

Multiplicación de Enteros Largos

PRACTICAS DIVIDE Y VENCERAS
2011/2012

Multiplicación de Enteros Largos

- Chequear si un número es primo requiere muchas multiplicaciones de enteros largos (desde dos a millones de dígitos)
- Para resolver este problema debemos implementar algoritmos eficientes capaces de trabajar con estos valores
 - Método clásico (escuela)
 - Método basado en Divide y Vencerás

Algoritmo clásico

Tamaño: n = número dígitos

- Algoritmo clásico: $1234 * 5678 =$
 $1234 * [5 * 1000 + 6 * 100 + 7 * 10 + 8] =$

Operaciones básicas:

- Multiplicaciones de dígitos $O(1)$;
 - Sumas de dígitos $O(1)$
 - Desplazamientos $O(1)$
- Eficiencia algoritmo: $O(n^2)$

Mult. Enteros Largos D&V

- Para aplicar D&V debemos de poder obtener la solución en base a problemas de tamaño menor

- Truco:

- $5632 = 56 \cdot 100 + 32$ y $3427 = 34 \cdot 100 + 27$

- $(56 \cdot 100 + 32) \cdot (34 \cdot 100 + 27) =$

Se reducen las dos multiplicaciones de 4 cifras a cuatro multiplicaciones de 2 cifras, mas tres sumas y varios desplazamientos

$$56 \cdot 32 \cdot 10000 + (56 \cdot 27 + 32 \cdot 34) \cdot 100 + (32 \cdot 27)$$

Divide y Vencerás básico

Dividir

$$X=12345678$$

$$x_i = 1234 \quad x_d = 5678$$

$$X = x_i \cdot 10^4 + x_d$$

$$Y = 24680135$$

$$y_i = 2468 \quad y_d = 0135$$

$$Y = y_i \cdot 10^4 + y_d$$

Combinar

$$\begin{aligned} X \times Y &= [x_i 10^4 + x_d] \times [y_i 10^4 + y_d] \\ &= x_i y_i 10^8 + (x_i y_d + x_d y_i) 10^4 + x_d y_d \end{aligned}$$

Mult. Enteros Largos D&V

- En general,
 - $X = x_i \cdot 10^{n/2} + x_d \cdot 10^{n/2}$
 - $Y = y_i \cdot 10^{n/2} + y_d \cdot 10^{n/2}$
 - $X \cdot Y = (x_i \cdot y_i) \cdot 10^n + (x_i \cdot y_d + x_d \cdot y_i) \cdot 10^{n/2} + x_d \cdot y_d$

```

Función DV_básico (X,Y,n) {
    if P es pequeño return X*Y;
    else {
        Obtener xi, xd, yi, yd;           //DIVIDIR

        z1 = DV_básico (xi, yi, n/2);
        z2 = DV_básico (xi, yd, n/2);
        z3 = DV_básico (xd, yi, n/2);
        z4 = DV_básico (xd, yd, n/2);

        aux= Sumar(z2,z3);                //COMBINAR
        z1  = Desplazar_Dcha(z1,n);
        aux = Desplazar_Dcha(aux,n/2);
        z = Sumar(z1,aux,z4 );
        return z;
    }
}

```

Eficiencia

```
Función DV_basico (X,Y,n) {  
    if P es pequeño return X*Y;  
    else {  
        Obtener xi, xd, yi, yd;  
        z1 = DV_basico (xi,yi,n/2);  
        z2 = DV_basico (xi,yd,n/2);  
        z3 = DV_basico (xd,yi,n/2);  
        z4 = DV_basico (xd,yd,n/2);  
  
        aux= Sumar(z2,z3);  
        z1  = Desplazar_Dcha(z1,n);  
        aux = Desplazar_Dcha(aux,n/2);  
        z = Sumar(z1,aux,z4);  
        return z;  
    }  
}
```

$O(1)$

$O(n)$

$T(n/2)$

$T(n/2)$

$T(n/2)$

$T(n/2)$

$O(n)$

$O(n)$

$O(n)$

$O(n)$

Eficiencia del algoritmo DV_bas

- $T(n) = 4T(n/2) + n$

$T(n)$ está en el orden $O(n^2)$

El cuello de botella está en el número de multiplicaciones de tamaño $n/2 \Rightarrow 4$

Para mejorar la eficiencia necesitamos reducir el número de multiplicaciones que hacemos.

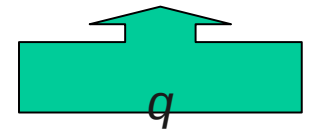
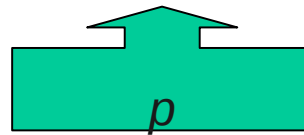
Mult. Enteros Largos D&V

■ Considerar

$$r = (x_i + x_d) * (y_i + y_d) = (x_i * y_i) + (x_i * y_d + x_d * y_i) + x_d * y_d$$



Luego, podemos calcular



$$X * Y = p * 10^n + (r - p - q) * 10^{n/2} + q$$

1 multiplicación tamaño n --> 3 mult. tamaño $n/2$

Ahorramos tiempo al operar ?

- ♦ Supongamos $X*Y = p*10^n + (r-p-q)*10^{n/2} + q$
 - ♦ Algoritmo Clásico (AC): $h(n) = c n^2$
 - ♦ Sea $g(n)$ operaciones en el algoritmo DV excepto las 3 mutiplicaciones de tamaño $n/2$.
- ♦ Ecuación DV con el AC para tamaño $n/2$:
 - ♦ $3h(n/2) + g(n) = 3c(n/2)^2 + g(n) = \frac{3}{4} cn^2 + g(n) = \frac{3}{4} h(n) + g(n)$
- ♦ Como $h(n)$ es $O(n^2)$ y $g(n)$ es $\square O(n) \rightarrow \Rightarrow$ ahorro 25%.
- ♦ Ganancia de tiempo – No dismunición de orden
- ♦ ¿Cómo resolver los subcasos?

```

Función DV (X,Y,n) {
  if P es pequeño return X*Y;
  else {
    Obtener xi, xd, yi, yd;           //DIVIDIR
    s1 = Sumar(xi,xd);
    s2 = Sumar(yi,yd);

    p = DV (xi,yi,n/2);
    q = DV (xd,yd,n/2);
    r = DV (s1,s2,n/2);

    aux = Sumar(r,-p,-q);           //COMBINAR
    p = Desplazar_Dcha(p,n);
    aux = Desplazar_Dcha(aux,n/2);
    z = Sumar(p,aux,q);
    return z;
  }
}

```

Eficiencia Divide y Vencerás

$$T(n) = 3T(n/2) + 8n = 3T(n/2) + O(n)$$

$$T(n) \in O(n^{\log_2 3}) = O(n^{1.585})$$

n	N^2	$N^{1.585}$
10	100	38.46
100	10000	1479.11
1000	1000000	56885.29
10000	100000000	2187751.62

D&V: Umbrales

Mult. Enteros Largos D&V

Si umbral es igual a 1, entonces

D&V (5.000 cifras) \Rightarrow 41 seg.

Clásico (5.000 cifras) \Rightarrow 25 seg

A partir de 32.789 cifras es mejor D&V (15 minutos !!!)

Si umbral es igual a 64

D&V (5.000 cifras) \Rightarrow 6 seg.

D&V(32.789 cifras) \Rightarrow 2 minutos !!

Selección umbral es problemática:

Depende del algoritmo y de la implementación

Se estima empíricamente.