MA/CSSE 473 Day 17

Divide-and-conquer Convex Hull

Strassen's Algorithm: Matrix Multiplication

MA/CSSE 473 Day 17

- **Student Questions**

- Convex Hull (Divide and Conquer)
- Matrix Multiplication (Strassen)

Reminder: The Master Theorem

- The Master Theorem for Divide and Conquer recurrence relations:
- Consider

$$T(n) = aT(n/b) + \Theta(n^k)$$

- The solution is (40 is the highest possible)
  - $\Theta(n^k)$ if $a < b^k$
  - $\Theta(n^k \log n)$ if $a = b^k$
  - $\Theta(n^{\log_b a})$ if $a > b^k$

Convex Hull Problem

- Again, sort by x-coordinate, with tie going to larger y-coordinate.

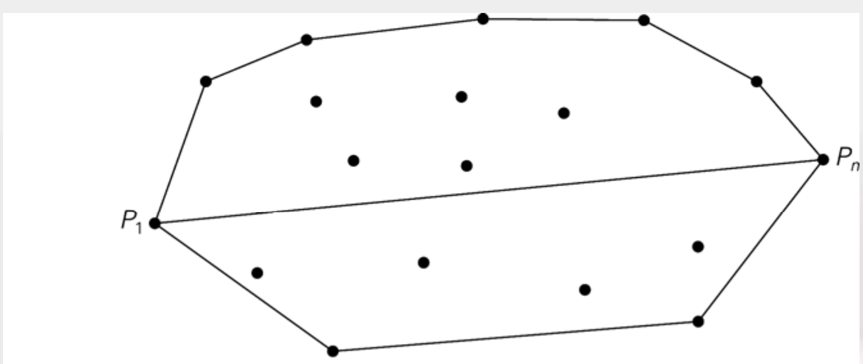

FIGURE 4.8 Upper and lower hulls of a set of points

Recursive calculation of Upper Hull

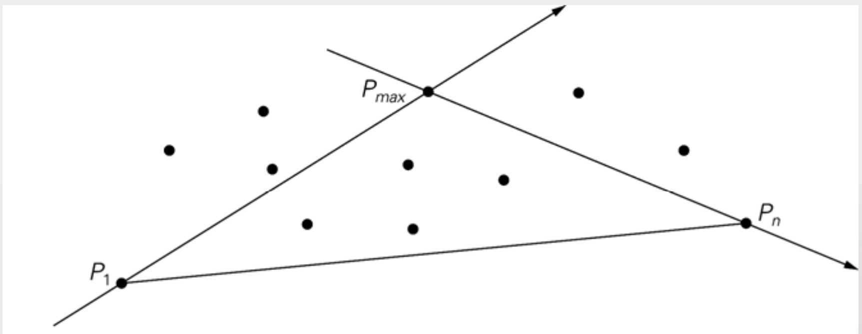

FIGURE 4.9 The idea of quickhull

Simplifying the Calculations

We can simplify two things at once:

- Finding the distance of P from line $P_1P_{2, \text{and}}$
- Determining whether P is "to the left" of P_1P_2
  - The area of the triangle through $P_1=(x_1,y_1)$, $P_2=(x_2,y_2)$, and $P_3=(x_3,y_e)$ is ½ of the absolute value of the determinant

$$\begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix} = x_1y_2 + x_3y_1 + x_2y_3 - x_3y_2 - x_2y_1 - x_1y_3$$

    - For a proof of this property, see http://mathforum.org/library/drmath/view/55063.html
    - How do we use this to calculate distance from P to the line?
  - The sign of the determinant is positive if the order of the three points is clockwise, and negative if it is counter-clockwise
    - Clockwise means that P_3 is "to the left" of directed line segment P_1P_2
- Speeding up the calculation

Efficiency of quickhull algorithm

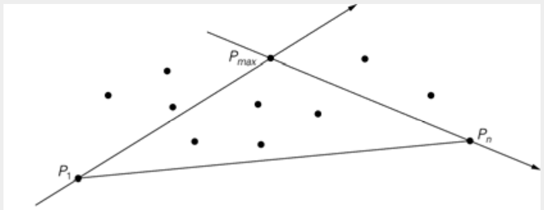

FIGURE 4.9 The idea of quickhull

- What arrangements of points give us worst case behavior?
- Average case is much better. Why?

Ordinary Matrix Multiplication

How many additions and multiplications are needed to compute the product of two 2x2 matrices?

$$\begin{bmatrix} C_{00} & C_{01} \\ C_{10} & C_{11} \end{bmatrix} = \begin{bmatrix} A_{00} & A_{01} \\ A_{10} & A_{11} \end{bmatrix} * \begin{bmatrix} B_{00} & B_{01} \\ B_{10} & B_{11} \end{bmatrix}$$

Strassen's Matrix Multiplication

Strassen observed [1969] that the product of two matrices can be computed as follows:

$$\begin{bmatrix} C_{00} & C_{01} \\ C_{10} & C_{11} \end{bmatrix} = \begin{bmatrix} A_{00} & A_{01} \\ A_{10} & A_{11} \end{bmatrix} * \begin{bmatrix} B_{00} & B_{01} \\ B_{10} & B_{11} \end{bmatrix}$$

$$= \begin{bmatrix} M_1 + M_4 - M_5 + M_7 & M_3 + M_5 \\ M_2 + M_4 & M_1 + M_3 - M_2 + M_6 \end{bmatrix}$$

Values of M_1, M_2, … , M_7 are on the next slide

Formulas for Strassen's Algorithm

$M_1 = (A_{00} + A_{11}) * (B_{00} + B_{11})$

$M_2 = (A_{10} + A_{11}) * B_{00}$

$M_3 = A_{00} * (B_{01} - B_{11})$

$M_4 = A_{11} * (B_{10} - B_{00})$

$M_5 = (A_{00} + A_{01}) * B_{11}$

$M_6 = (A_{10} - A_{00}) * (B_{00} + B_{01})$

$M_7 = (A_{01} - A_{11}) * (B_{10} + B_{11})$

How many additions and multiplications?

The Recursive Algorithm

- We multiply square matrices whose size is a power of 2 (if not, pad with zeroes)
- Break up each matrix into four N/2 x N/2 submatrices.
- Recursively multiply the parts.
- How many additions and multiplications?
  - If we do "normal matrix multiplication" recursively using divide and conquer?
  - If we use Strassen's formulas?

Analysis of Strassen's Algorithm

If N is not a power of 2, matrices can be padded with zeros.

Number of multiplications:

$$M(N) = 7M(N/2) + C, \quad M(1) = 1$$

Solution: $M(N) = \Theta(N^{\log_2 7}) \approx N^{2.807}$
vs. N^3 of brute-force algorithm.

What if we also count the additions?

Algorithms with better asymptotic efficiency are known but they are even more complex.