7 Divide and Conquer

Tuesday, October 29, 2024 11:23 AM

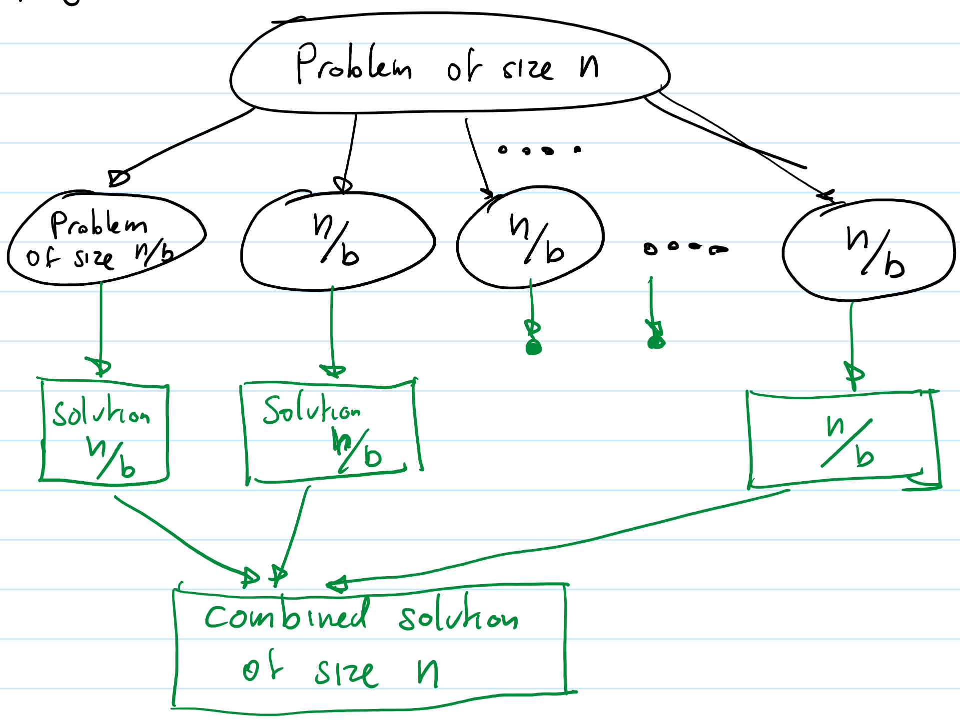
The basic strategy

1.- Divide the problem into several subproblems
- ideally of the same size
- usually 2

2.- the subproblems are solved
- usually recursivelly.

3.- the solutions to the subproblems are combined to get a solution to the original problem.

Diagram:



usually $b = 2$

a subproblems need to be solved
$a \leq b$

Note: Some of the most important are of this kind

Example: Sum of an Array.

$$\text{sum}(A[0..n-1])$$
$$= \text{sum}(A[0..\lfloor \tfrac{n}{2} \rfloor -1]) + \text{sum}(A[\lfloor \tfrac{n}{2} \rfloor ... n-1])$$

Base-Case

$sum([x]) = X$ size $A = 1$

- Same number of additions required.

Divide-and-Conquer is good, but not always leads to a better solution.

Analysis: basic-operation: recombination.

$$A(n) = 2 \cdot A\left(\frac{n}{2}\right) + 1$$
$$A(1) = 0$$

recurrance.

- Analysis Framework for Divide and conquer algorithms.

Divide and conquer will lead to reccurances:

- input of size n
- divided into b parts of equal size.
- a parts that need to be solved
- the cost of recombination being $f(n)$

the General formula is

$$T(n) = a \cdot T\left(\frac{n}{b}\right) + f(n)$$

"General divide and conquer reccurance"

Let n be a power of b $n = b^k$ $k = \log_b n$

$$T(b^k) = a \cdot T(b^{k-1}) + f(b^k)$$

$$T(b^{k-1}) = a \cdot T(b^{k-2}) + f(b^{k-1})$$

$$= a\left[a \cdot T(b^{k-2}) + f(b^{k-1})\right] + f(b^k)$$

$$= a^2 \cdot T(b^{k-2}) + a \cdot f(b^{k-1}) + f(b^k)$$

$$= a^2 \cdot \left[a \cdot T(b^{k-3}) + f(b^{k-2})\right] + a \cdot f(b^{k-1}) + f(b^k)$$

$$= a^3 \cdot T(b^{k-3}) + a^2 \cdot f(b^{k-2}) + a^1 \cdot f(b^{k-1}) + f(b^k)$$

after i substitutions

$$= a^i \cdot T(b^{k-i}) + a^{i-1} \cdot f(b^{k-i+1}) + a^{i-2} \cdot f(b^{k-i+2}) + \dots + a^1 \cdot f(b^{k-1}) + f(b^k)$$

