12(2^k-1)$$

$$T(n)=2^k T(\frac{n}{4^k})+\frac{19\mathrm{n}}{2}\sum_{i=0}^{k-1}(0.5^i)+12(2^k-1)$$

$$T(n)=2^k T(\frac{n}{4^k})+\frac{19\mathrm{n}}{2}(\frac{1-0.5^k}{1-0.5})+12(2^k-1)$$

$$T(n)=2^k T(\frac{n}{4^k})+19\mathrm{n}-19\mathrm{n}*0.5^k+12(2^k-1)$$

$$Si\ n=4^k \rightarrow k=\log_4(n)$$

$$T(n)=4^{\frac{1}{2}\log_4(n)}T(\frac{n}{n})+19\mathrm{n}-19\mathrm{n}*4^{\frac{-1}{2}\log_4(n)}+12(4^{\frac{1}{2}\log_4(n)}-1)$$

$$T(n)=n^{\frac{1}{2}}+19\mathrm{n}-19\mathrm{n}*n^{\frac{-1}{2}}+12(n^{\frac{1}{2}}-1)$$

$$T(n)=19\mathrm{n}-6\mathrm{n}^{\frac{1}{2}}-12\in O(n)$$

Ejercicio 3:

Obtenga la complejidad algorítmica temporal de foo3 en notación Gran O.
Asuma que arreglo tiene largo N.

```
int foo3(int* &arreglo,int N) // ← T(n)
{
	if(N > 0) // ← 1
	{
		for(int y,x = 0; x < N; x++) // ← 3n+2
		{
			for(int y = 0; y < N; y++) //    3n^2+2n
			{
				if(x < y) //    n^2
				{
					arreglo[x] += arreglo[y]; // ← Descartado
				}
				else if(x > y)    n^2
				{
					arreglo[x] -= arreglo[y];    4(n^2)
				}
			}
		}
		return arreglo[N – 1] * foo3(arreglo, N – 2) + 7* foo3(arreglo, N – 2) ; // 7 + 2T(n - 2)
	}
```

```
        else
        {
                return 1; // ← Descartado
        }
}
```

Ecuacion de recurrencia:

