Divide & Conquer

1- Divide problem into n subproblems

2- Conquer: i.e. solve the subproblems recursively, or if trivial solve the problem itself

3- Combine the solution to the subproblems

```
MERGE-SORT(A, p, r)
  if p<r then
     q = ⌊(p+r)/2⌋
     MERGE-SORT(A, p, q)
     MERGE-SORT(A, q+1, r)
     MERGE(A, p, q, r)
  endif
```

Analyzing Merge-sort

Divide - Takes ______

Conquer - If the original problem takes $T(n)$ time, the two subproblems take ______

Combine - Takes ______

in general, our recurrence equation for a D&C solution will look like:

$$T(n) = \begin{cases} \Theta(1) & \text{if } n \leq c \\ aT(n/b) + D(n) + C(n) \end{cases}$$

Master Method

It is a cookbook method for solving recurrences of the form

$$T(n) = aT(n/b) + f(n)$$

- where $a \geq 1$, $b \geq 1$ are constants

- $f(n)$ is an asymptotically positive function.

Master Theorem

- Given the above defenition of the recurrence relation, $T(n)$ can be bounded asymptotically as follows:

1- If $f(n) = O(n^{\log_b a - \epsilon})$ for some $\epsilon > 0$,

then $T(n) = \Theta(n^{\log_b a})$

2- If $f(n) = \Theta(n^{\log_b a})$ then

$$T(n) = \Theta(n^{\log_b a} \lg n)$$

3- If $f(n) = \Omega(n^{\log_b a + \varepsilon})$ for some constant $\varepsilon > 0$, and if $a f(n/b) \leq c f(n)$ for some constant $c < 1$ and all sufficiently large n, then

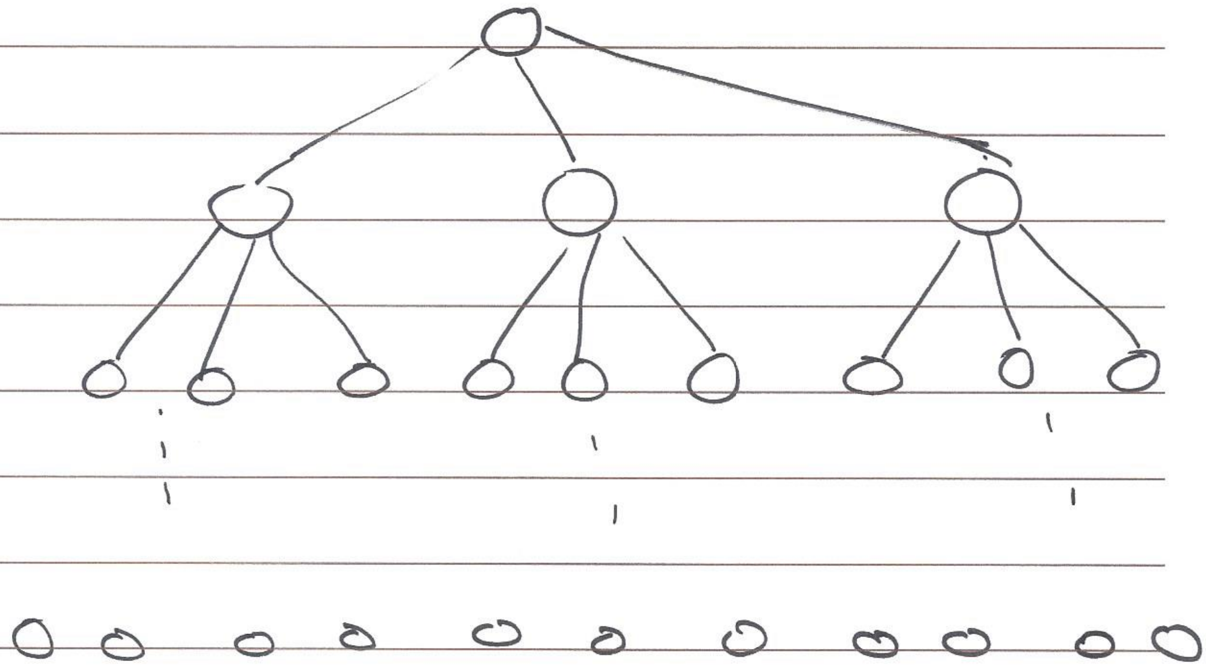
$$T(n) = \Theta(f(n))$$

in general if $f(n) = \Theta(n^{\log_b a} \lg^k n)$ where $k \geq 0$

Then

$$T(n) = \Theta(n^{\log_b a} \lg^{k+1} n)$$

Intuition Behind The Master Method



Case 1: Complexity driven by the no. of leaf nodes in the recursion tree.

Case 3: Complexity driven by the cost of the root node in the recursion tree

Case 2: Cost of operations are the same at every level of the recursion tree.

Stock Market Problem

Dense Matrix Multiplications

Compute 7 $n/2 \times n/2$ intermediate matrices

$$P = (A_{11} + A_{22})(B_{11} + B_{22})$$
$$Q = (A_{21} + A_{22})\, B_{11}$$
$$R = A_{11}(B_{12} - B_{22})$$
$$S = A_{22}(B_{21} - B_{11})$$
$$T = (A_{11} + A_{12})\, B_{22}$$
$$U = (A_{21} - A_{11})(B_{11} + B_{12})$$
$$V = (A_{12} - A_{22})(B_{21} + B_{22})$$

Finding Min & Max in an unsorted array

Closest pair of points problem (2D)

Implementation

closest-pair (p)

Construct P_x : list of points sorted by x-coord.