Let $i = k$

$$= a^k \cdot T(1) + a^{k-1} \cdot f(b^1) + a^{k-2} \cdot f(b^2) + a^{k-3} \cdot f(b^3) + \dots + a^1 \cdot f(b^{k-1}) + a^0 f(b^k)$$

$$= a^k \cdot T(1) + \frac{a^k}{a} \cdot f(b^1) + \frac{a^k}{a^2} \cdot f(b^2) + \frac{a^k}{a^3} \cdot f(b^3) + \dots + \frac{a^k}{a^{k-1}} \cdot f(b^{k-1}) + \frac{a^k}{a^k} \cdot f(b^k)$$

$$= a^k \left[T(1) + \sum_{j=1}^{k} \frac{f(b^j)}{a^j} \right] \qquad a^k = a^{\log_b n} = n^{\log_b a}$$

$$T(n) = n^{\log_b a} \left[T(1) + \sum_{j=1}^{\log_b n} f(b^j) / a^j \right].$$

what impacts the order of growth of $T(n)$:

- the value of b
- the value of a
- the order of growth of $f(n)$

The Master theorem:

Given a recurrance:

$$T(n) = a \cdot T\left(\frac{n}{b}\right) + f(n)$$

If $f(n) \in \Theta(n^d)$ where $d \geq 0$ then

$$T(n) \in \begin{cases} \Theta(n^d) & \text{if } a < b^d \quad (1) \\ \Theta(n^d \log n) & \text{if } a = b^d \quad (2) \\ \Theta(n^{\log_b a}) & \text{if } a > b^d \quad (3) \end{cases}$$

apply:

$$A(n) = 2 \cdot A\left(\frac{n}{2}\right) + 1 \qquad a = 2 \quad d = 0$$
$$A(1) = 0 \qquad b = 2$$

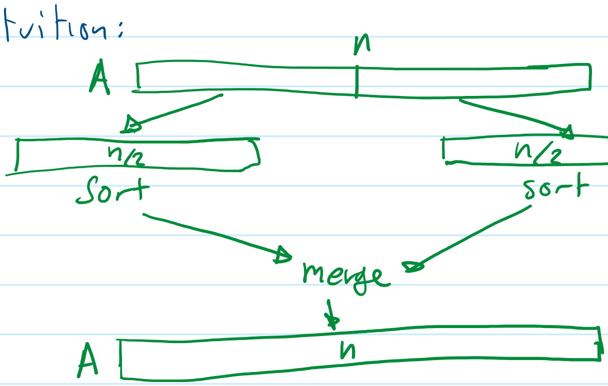
$$f(n) = 1 \in \Theta(1) = \Theta(n^0)$$

$a > b^d$

$$A(n) \in \Theta\left(n^{\log_b a}\right) = \Theta\left(n^{\log_2 2}\right) = \Theta(n)$$

- Merge Sort

Intuition:



Pseudocode MergeSort($A[0 \ldots n-1]$)

if $n > 1$

copy $A[0 \ldots \lfloor n/2 \rfloor - 1]$ to $B[0 \ldots \lfloor n/2 \rfloor - 1]$

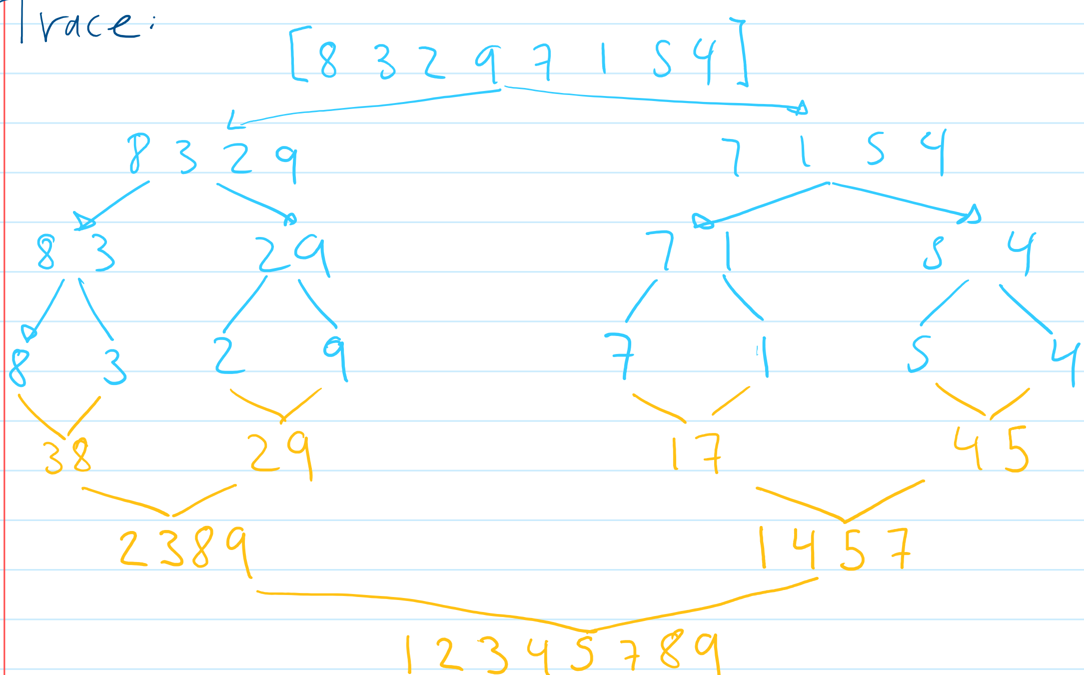
Copy $A[\lfloor n/2 \rfloor \ldots n-1]$ to $C[0 \ldots \lceil n/2 \rceil - 1]$

