Computational Geometry for Non-Geometers:
Recent Developments on Some Classical Problems

Timothy Chan
School of CS
U of Waterloo
4 Toy Problems

(exercises in divide-&-conquer/data structures...)
Problem 1: Inversion Counting

• Given permutation $\pi$ of $\{1, \ldots, n\}$, count # of pairs $(i, j)$ with $i < j$ & $\pi(i) > \pi(j)$

2 1 4 5 8 3 7 6 10 9

• Exercise: $O(n \log n)$ time
Problem 2: Maxima

- Given point set $S$ in 2D,
  - $p=(x,y)$ dominates $q=(x',y')$ if $x > x'$ & $y > y'$
Problem 2: Maxima

• Given point set S in 2D,
  - \( p=(x,y) \) dominates \( q=(x',y') \) if \( x > x' \) & \( y > y' \)
  - \( q \) in \( S \) is maximal if no pt in \( S \) dominates \( q \)
  - find all maximal points in \( S \)

• Exercise: \( O(n \log n) \) time in 2D

[History: Kung,Luccio,Preparata'75, Gabow,Bentley,Tarjan'84]
Problem 3: Orthogonal Segment Intersection

• Given $n$ horizontal/vertical line segments in 2D, report all intersections

• Exercise: $O(n \log n + k)$ time ($k =$ output size)

[History: Bentley, Ottmann'79, Overmars'87,...]
Problem 4: Rectangle Enclosure

- Given $n$ axis-aligned rectangles in 2D, report all pairs $(r,s)$ where $r$ encloses $s$

- Exercise: $O(n \text{ polylog } n + k)$ time ($k = \text{output size}$)

[History: Bentley, Wood'80, Vaishnavi, Wood'80, Lee, Preparata'82, Gupta, Janardan, Smid, DasGupta'95]
Common Thread: Dominance Searching

• All 4 problems reduce to 2D/3D/4D (red-blue) dominance counting/existence/reporting…

[which in turn are examples of (offline) orthogonal range searching]
Problem 1: Inversion Counting

- Given permutation $\pi$ of $\{1,...,n\}$, count # of pairs $(i,j)$ with $i < j$ & $\pi(i) > \pi(j)$

- Exercise: $O(n \log n)$ time

- Diagram: (2 1 4 5 8 3 7 6 10 9) point $(j, -\pi(j))$ dominates point $(i, -\pi(i))$ in 2D
Problem 2: Maxima

• Given point set $S$ in 2D,
  - $p=(x,y)$ dominates $q=(x',y')$ if $x > x' \& y > y'$
  - $q$ in $S$ is maximal if no pt in $S$ dominates $q$
  - find all maximal points in $S$

• Exercise: $O(n \log n)$ time in 2D

[History: Kung,Luccio,Preparata'75, Gabow,Bentley,Tarjan'84]
Problem 3: Orthogonal Segment Intersection

- Given n horizontal/vertical line segments in 2D, report all intersections.

Exercise: $O(n \log n + k)$ time ($k = \text{output size}$)

[History: Bentley, Ottmann'79, Overmars'87, ...]
Problem 4: Rectangle Enclosure

- Given n axis-aligned rectangles in 2D, report all pairs (r,s) where r encloses s

- Exercise: $O(n \text{ polylog } n + k)$ time ($k =$ output size)

[History: Bentley, Wood'80, Vaishnavi, Wood'80, Lee, Preparata'82, Gupta, Janardan, Smid, DasGupta'95]
The "Obvious" Divide-&-Conquer Alg'm

- For (red-blue) dominance counting/existence/reporting

- In 2D:

\[ T(n) = 2 \ T(n/2) + O(n \log n) \quad \Rightarrow \quad O(n \log^2 n) \]

- By pre-sorting:

\[ T(n) = 2 \ T(n/2) + O(n) \quad \Rightarrow \quad O(n \log n) \]

[+k for reporting]
The "Obvious" Divide-&-Conquer Alg'm

• In d-D:
  \[ T_d(n) = 2T_d(n/2) + T_{d-1}(n) + O(n) \]
  \[ \Rightarrow T_d(n) = O(n \log^{d-1} n) \]

• Side Remark: sol'n not tight for nonconst d...
  \[ T_d(n) \leq 2^{O(d)} n^{1+\epsilon} \]
  e.g., for \( d \leq \delta \log n \), \( T_d(n) = O(n^{1+\epsilon}) \)

[Hint: change of var \( m = c^d n \)
  \[ T'(m) \leq 2 T'(m/2) + T'(m/c) + O(m) \]
Detour: Appl’n to All-Pair Shortest Paths