$$T(0)=1$$
$$T(n)=2\mathrm{T}(n-2)+9\mathrm{n}^2+5\mathrm{n}+10$$
$$T(n-2)=2\mathrm{T}(n-4)+9(n-2)^2+5(n-2)+10$$
$$T(n)=2[2\mathrm{T}(n-4)+9(n-2)^2+5(n-2)+10]+9\mathrm{n}^2+5\mathrm{n}+10$$
$$T(n)=2*2\mathrm{T}(n-4)+2*9(n-2)^2+9\mathrm{n}^2+2*5(n-2)+5\mathrm{n}+2*10+10$$
$$T(n)=2^k T(n-2^k)+\sum_{i=0}^{k-1}(2^i*9(n-2^i)^2)+\sum_{i=0}^{k-1}(2^i*5(n-2^i))+\sum_{i=0}^{k-1}(2^i*10)$$
$$T(n)=2^k T(n-2^k)+\sum_{i=0}^{k-1}(2^i*9(n^2-2\mathrm{n}*2^i+4^i))+\sum_{i=0}^{k-1}(2^i*5(n-2^i))+\sum_{i=0}^{k-1}(2^i*10)$$
$$T(n)=2^k T(n-2^k)+\sum_{i=0}^{k-1}(9\mathrm{n}^2*2^i-18\mathrm{n}*4^i+9*8^i)+\sum_{i=0}^{k-1}(5\mathrm{n}*2^i-5*4^i)+10(2^k-1)$$
$$T(n)=2^k T(n-2^k)+9\mathrm{n}^2\sum_{i=0}^{k-1}(2^i)-18\mathrm{n}\sum_{i=0}^{k-1}(4^i)+9\sum_{i=0}^{k-1}(8^i)+5\mathrm{n}\sum_{i=0}^{k-1}(2^i)-5\sum_{i=0}^{k-1}(4^i)+10(2^k-1)$$
$$T(n)=2^k T(n-2^k)+9\mathrm{n}^2(2^k-1)-18\mathrm{n}\left(\frac{1-4^k}{1-4}\right)+9\left(\frac{1-8^k}{1-8}\right)+5\mathrm{n}(2^k-1)-5\left(\frac{1-4^k}{1-4}\right)+10(2^k-1)$$
$$T(n)=2^k T(n-2^k)+9\mathrm{n}^2(2^k-1)-6\mathrm{n}(4^k-1)+\frac{9}{7}(8^k-1)+5\mathrm{n}(2^k-1)-\frac{5}{3}(4^k-1)+10(2^k-1)$$
$$T(n)=2^k T(n-2^k)+9\mathrm{n}^2(2^k-1)-6\mathrm{n}(2^{2\mathrm{k}}-1)+\frac{9}{7}(2^{3\mathrm{k}}-1)+5\mathrm{n}(2^k-1)-\frac{5}{3}(2^{2\mathrm{k}}-1)+10(2^k-1)$$
$$Si\ n=2^k \rightarrow k=\log_2(n)$$
$$T(n)=n\,T(n-n)+9\mathrm{n}^2(2^{\log_2(n)}-1)-6\mathrm{n}(2^{2\log_2(n)}-1)+\frac{9}{7}(2^{3\log_2(n)}-1)+5\mathrm{n}(2^{\log_2(n)}-1)-\frac{5}{3}(2^{2\log_2(n)}-1)+10(2^{\log_2(n)}-1)$$
$$T(n)=n+9\mathrm{n}^2(n-1)-6\mathrm{n}(n^2-1)+\frac{9}{7}(n^3-1)+5\mathrm{n}(n-1)-\frac{5}{3}(n^2-1)+10(n-1)$$
$$T(n)=n+9\mathrm{n}^3-9\mathrm{n}^2-6\mathrm{n}^3+6\mathrm{n}+\frac{9}{7}n^3-\frac{9}{7}+5\mathrm{n}^2-5\mathrm{n}-\frac{5}{3}n^2+\frac{5}{3}+10\mathrm{n}-10$$
$$T(n)=\frac{30}{7}n^3+\frac{-17}{3}n^2+12\mathrm{n}+\frac{-202}{21}\in O(n^3)$$

Ejercicio 4:

Obtenga la complejidad algorítmica temporal de foo4 en notación Gran O.

```
int foo4(int* N) // ← T(n)
{
	if(N > 0) // ← 1
	{
		int y = potencia(0.5,N),z = 0,x; // ← (5n + 3) + 2
		for(x = 0; x < y; x++) // ← 3(0) + 2
		{
			z++;	// ← 0
		}
		return z – foo4(N – 1); // ← 2 + T(n – 1);
	}
	else
		return 0;
}

int potencia(float base,int exp) // 5n + 3
{
	float aux = base; // ← 1
	for(int x = 0; x < exp; x++) // ← 3n + 2
	{
		float *= aux; // ← 2n
	}
	return int(base);
}
```

Ecuacion de recurrencia:

$$T(0)=1$$
$$T(n)=T(n-1)+5\text{n}+10$$
$$T(n-1)=T(n-2)+5(n-1)+10$$
$$T(n)=[T(n-2)+5(n-1)+10]+5\text{n}+10$$
$$T(n)=T(n-2)+5\text{n}+5(n-1)+10+10$$
$$T(n)=T(n-k)+\sum_{i=0}^{k-1}5(n-i)+10\text{k}$$
$$T(n)=T(n-k)+5\text{nk}-5\sum_{i=0}^{k-1}i+10\text{k}$$
$$T(n)=T(n-k)+5\text{nk}-5[\frac{(k-1)k}{2}]+10\text{k}$$
$$T(n)=T(n-k)+5\text{nk}-\frac{5\text{k}^2}{2}+\frac{5\text{k}}{2}+10\text{k}$$
$$T(n)=T(n-k)+5\text{nk}-\frac{5\text{k}^2}{2}+12{,}5\,k$$

Si k = n

$$T(n) = T(0) + 5n^2 - \frac{5n^2}{2} + 12{,}5\,n$$

$$T(n) = \frac{5n^2}{2} + 12{,}5\,n + 1 \in O(n^2)$$