MergeSort(B)

MergeSort(C)

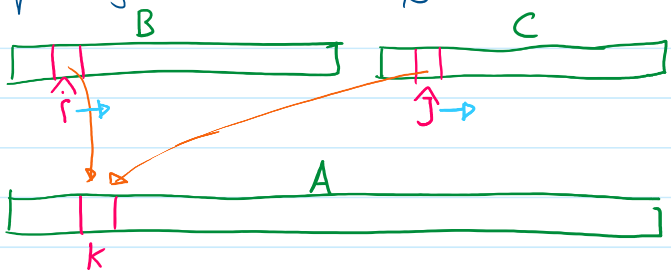
Merge (B, C, A)

Trace:



Merge:

- Comparing two sorted arrays



FUNCTION Merge($B[0..p-1], C[0...q-1], A[0...p+q-1]$)

```
  i←0; j←0; k←0;

  WHILE i<p and j<q DO
      IF B[i] < C[j] then
          A[k] ← B[i]
          i ← i+1
      ELSE
          A[k] ← C[j]
          j ← j+1
      k ← k+1
  IF i=p THEN
      copy C[j..q-1] to A[k.. p+q-1]
  ELSE
      copy B[i..p-1] to A[k.. p+q-1]
```

MergeSort is cool 

- ⊕ is "Stable": elements of same value retain their position relative to each other.
- ⊖ needs extra storage.

Analysis.

basic operation: $<$ comparison.

$$C(n) = 2 \cdot C\left(\frac{n}{2}\right) + T_{merge}(n)$$

$$T_{merge}(n) = n-1$$

$$C(n) = 2 \cdot C\left(\frac{n}{2}\right) + n-1$$

Apply Masters theorem:

$a = 2$ $\quad$ $b = 2$ $\quad$ $f \in \Theta(n^1)$ $\quad$ $d = 1$

Apply Masters Theorem:

$a = 2$ $\quad$ $n-1 \in \Theta(n^1)$ $\quad$ $d = 1$

$b = 2$

then

$a = b^d \qquad 2 = 2^1$

then.

$$C(n) \in \Theta(n^d \cdot \log n) = n \cdot \log n$$

Merge Sort is better!!

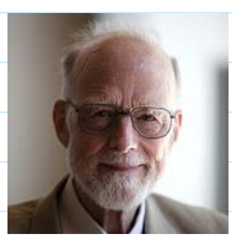
How much better.

n	bubble sort	merge sort
100	10,000	665
10,000	100,000,000	132,880

(n^2 algorithms are deceptive:
useful for small inputs.
but they eventually explode)

- Quick Sort

'60 C.A.R Hoare
"Tony"



- Russian-to-English translator.

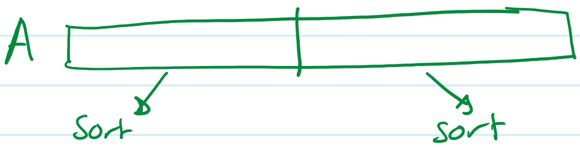
Russian Text.

Sort russian text

Rus to Eng

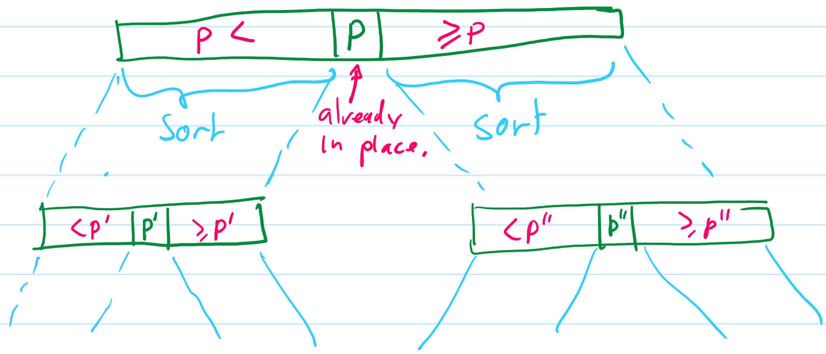
Sorted.

A



But: How can we avoid merging?

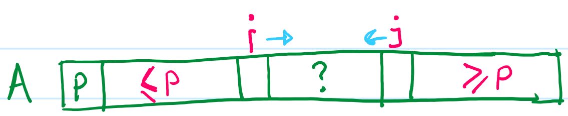
The solution is partition:



Algorithm QuickSort(A[l..r])
  If l < r then
    s ← partition(A[l..r])
    QuickSort(A[l..s-1])
    Quicksort(A[s+1..r])

= A new way of partitioning (Hoare Partition)

Intuition:



- if $A[i] \leq p$ then $i \leftarrow i+1$
- if $A[j] \geq p$ then $j \leftarrow j-1$

eventually: $A[i] > p$ and $A[j] < p$
  then swap(A[i], A[j])

When do we stop? $i = j$ or $j < i$ ∴ $j \leq i$

FUNCTION HoarePartition(A[l..r])
  $p \leftarrow A[l]$
  $i \leftarrow l$; $j \leftarrow r+1$

Trace

P 0 1 2 3 4 5 6 7

```
p ← A[l]
i ← l;  j ← r+1
REPEAT
    REPEAT i ← i+1 UNTIL A[i] ≥ p *
    REPEAT j ← j-1 UNTIL A[j] ≤ p
    swap(A[i], A[j])
UNTIL j ≤ i
Swap(A[i], A[j])
Swap(A[l], A[j])
RETURN j
```

★ NOTE ★ this could go off the end of the array. it needs an extra condition.

