class code: hljk4tm

Integer multiplication

Primary school algorithm takes $O(n^2)$ time

Karatsuba algorithm

```
karatsuba(x, y)                                   — T(n)
 if (n==1)                                        θ(1)
    return x*y                                    θ(1)
 else                                             θ(1)
    a, b = first and second half of x             O(n)
    c, d =   ''      ''     ''   ''  of y         O(n)
    compute  p = a+b                              O(n)
       ''    q = c+d                              O(n)
    ac = karatsuba(a, c)                          T(n/2)
    bd = karatsuba(b, d)                          T(n/2)
    pq = karatsuba(p, q)                          T(n/2)
    adbc = pq - ac - bd                           O(n)
    return 10^n ac + 10^(n/2) adbc + bd           O(n)
```

overall

$T(n) = 3T(\frac{n}{2}) + cn$

$T(n) = \theta(n^{\log_2 3})$

$\log_2 3 \approx 1.59$

$T(n) = \theta(n^{1.59})$

matrix multiplication

Input: $A = [a_{ij}]$, $B = [b_{ij}]$

output: $C = [c_{ij}] = A \cdot B$

$$\begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1n} \\ \vdots & & c_{ij} & \\ c_{n1} & c_{n2} & \cdots & c_{nn} \end{bmatrix} = \begin{array}{c} \\ i \\ \\ \end{array} \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \boxed{\quad\quad\quad\quad} & & \\ a_{n1} & \cdots & a_{nn} \end{bmatrix} \times \begin{array}{c} \quad j \\ \begin{bmatrix} b_{11} & \cdots & \boxed{\;} & \cdots & b_{1n} \\ & & & & \\ b_{n1} & \cdots & & \cdots & b_{nn} \end{bmatrix} \end{array}$$

$$c_{ij} = a_{i1} b_{1j} + a_{i2} b_{2j} + \cdots + a_{in} b_{nj}$$

$$= \sum_{k=1}^{n} a_{ik} b_{kj}$$

mat-mult(A, B)

```
for i = 1 to n
    for j = 1 to n
        c_ij = 0
        for k = 1 to n
            c_ij = c_ij + a_ik · b_kj

return C
```

Time: $O(n^3)$

can we do better?

Divide and conquer

$$C = AB$$

$$\begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \times \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix}$$

$$\left.\begin{aligned} C_{11} &= A_{11}B_{11} + A_{12}B_{21} \\ C_{12} & \\ C_{21} & \\ C_{22} & \end{aligned}\right\}$$

mat-mul (A, B)
if ($n == 1$)

$c_{11} = a_{11} b_{11}$

else
partition A into A_{11} A_{12} A_{21} A_{22}
" B " B_{11} B_{12} B_{21} B_{22}

$C_{11} =$ ________________

$C_{12} =$ ____________

$C_{21} =$ ________

$C_{22} =$ ____________

return C

Total time

$T(n) = 8T(\frac{n}{2}) + O(n^2)$

$T(n) = O(n^3)$

No improvement

Idea: need to reduce # of recursive multiplication.

Strassen's algorithm (1969). It uses 7 multiplications.

$$E_1 = A_{11}(B_{12} - B_{21})$$
$$E_2 = (A_{11} + A_{12}) B_{22}$$
$$E_3 = (A_{21} + A_{22}) B_{11}$$
$$E_4 = A_{22}(B_{21} - B_{11})$$
$$E_5 = (A_{11} + A_{12})(B_{11} + B_{22})$$
$$E_6 = (A_{12} - A_{22})(B_{21} + B_{22})$$
$$E_7 = (A_{11} - A_{21})(B_{11} + B_{12})$$

$$\begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix} = \begin{bmatrix} E_5 + E_4 - E_2 + E_6 & E_1 + E_2 \\ E_3 + E_4 & E_5 + E_1 - E_3 - E_7 \end{bmatrix}$$

Total time:

$$T(n) = 7T\left(\frac{n}{2}\right) + O(n^2)$$

$$T(n) = O(n^{\log_2 7})$$
$$\approx O(n^{2.808})$$

Progress of the algorithm

$O(n^3)$ — standart

$O(n^{2.808})$ — Strassen 1969

$O(n^{2.796})$ — Pan (1978)

$O(n^{2.522})$ — schonhage (1981)

$O(n^{2.517})$ — Romani (1982)

$O(n^{2.496})$ — coppersmith and winogard (1982)

$O(n^{2.479})$ — strassen (1986)

$O(n^{2.376})$ — coppersmith and winogard (1989)

$O(n^{2.374})$ — strothers (2010)

$O(n^{2.3728642})$ — V williams (2011)

$O(n^{2.3728639})$ — Le Gall (2014)

.

.

.

Powering a number

Problem comput a^n where $n \in \mathbb{N}$

Algorithm: multiply a n times.

Time $O(n)$

can we do better?

Divide and conquer

$$a^n = \begin{cases} a^{n/2} \cdot a^{n/2} & \text{if } n \text{ is even} \\ a^{n/2} \cdot a^{n/2} \cdot a & \text{if } n \text{ is odd} \end{cases}$$

$T(n) = 2T(\frac{n}{2}) + \Theta(1)$

$\Theta(n)$

```
power(a, n)
  if(n == 1)
      return a
  else
    tmp = power(a, n/2)
     if n is even
          return tmp * tmp
     else
          return tmp * tmp * a
```

$$T(n) = T\left(\frac{n}{2}\right) + \theta(1)$$

$$T(n) = \theta(\lg n)$$

Ti