Divide and Conquer

Ming-Hwa Wang, Ph.D.
COEN 279/AMTH 377 Design and Analysis of Algorithms
Department of Computer Engineering
Santa Clara University

Divide and Conquer

- divide the problem into disjoint subproblems (hopefully split the problem in half) and solve them recursively, and the solution is then formed from the solutions of the subproblems
- binary search, merge sort, quicksort, quickselect

Square and Other Roots using Bisection Method

- The square root of $n \geq 1$ must be at least 1 and at most $n$. Let $l = 1$, $r = n$, and $m = (l + r)/2$. If $n \geq m^2$, then the root $> m$ and repeats with $l = m$. If $n < m^2$, repeats with $r = m$.
- Finding roots of an equation: if $f$ is a continuous function, and $f(l) < 0$ and $f(r) > 0$, there must be a root between $l$ and $r$.
- Use interpolate instead of always testing midpoint.

Towers of Hanoi

- The towers of Hanoi puzzle consists of 3 pegs $A$, $B$, and $C$. Initially peg $A$ has on it $n$ disks, starting with the largest one on the bottom and successively smaller ones on top. The object of the puzzle is to move the disks one at a time from peg to peg, never placing a larger one on top of a smaller one, eventually ending with all disks on peg $B$.
- Move the $n-1$ smallest disks from peg $A$ to peg $C$, exposing the $n^{th}$ smallest disk on peg $A$. Move that disk from $A$ to $B$. Then move the $n-1$ smallest disks from $C$ to $B$.

Tournament Scheduling Problem

- A round robin tournament schedule, for $n = 2^k$ players. Each player must play every other player, and each player must play one match per day for $n-1$ days. The tournament schedule is thus an $n$ row by $n-1$ column table whose entry in row $i$ and column $j$ is the player $i$ must contend with on the $j^{th}$ day.
- Recursively find a schedule for on half of these players and so on. When we get down to two players (the base case), simply pair them up.

Close-Points Problem

- find the closest pair of points of a list of points in a plane
- $N$ points need $N(N-1)/2$ pairs, $O(N^2)$
- Sort all points by $x$ coordinate in $O(N \lg N)$ time, draw an imaginary vertical line that partitions the point set into two halves, $P_L$ and $P_R$. Either the closest points are both in $P_L$, or they are both in $P_R$, or one is in $P_L$ and the other is in $P_R$. Let’s call these distances $d_L$, $d_R$, and $d_C$. Let $\delta = \min(d_L, d_R)$. Only need to compute $d_C$ if $d_C$ improves on $\delta$. If $d_C$ is such a distance, then the two points that define $d_C$ must be within $\delta$ of the dividing line; refer this area as a strip. Only $O(N^{1/2})$ points are in the strip on average. $O(N)$ time to compute these points.

Arithmetic Problems

- Multiplying integers
  - multiply two $N$-digit numbers $X$ and $Y$ requires $\Theta(N^2)$ operations because each digit in $X$ is multiplied by each digit in $Y$.
  - break $X$ and $Y$ into 2 halves, then $XY = X_1Y_110^n + (X_1Y_2 + X_2Y_1)10^{n/2} + X_2Y_2$ needs 4 multiplications. $T(N) = 4T(N/2) + O(N) = O(N^2)$
  - observe that $X_1Y_2 + X_2Y_1 = (X_1 - X_2)(Y_2 - Y_1) + X_1Y_2 + X_2Y_2$. $T(N) = 3T(N/2) + O(N) = O(N^{2.5})$ the exponentiation problem
    $a^n = \begin{cases} a & \text{if } n = 1 \\ (a^{n/2})^2 & \text{if } n \text{ is even} \\ a(a^{n-1}) & \text{otherwise} \end{cases}$
- Cryptography: the art and science of secret communication over insecure channels
  - modular arithmetic
    - $xy \mod z = ((x \mod z)(y \mod z)) \mod z$
    - $x^n \mod z = (x \mod z)^n \mod z$
  - Translate private message $m$ into ciphertext $c$ using enciphering algorithm and a key $k$, which is a shared secret between the sender and the receiver before the communication take place. An eavesdropper who intercepts $c$ without knowing $k$ will be unable to decipher $m$.
- Public-key cryptography, the RSA (Rivest, Shamir and Adleman) cryptographic system: for any two citizens to be able to communicate privately without prior coordination.
- Matrix multiplication
  - multiply 2 matrices $C = AB$, $C_{ij} = \text{dot product of the $i^{th}$ row in } A \text{ with the $j^{th}$ column in } B$. $O(N^3)$
  - divide each matrix into 4 quadrants, performs $8N/2$ by $N/2$ matrix multiplications and $4N/2$ by $N/2$ matrix additions. $T(N) = 8T(N/2) + O(N^3)$

\[
\begin{align*}
C_{1,1} &= A_{1,1}B_{1,1} + A_{1,2}B_{2,1} \\
C_{1,2} &= A_{1,1}B_{1,2} + A_{1,2}B_{2,2} \\
C_{2,1} &= A_{2,1}B_{1,1} + A_{2,2}B_{2,1}
\end{align*}
\]
\[ C_{2,2} = A_{2,1} B_{1,2} + A_{2,2} B_{2,2} \]

- Strassen uses only seven recursive calls. \( T(N) = 7T(N/2) + O(N^2) = O(N^{\log_27}) = O(N^{2.81}) \)
  
  \[ M_1 = (A_{1,2} - A_{2,2})(B_{2,1} + B_{2,2}) \]
  \[ M_2 = (A_{1,1} + A_{2,2})(B_{1,1} + B_{2,2}) \]
  \[ M_3 = (A_{1,1} - A_{2,1})(B_{1,1} + B_{1,2}) \]
  \[ M_4 = (A_{1,1} + A_{1,2})B_{2,2} \]
  \[ M_5 = A_{1,1}(B_{1,2} - B_{2,2}) \]
  \[ M_6 = A_{2,2}(B_{2,1} - B_{1,1}) \]
  \[ M_7 = (A_{2,1} + A_{2,2})B_{1,1} \]
  \[ C_{1,1} = M_1 + M_2 - M_4 + M_6 \]
  \[ C_{1,2} = M_4 + M_5 \]
  \[ C_{2,1} = M_5 + M_7 \]
  \[ C_{2,2} = M_2 - M_3 + M_5 - M_7 \]

- Strassen's algorithm has big coefficient for its big-Oh. Not good for parallelization and sparse matrices. However, it is an important theoretical milestone "Even though a problem seems to have an intrinsic complexity, nothing is certain until proven."