Trace

p = 6

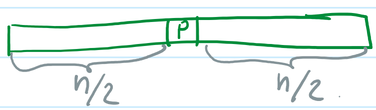
0	1	2	3	4	5	6	7
6	10	7	9	3	4	8	1
6	1	7	9	3	4	8	10
6	1	4	9	3	7	8	10
6	1	4	3	9	7	8	10
6	1	4	9	3	7	8	10
6	1	4	3	9	7	8	10
3	1	4	$\boxed{6}$	9	7	8	10

Analysis.

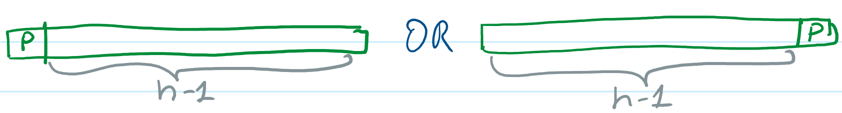
basic operation: comparison.

Partitionin Cases:

Best:



Worst:



- Hoare Partitioning takes n or $n+1$ comparisons.
- Best: $C_{best}(n) = 2 \cdot \underbrace{C_{best}(n/2)}_{\text{Quicksort each subarray}} + \underbrace{n}_{\text{partitioning}}$

Apply Masters Theorem:

$a = 2$ $\quad f(n) = n \in \Theta(n^1)$

$b = 2$ $\quad d = 1$

Case: $a = b^d$

so: $C_{best}(n) \in \Theta(n \cdot \log n)$

- Worst:

$$C_{worst}(n) = (n+1) + n + (n-1) + (n-2) + (n-3) \dots + 3$$
$$= \frac{(n+1)(n+2)}{2} - 3 \in \Theta(n^2)$$

- Avg Case:
  - s: position of the pivot $0 \leq s \leq n-1$
  - suppose each position of the pivot is equally likely; then the pivot can land in a position with probability $1/n$

$$C_{avg}(n) = \frac{1}{n} \cdot \underbrace{\sum_{s=0}^{n-1}}_{\text{positions for pivot}} \left[\underbrace{(n+1)}_{\text{partition}} + \underbrace{C_{avg}(s)}_{\text{Quicksort left subarray}} + \underbrace{C_{avg}(n-1-s)}_{\text{Quicksort right subarray.}} \right]$$

Magic Happens ::

$$C_{avg}(n) \approx 2n \ln n \approx 1.39 \cdot n \cdot \log_2 n$$

On average: Quicksort makes 39% more comparisons than in the best case.

Quicksort

⊕ in-place: no need for extra memory

⊖ not-stable:

Improvements:

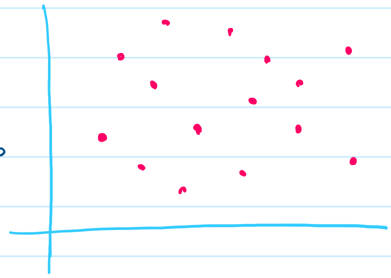
- Improve selection of pivot
  - look at 3 elements, keep the middle one.
  - pick at random.
- Switch to insertion sort for small subarrays $n \leq 5..15$
- 3-way partitioning (using 2 pivots)

It's possible to achieve 20% to 30% speedups.

- CLOSEST-PAIR

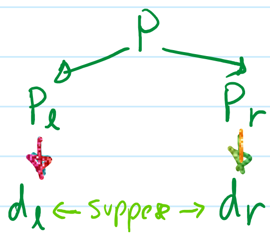
Problem :

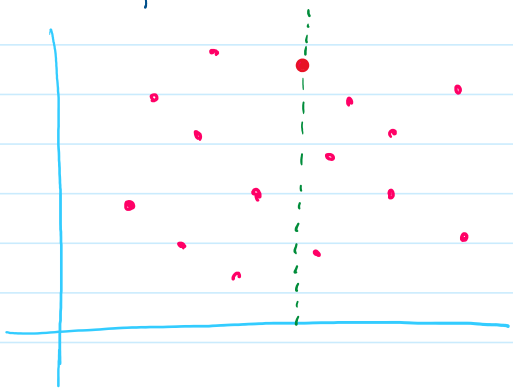
find the cosest two points.