" P_y: list of points sorted by y coord.

(p_0, p_1) = Closest-pair-Rec(P_x, P_y)

Closest-pair-Rec (P_x, P_y)

if $|P| \leq 3$ then

solve it directly

else

Construct Q_x ... left half of P_x

" Q_y ... list of points in Q_x sorted by y coord.

Construct R_x ... right half of P_x

" R_y ... list of points in R_x sorted by y coord.

$(q_0, q_1) = $ closest_pair_Rec (Q_x, Q_y)

$(r_0, r_1) = $ closest_pair_Rec (R_x, R_y)

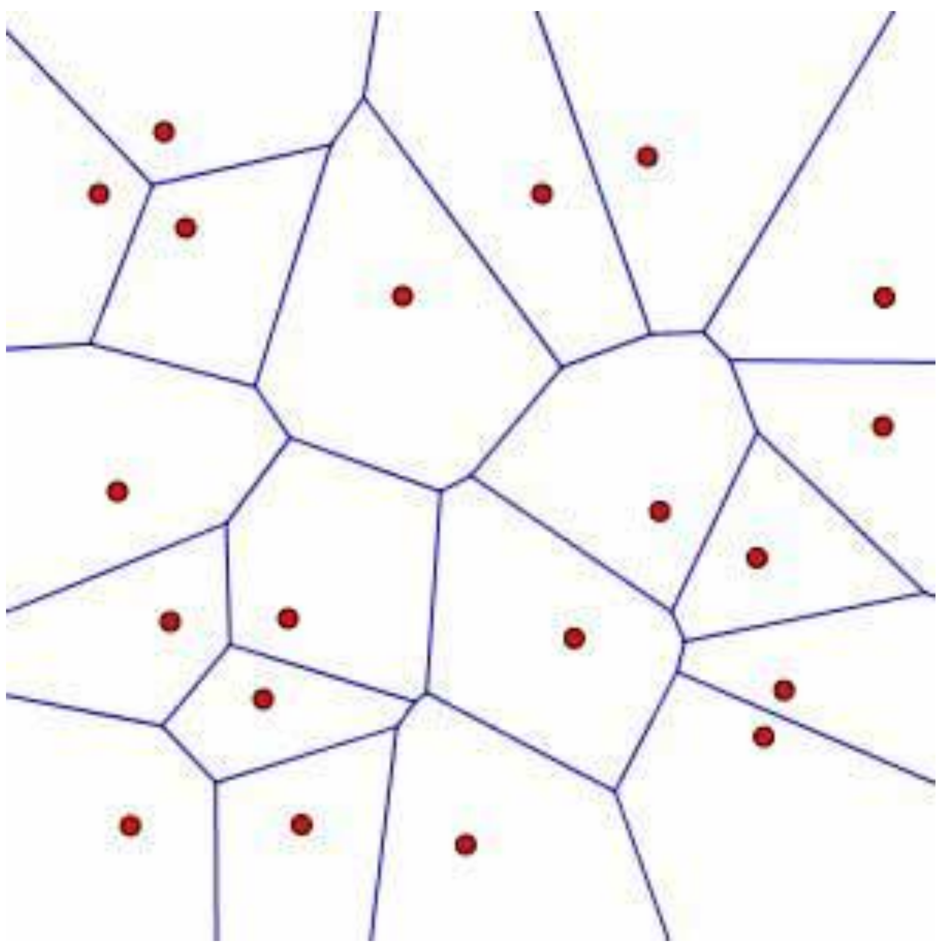
$\delta = \min(d(q_0, q_1), d(r_0, r_1))$

S = set of points in P within distance of δ from L.

Construct S_y -- set of points in S sorted by y coord.

for each point $s \in S_y$, compute distance from s to each of next 11 points in S_y.
let (s, s') be pair with min. distance
if $d(s, s') < \delta$ then
    Return (s, s')
else if $d(q_0, q_1) < d(r_0, r_1)$ then
    Return (q_0, q_1)
else
    Return (r_0, r_1)
endif

All Nearest Neighbors Problem



Discussion 5

1. Suppose we have two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, along with T_1 which is a MST of G_1 and T_2 which is a MST of G_2. Now consider a new graph $G = (V, E)$ such that $V = V_1 \cup V_2$ and $E = E_1 \cup E_2 \cup E_3$ where E_3 is a new set of edges that all cross the cut (V_1, V_2).

Consider the following algorithm, which is intended to find a MST of G.

Maybe-MST(T_1, T_2, E_3)

e_{min} = a minimum weight edge in E_3

$T = T_1 \cup T_2 \cup \{ e_{min}\}$

return T

Does this algorithm correctly find a MST of G? Either prove it does or prove it does not.

2. Solve the following recurrences using the Master Method:

a. $A(n) = 3 A(n/3) + 15$
b. $B(n) = 4 B(n/2) + n^3$
c. $C(n) = 4 C(n/2) + n^2$
d. $D(n) = 4 D(n/2) + n$

3. There are 2 sorted arrays A and B of size n each. Design a D&C algorithm to find the median of the array obtained after merging the above 2 arrays (i.e. array of length 2n). Discuss its runtime complexity.

4. A tromino is a figure composed of three 1x1 squares in the shape of an L.
Given a 2^nx2^n checkerboard with 1 missing square, tile it with trominoes.
Design a D&C algorithm and discuss its runtime complexity.