Data Structures and Algorithms

Spring 2008-2009

Outline

Outline

Faster than $n - 1$ Multiplications?

- How much time is required to compute x^n?
- We can do better than obvious $n - 1 = O(n)$ multiplications
- $x^n = (x^{\frac{n}{2}})^2$ (n even) or $x^n = x(x^{\frac{n}{2}})^2$ (n odd)
- Decompose n into its binary representation and *implicitly* multiply appropriate powers of x

Compute x^{27}

Binary repn. of 27 : $1 * 2^4 + 1 * 2^3 + 1 * 2^1 + 1 * 2^0$

$$\begin{aligned} x^{27} &= x^{16} * x^8 * x^2 * x \\ &= (x^8 * x^4 * x)^2 * x \\ &= ((x^4 * x^2)^2 * x)^2 * x \\ &= (((x^2 * x)^2)^2 * x)^2 * x \end{aligned}$$

General Case

In general,

$$
\begin{aligned}
n &= a_{k-1}2^{k-1} + a_{k-2}2^{k-2} + \cdots + a_1 2^1 + a_0, \quad k = \lceil \log n \rceil \\
&= \sum_{i=0}^{\lceil \log n \rceil} a_i 2^i, \text{ where } a_i = \{0, 1\} \\
x^n &= x^{\sum_{i=0}^{\lceil \log n \rceil} a_i 2^i} \\
&= (x^{\sum_{i=1}^{\lceil \log n \rceil} a_i 2^i}) x^{a_0} \\
&= (x^{2\sum_{i=1}^{\lceil \log n \rceil} a_i 2^{i-1}}) x^{a_0} \\
&= (x^{\sum_{i=1}^{\lceil \log n \rceil} a_i 2^{i-1}})(x^{\sum_{i=1}^{\lceil \log n \rceil} a_i 2^{i-1}}) x^{a_0}
\end{aligned}
$$

An Efficient Exponentiation Function

```
long int
pow(const long int& x, const int n)
{
  if (n == 0) return 1;
  if (n == 1) return x;  // speed-up; not necessary

  if (n%2 == 0) return pow(x*x, n/2);  // n is even
  else return pow(x*x, n/2) * x;
}
```

Running time is $O(\log n)$ since

1. no. of recursive calls is $\log n$ (2nd param is $n/2$)
2. at most two mults done in each recursive call