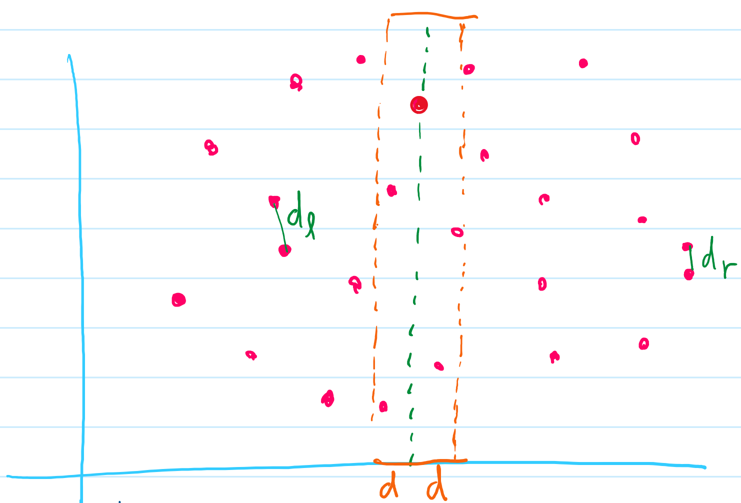
Let us Divide and Conquer.

1) split points by median





2) How to combine sub-solutions d_l d_r to build a complete solution.



Let d is the smallest of d_r d_l.

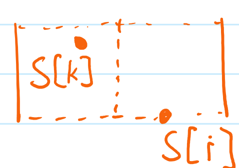
- the only way for a closer pair to exists is if it's extremes are in each side of the partition
- We can narrow even further:

at most 8 points to pair p with.

FUNCTION EffClosestPair (P: collection of points sorted by x value
Q: collection of points sorted by y value)

If $|P| \le 3$ then
    Solve by Brute force.
else
    copy first $\lceil n/2 \rceil$ points of P to P_ℓ
    copy same $\lceil n/2 \rceil$ points from Q to Q_ℓ
    copy remaining $\lfloor n/2 \rfloor$ points of P to P_r
    copy remaining $\lfloor n/2 \rfloor$ points of Q to Q_r

    $d_\ell \leftarrow$ EffClosestPair (P_ℓ, Q_ℓ)
    $d_r \leftarrow$ EffClosestPair (P_r, Q_r)
    # merge solutions
    $d \leftarrow \min(d_\ell, d_r)$
    $m \leftarrow P[\lceil n/2 \rceil - 1].x$
    Copy from Q all points which $|x - m| < d$ into $S[0 .. n_s - 1]$
    $d_{min} \leftarrow d^2$
    FOR $i \leftarrow 0$ to $n_s - 1$ DO
        $k \leftarrow i + 1$
        WHILE $k < n_s$ and $(S[k].y - S[i].y)^2 < d_{min}$ DO
            $d' = (S[k].x - S[i].x)^2 + (S[k].y - S[i].y)^2$
            IF $d' < d_{min}$ THEN
                $d_{min} \leftarrow d'$
            $k \leftarrow k + 1$

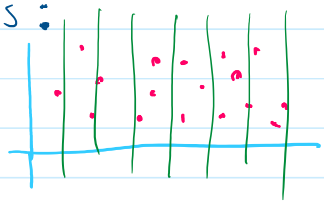


RETURN $\sqrt{d_{min}}$

S[i]

RETURN $\sqrt{d_{min}}$

Analysis:



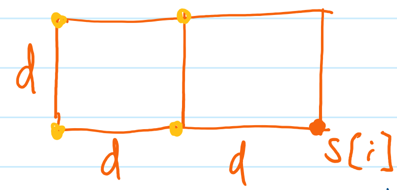
$$T(n) = 2 \cdot T\left(\frac{n}{2}\right) + f(n)$$

cost of searching the middle strip

- $f(n)$ cost of searching the middle strip

basic-operation =

$$C(n_s) = \sum_{i=0}^{n_s-1} 8 = 8 \cdot n_s \in \Theta(n)$$



(at most 8 points)

- $f(n) \in \Theta(n)$

Apply Master theorem.

$a = 2$

$b = 2$

$f(n) \in \Theta(n^d)$

$d = 1$

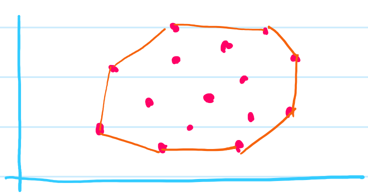
then:

$$T(n) \in \Theta(n \cdot \log n)$$

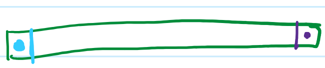
Presorting is not an issue:

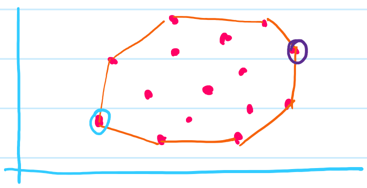
- A good sort is $\Theta(n \cdot \log n)$

- CONVEX HULL

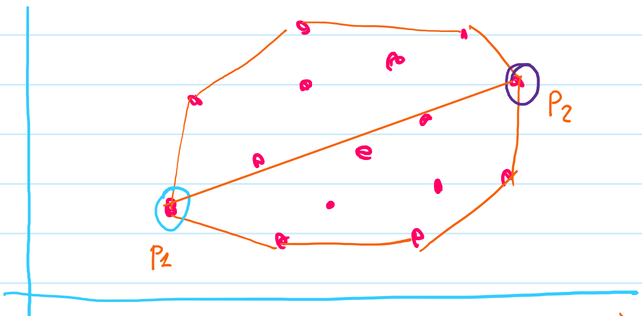


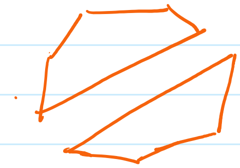
Idea: Sort the points by x value.





