

### 2.1.1 Multiplicación entera

Continuando con el concepto de divide y vencerás, el presente caso toma la subdivisión del problema, y aplicar lo visto en el punto anterior, en nuestro caso se trata de hacer multiplicaciones de números enteros grandes, en donde el algoritmo clásico se muestra a continuación

$$\begin{aligned}1234 * 5678 &= 1234 * (5*1000 + 6*100 + 7*10 + 8) \\ &= 1234 * 5 * 1000 + 1234 * 6 * 100 + 1234 * 7 * 10 + 1234 * 8\end{aligned}$$

En donde las operaciones básicas son:

Multiplicación de dígitos  $O(1)$

Suma de dígitos  $O(1)$

Desplazamientos  $O(1)$

Y la eficiencia del algoritmo es  $O(n^2)$

Ahora empleando el algoritmo de divide y vencerás, se tiene

$$1234 = 12 * 100 + 34$$

$$5678 = 56 * 100 + 78$$

$$\begin{aligned}1234 * 5678 &= (12 * 100 + 34) * (56 * 100 + 78) \\ &= 12 * 56 * 10000 + (12 * 78 + 34 * 56) * 100 + 34 * 78\end{aligned}$$

De lo anterior se puede notar se reduce la multiplicación de 4 cifras a:

- 4 multiplicaciones de dos cifras
- Tres sumas
- Varios desplazamientos

De modo que el algoritmo divide y vencerás en su versión simple sería:

Multiplicación de enteros de n cifras

#### 1. Dividir

$$\begin{aligned}x &= 1234 * 5678 = x_i * 10^4 + x_d & x_i &= 1234 & x_d &= 5678 \\ y &= 2468 * 0123 = y_i * 10^4 + y_d & y_i &= 2468 & y_d &= 0135\end{aligned}$$

#### 2. Combinar

$$\begin{aligned}x * y &= (x_i * 10^4 + x_d) * (y_i * 10^4 + y_d) \\ &= x_i * y_i * 10^8 + (x_i * y_d + x_d * y_i) * 10^4 + x_d * y_d\end{aligned}$$

### 3. De forma genérica

$$x = (x_i \cdot 10^{n/2} + x_d)$$

$$y = (y_i \cdot 10^{n/2} + y_d)$$

$$\begin{aligned} x * y &= (x_i \cdot 10^{n/2} + x_d) * (y_i \cdot 10^{n/2} + y_d) \\ &= x_i * y_i * 10^n + (x_i * y_d + x_d * y_i) * 10^{n/2} + x_d * y_d \end{aligned}$$

Ahora en código Java 2.1.1.1 se tiene el algoritmo:

#### **Código 2.1.1.1** Multiplicación de enteros grandes

```
/**
 *
 * @author Saul De La O Torres
 */
public class MultiplicadorEntero {
    public int multiplicar( int u, int v ) {
        int w, x, y, z;
        int n = obtenerMayorEnTamanioDeDigitos(u, v);
        if( esPequenio( n ) ) {
            return u * v;
        }
        // Dividir
        int s = n / 2;
        double temporal = Math.pow(10, s);
        w = u / (int)temporal;
        x = u % (int)temporal;
        y = v / (int)temporal;
        z = v % (int)temporal;
        // combinar
        return multiplicar( w, y )*(int)Math.pow(10, 2*s) +
            (multiplicar( w, z )+multiplicar( x, y ))*(int)Math.pow(10, s) +
            multiplicar( x, z );
    }
    private int obtenerMayorEnTamanioDeDigitos(int x, int y) {
        String xString = "" + x;
        String yString = "" + y;
        if( xString.length()>=yString.length() ) {
            return xString.length();
        }
        return yString.length();
    }
}
/**
 * El numero es pequenio si es de un digito
 *
 * @param x el entero donde se verifica el numero de digitos que tiene
```

```

*
* @return devuelve true si es pequeño y false en caso contrario
*/
private boolean esPequenio( int numero ) {
    String xString = "" + numero;
    return xString.length()<=1;
}
}

```

### Tarea.

1. Crear una clase principal que demuestre que el código anterior funcione correctamente
2. En Java hay un tipo de dato primitivo mayor que int y una clase que trabaja enteros muy grandes, crear los métodos sobrecargados en la clase que permita multiplicar los enteros del tipo correspondiente.
3. Hay que entregar un reporte y las clases creadas en un archivo comprimido como sigue:

apellidoPaterno\_apellidoMaterno\_Nombre(s)\_MultiplicacionEnteros.zip