Taller ProgComp: Track Matemática

Exponenciación Binaria

Gabriel Carmona Tabja

Universidad Técnica Federico Santa María,
Università di Pisa

April 15, 2024

Part I

Elevar un número

ELEVAR UN NÚMERO

Problema

Dado un dos enteros x e y, determinar x^y.

Ejemplos

- $x = 2, y = 4, x^y = 16$
- $x = 3, y = 23, x^y = 94143178827$

SOLUCIÓN INICIAL

```
long long pow(long long x, long long y, long long mod) {
  ll res = 1;
  for(int i = 0; i < y; i++) {
    res = (res * x) % mod;
  }
  return res;
}
```

SOLUCIÓN INICIAL

```
long long pow(long long x, long long y, long long mod) {
  ll res = 1;
  for(int i = 0; i < y; i++) {
    res = (res * x) % mod;
  }
  return res;
}
```

¿Cuál es la complejidad?

Solución inicial

```
long long pow(long long x, long long y, long long mod) {
  ll res = 1;
  for(int i = 0; i < y; i++) {
    res = (res * x) % mod;
  }
  return res;
}
```

¿Cuál es la complejidad? $O(y)$
¿Será bueno?

SOLUCIÓN INICIAL

```
long long pow(long long x, long long y, long long mod) {
  ll res = 1;
  for(int i = 0; i < y; i++) {
    res = (res * x) % mod;
  }
  return res;
}
```

¿Cuál es la complejidad? $O(y)$
¿Será bueno?

- Si $y = 10^4$, soportable
- Si $y = 10^9$, lo perdimos todo

Part II

Exponenciación Binaria

Exponenciación Binaria

Propiedad

$$x^y = x^{\frac{y}{2}\cdot 2} = (x^{\frac{y}{2}})^2$$

Exponenciación Binaria

Propiedad

$$x^y = x^{\frac{y}{2} \cdot 2} = (x^{\frac{y}{2}})^2$$

Entonces,

$$x^y = \begin{cases} x^{y/2} \cdot x^{y/2} & \text{si } y \text{ es par} \\ x^{y/2} \cdot x^{y/2} \cdot x & \text{si } y \text{ es impar} \end{cases}$$

Tenemos un algoritmo de *Divide and Conquer*

Código

```
// implementacion recursiva
long long binpow(long long x, long long y, long long mod) {
  if(y == 0) return 1;

  long long temp = binpow(x, y / 2, mod);
  if(y % 2) {
    return (x * ((temp * temp) % mod)) % mod;
  }
  return (temp * temp) % mod;
}

// implementacion iterativa
long long binpow(long long x, long long y, long long mod) {
  long long res = 1;
  while(y > 0) {
    if(y % 2) {
      res = (res * x) % mod;
    }
    x = (x * x) % mod;
    y /= 2;
  }
  return res;
}
```

¿Cuál es la complejidad?

Código

```cpp
// implementacion recursiva
long long binpow(long long x, long long y, long long mod) {
  if(y == 0) return 1;

  long long temp = binpow(x, y / 2, mod);
  if(y % 2) {
    return (x * ((temp * temp) % mod)) % mod;
  }
  return (temp * temp) % mod;
}

// implementacion iterativa
long long binpow(long long x, long long y, long long mod) {
  long long res = 1;
  while(y > 0) {
    if(y % 2) {
      res = (res * x) % mod;
    }
    x = (x * x) % mod;
    y /= 2;
  }
  return res;
}
```

¿Cuál es la complejidad?

$O(\log y)$, mucho mejor que lo anterior :)

CÓDIGO

```
// implementacion recursiva
long long binpow(long long x, long long y, long long mod) {
  if(y == 0) return 1;

  long long temp = binpow(x, y / 2, mod);
  if(y % 2) {
    return (x * ((temp * temp) % mod)) % mod;
  }
  return (temp * temp) % mod;
}

// implementacion iterativa
long long binpow(long long x, long long y, long long mod) {
  long long res = 1;
  while(y > 0) {
    if(y % 2) {
      res = (res * x) % mod;
    }
    x = (x * x) % mod;
    y /= 2;
  }
  return res;
}
```

¿Cuál es la complejidad?

$O(\log y)$, mucho mejor que lo anterior :)

También se puede aplicar a matrices, es cosa de definir la operación multiplicación :).

References I