We gain 2 points: the extremes are in the convex hull.



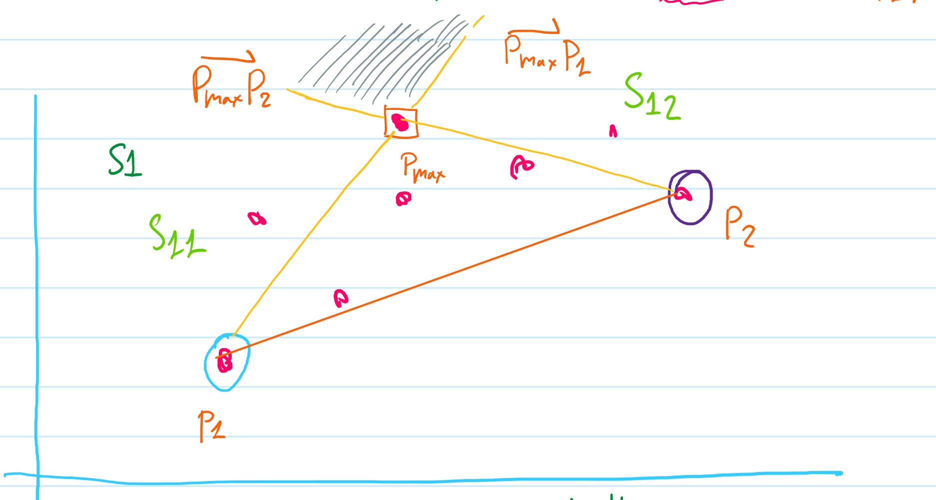


S_1 = points to the left of $\overrightarrow{p_1 p_2}$

S_2 = points to the right of $\overrightarrow{p_1 p_2}$

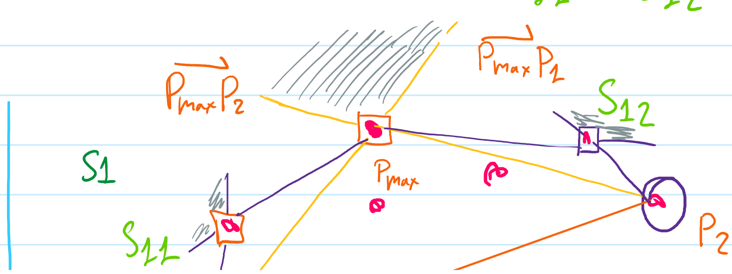
Consider S_1

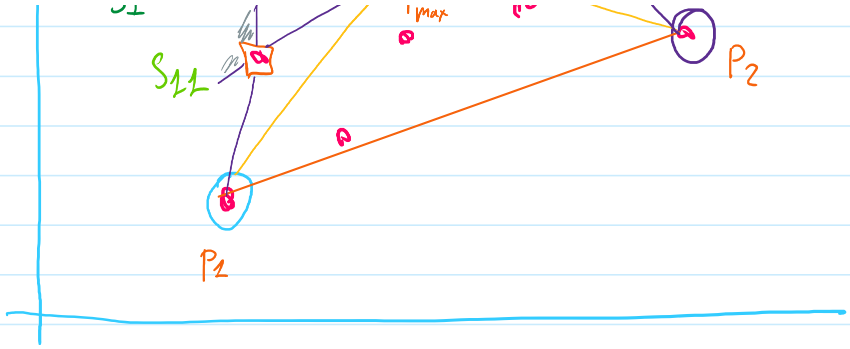
Identify P_{max} : the point further away from $\overrightarrow{P_1 P_2}$



- P_{max} is in the convex hull
- points in the triangle $\Delta P_1 P_2 P_{max}$ are not in the hull
- there can be no point to the left of both $\overrightarrow{P_1 P_{max}}$ and $\overrightarrow{P_{max} P_2}$

Repeat the same process on S_{11} and S_{12}





Base Case: $|S| = 0$: the previous reference line is the hull
or $|S| = 1$: lines to the remaining point are in the hull.

- How to compare 3 points? :

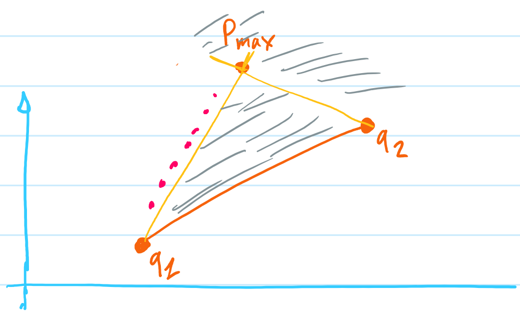
Given $q_1\ q_2\ q_3$ points, we can obtain information on the triangle $\Delta q_1 q_2 q_3$ by looking at the determinant of the matrix

$$\begin{bmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{bmatrix} \overset{\text{det}}{=} x_1 \cdot y_2 + x_3 \cdot y_1 + x_2 \cdot y_3 - x_3 \cdot y_2 - x_2 \cdot y_1 - x_1 \cdot y_3$$

- The determinant is twice the area of the triangle.
- If q_3 is to the left of $\overrightarrow{q_1 q_2}$ the area will be positive otherwise it will be negative.

Analysis:

consider the worst:



$$T_{\text{worst}}(n) = n + (n-1) + (n-2) + (n-3) + \ldots + 1 = \frac{n(n+1)}{2}$$
$$\in \Theta(n^2)$$

$$T_{\text{avg}}(n) \in \Theta(n \log n)$$

with a normal distribution of the points: $T(n) \in (n)$

- MULTIPLICATION

e.g.

```
   6 6 5 3
   1 7 8 7 5
 x 3 1 0 2 7
 -----------
   1 3 5 1 2 5
 3 5 7 5 0 0
```

Analysis: basic operation

Single digit multiplication.

$M(n) = n^2$

```
        a1        a2
  a   1 2 3 3 | 5 7 2 7
x b   0 0 9 2 | 3 5 7 8
        b1        b2
```

$a_1 * b_1$
$a_1 * b_2$
$a_2 * b_1$
$a_2 * b_2$

"Anatoly Karatsuba"

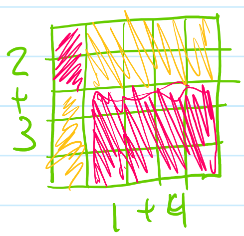
$$a * b = (a_1 \cdot b_1)\cdot 10^8 + \left((a_2 \cdot b_1) + (a_1 \cdot b_2)\right)\cdot 10^4 + (a_2 \cdot b_2)$$

Let us reformulate to save on multiplications.

e.g.
```
   23
 * 14
 ----
```

$$23 \cdot 14 = (2\cdot 1)\cdot 10^2 + \left((4\cdot 2) + (1\cdot 3)\right)\cdot 10^1 + (3\cdot 4)\cdot 10^0$$

Let's change Perspective :



$$(4\cdot 2 + 1\cdot 3) = 5\cdot 5 - (2\cdot 1) - (3\cdot 4)$$

$$23 \cdot 14 = (2\cdot 1)\cdot 10^2 + \left((5\cdot 5) - (2\cdot 1) - (3\cdot 4)\right)\cdot 10^1 + (3\cdot 4)\cdot 10^0$$

Repeated.

In general for 2 digit numbers

$a = a_1 a_0 \qquad b = b_1 b_0$

$$a * b = C_2 \cdot 10^2 + C_1 \cdot 10^1 \cdot C_0$$

$C_2 = a_1 * b_1$

$C_0 = a_0 * b_0$

$C_1 = (a_1 + a_0) * (b_1 + b_0) - C_2 - C_0$

Lets scale it to multiple digit numbers. of size n digits

$a = a_1 a_0 \qquad a_1, a_0$ are multidigit numbers. e.g.

$a = a_1 \cdot 10^{n/2} + a_0$

3127

$3\cdot 10^3 + 1\cdot 10^2 + 2\cdot 10^1 + 7\cdot 10^0$

$a = a_1 a_0$ a_1, a_0 are multidigit numbers. e.g.

$a = a_1 \cdot 10^{n/2} + a_0$

3127

$3 \cdot 10^3 + 1 \cdot 10^2 + 2 \cdot 10^1 + 7 \cdot 10^0$

$b = b_1 b_0$ b_1, b_0 are multidigit numbers

$31 \cdot 10^2 + 27 \cdot 10^0$

$b = b_1 \cdot 10^{n/2} + b_0$

$$a * b = c_2 \cdot 10^n + c_1 \cdot 10^{n/2} + c_0$$

where: $c_2 = a_1 * b_1$

$c_0 = a_0 * b_0$

$c_1 = (a_1 + a_0) * (b_1 + b_0) - (c_2 + c_0)$

Analysis:

$$M(n) = 3 \cdot M(n/2) \qquad M(1) = 1$$

Let $n = 2^k$ $k = \log_2 n$

$$M(2^k) = 3 \cdot M(2^{k-1})$$ by backwards substitution

$$= 3 \cdot (3 \cdot M(2^{k-2}))$$

$$= 3^2 \cdot M(2^{k-2})$$

$$= 3^2 \cdot (3 \cdot M(2^{k-3}))$$

$$= 3^3 \cdot M(2^{k-3})$$ after i substitutions

$$= 3^i \cdot M(2^{k-i})$$ Let $i = k$

$$= 3^k \cdot M(2^{k-k})$$

$$= 3^k \cdot M(1)$$

$$= 3^k$$

$$M(n) = 3^{\log_2 n} = n^{\log_2 3} \approx n^{1.585}$$

But what about additions/subtractions?

$$A(n) = 3 \cdot A(n/2) + 5n$$

Apply Masters theorem.

$a = 3$ $f(n) = 5 \cdot n \in \Theta(n^1)$

$b = 2$ $d = 1$

$a \Leftrightarrow b^d$? $a > b^d$

Therefore:

$$A(n) \in \Theta\left(n^{\log_b a}\right) = \Theta\left(n^{\log_2 3}\right) = \Theta\left(n^{1.585}\right)$$

Why?

Multiplying large numbers is important:

- MATRIX MULTIPLICATION

"V. Strassen"

Original:

$$\begin{bmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \end{bmatrix} * \begin{bmatrix} b_{00} & b_{01} \\ b_{10} & b_{11} \end{bmatrix} = \begin{bmatrix} a_{00} \cdot b_{00} + a_{01} \cdot b_{10} & a_{00} \cdot b_{01} + a_{01} \cdot b_{11} \\ a_{10} \cdot b_{00} + a_{11} \cdot b_{10} & a_{10} \cdot b_{01} + a_{11} \cdot b_{11} \end{bmatrix}$$

8 multiplications, $M(n) = n^3$

Strassen:

$$\begin{bmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \end{bmatrix} * \begin{bmatrix} b_{00} & b_{01} \\ b_{10} & b_{11} \end{bmatrix} = \begin{bmatrix} m_1 + m_4 - m_5 + m_7 & m_3 + m_5 \\ m_2 + m_4 & m_1 + m_3 - m_2 + m_6 \end{bmatrix}$$

where:

$$\begin{aligned} m_1 &= (a_{00} + a_{11}) * (b_{00} + b_{11}) \\ m_2 &= (a_{10} + a_{11}) * b_{00} \\ m_3 &= a_{00} * (b_{01} - b_{11}) \\ m_4 &= a_{11} * (b_{10} - b_{00}) \\ m_5 &= (a_{00} + a_{01}) * b_{11} \\ m_6 &= (a_{10} - a_{00}) * (b_{00} + b_{01}) \\ m_7 &= (a_{01} - a_{11}) * (b_{10} + b_{11}) \end{aligned}$$

7 multiplications

$$\begin{aligned} m_3 + m_5 &= a_{00} * (b_{01} - b_{11}) + (a_{00} + a_{01}) * b_{11} \\ &= a_{00} * b_{01} \cancel{- a_{00} \cdot b_{11}} + \cancel{a_{00} \cdot b_{11}} + a_{01} * b_{11} \\ &= a_{00} * b_{01} + a_{01} * b_{11} \quad ✓ \end{aligned}$$

Apply to large matrices.

Square matrices of size a power of 2

$$\begin{array}{ccccc} A & & B & & C \\ \left[\begin{array}{c|c} A_{00} & A_{01} \\ \hline \end{array}\right] & * & \left[\begin{array}{c|c} B_{00} & B_{01} \\ \hline \end{array}\right] & = & \left[\begin{array}{c|c} C_{00} & C_{01} \\ \hline \end{array}\right] \end{array}$$

$$\left[\begin{array}{c|c} A_{00} & A_{01} \\ \hline A_{10} & A_{11} \end{array}\right] \cdot \left[\begin{array}{c|c} B_{00} & B_{01} \\ \hline B_{10} & B_{11} \end{array}\right] = \left[\begin{array}{c|c} C_{00} & C_{01} \\ \hline C_{10} & C_{11} \end{array}\right]$$

Obtain C from recursive appicatiors of strassen's algorithm.

Analysis:

- $M(n) = 7 \cdot M(n/2)$ • $M(1) = 1$
- $n = 2^k$ • $k = \log_2 n$

Solve the recurrance

$$M(2^k) = 7 \cdot M(2^{k-1})$$ appy backward substitution

$$= 7 \cdot (7 \cdot M(2^{k-2})) = 7^2 \cdot M(2^{k-2})$$

$$= 7 \cdot (7^2 \cdot M(2^{k-3})) = 7^3 \cdot M(2^{k-3})$$

after i substitutions

$$= 7^i \cdot M(2^{k-i})$$ let $i = k$

$$= 7^k \cdot M(2^{k-k})$$

$$= 7^k$$

$$M(n) = 7^{\log_2 n} = n^{\log_2 7} \approx \underline{n}^{2.807}$$

Lets count Additions & substractions

$$A(n) = 7 \cdot A(n/2) + \underbrace{18(n/2)^2}_{\text{adding two matrices of size } n/2}$$

Apply Masters Theorem

$a = 7$ $\quad f(n) = 18 \cdot (n/2)^2 \in \Theta(n^2)$

$b = 2$ $\quad d = 2$

$a \Leftrightarrow b^d$? $\quad a > b^d$

therefore:

$$A(n) \in \Theta(n^{\log_b a}) = \Theta(n^{\log_2 7}) = \Theta(n^{2.807})$$

there are better one's !!!

Timeline of matrix multiplication exponent

Year	Bound on omega	Authors
1969	2.8074	Strassen[1]
1978	2.796	Pan[10]
1979	2.780	Bini, Capovani [it], Romani[11]
1981	2.522	Schönhage[12]
1981	2.517	Romani[13]
1981	2.496	Coppersmith, Winograd[14]
1986	2.479	Strassen[15]
1990	2.3755	Coppersmith, Winograd[16]
2010	2.3737	Stothers[17]
2012	2.3729	Williams[18][19]
2014	2.3728639	Le Gall[20]
2020	2.3728596	Alman, Williams[21][22]
2022	2.371866	Duan, Wu, Zhou[23]
2024	2.371552	Williams, Xu, Xu, and Zhou[2]
2024	2.371339	Alman, Duan, Williams, Xu, Xu, and Zhou[24]

Screen clipping taken: 11/14/2024 12:08 PM

EOF