• Given real-weighted directed tripartite graph \((A \cup B \cup C, E)\), \(|A|=|C|=n, |B|=d\), find shortest (length-2) path from every \(a\) in \(A\) to every \(c\) in \(C\)

• Trivial sol’n: \(O(dn^2)\) time

• Better sol’n: Fix \(b_0\) in \(B = \{b_1,\ldots,b_d\}\).
  Find all shortest paths that use \(b_0\), i.e., find all \(a\) in \(A\), \(c\) in \(C\)
  s.t. \(\forall i=1,\ldots,d, \ w(a,b_0) + w(b_0,c) < w(a,b_i) + w(b_i,c)\)
  i.e. \(\langle w(a,b_i) - w(a,b_0) \rangle_i\) dominates \(\langle w(b_0,c) - w(b_i,c) \rangle_i\) in \(d\)-D
Detour: Appl’n to APSP (Cont’d)

\[ \forall i = 1, \ldots, d, \ w(a, b_0) + w(b_0, c) < w(a, b_i) + w(b_i, c) \]

i.e. \( \langle w(a, b_i) - w(a, b_0) \rangle_i \) dominates \( \langle w(b_0, c) - w(b_i, c) \rangle_i \) in d-D

\[ \Rightarrow \] for \( d = \delta \log n \), time \( o(n^2) + k \)

\[ \Rightarrow \] total time for all \( b_0 \): \( O(n^2) \)

\[ \Rightarrow \] general APSP in \( O(n^3 / \log n) \) time [C., WADS’05]

[improved over … Floyd–Warshall’59, Friedman’76, Takaoka’92, Dobosiewicz’90, Han’04, Takaoka’04,’05, Zwick’04]

• Current record: \( O(n^3 \log^3 \log n / \log^2 n) \) [C., STOC’07]
...Back to Dominance in Low-D...

• How to beat $O(n \log^{d-1} n)$?

• Rest of Talk:

  I. Faster Worst-Case Alg'ms
  
  II. Beyond Worst-Case Alg'ms ("Instance Optimality")
Improvement 1 (Maxima/Dominance Existence)

- Assume pre-sorted input
- Solve 2D problem in $O(n)$ time... by sweeping from right to left

$\Rightarrow$ in $d$-D, $O(n \log^{d-2} n)$ time

[Kung, Luccio, Preparata, JACM '75]
Improvement 2 (Maxima/Dominance Existence)

• Solve 3D problem faster by sweeping from right to left... & using data structures...

with van Emde Boas (vEB) trees, in $O(n \log \log n)$ time

$\Rightarrow$ in $d$-D, $O(n \log^{d-3} n \log \log n)$ time

[Gabow, Bentley, Tarjan, STOC'84]

Better? No progress since...
A New Result [C., Larsen, Pătraşcu’11]

- 4D problem can be solved in $O(n \log n)$ time

$\Rightarrow$ in $d$-D, $O(n \log^{d-3} n)$ time
Theorem [Chazelle, FOCS’89]

- Given 2 convex polyhedra in 3D with n vertices, we can compute their intersection, or their convex hull, in $O(n)$ time

[Proof: 26-page paper!]
Connection of Maxima to Convex Hull

- Assume coords are in \{1,\ldots,n\} (by pre-sorting)

- Map \(p=(i,j)\) to \(p^* = (3^i,3^j)\)

- Map \(q=(a,b)\) to halfspace
  \[q^{**} = \{ \frac{x}{3^a} + \frac{y}{3^b} \leq 2 \}\]

- \(p\) is dominated by \(q\)
  \[\iff i \leq a \land j \leq b \iff p^*\) lies in \(q^{**}\)

- minimal points of \(P\)
  \[\iff\) vertices of (lower-left) convex hull of \(P^*\)

- dominance range searching in \(P\)
  \[\implies\) halfspace range searching in \(P^*\)
<table>
<thead>
<tr>
<th>Authors</th>
<th>Space Complexity</th>
<th>Query Time Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aggarwal, Hansen, Leighton [STOC'90]</td>
<td>( n \log n )</td>
<td>( \log n + k )</td>
</tr>
<tr>
<td>C. [FOCS'98], Ramos [SoCG'99]</td>
<td>( n \log \log n )</td>
<td>( \log n + k )</td>
</tr>
<tr>
<td>Makris, Tsakalidis'98</td>
<td>( n \log \log \log n )</td>
<td>( \log n \log \log \log n + k )</td>
</tr>
<tr>
<td>Nekrich [SoCG'07]</td>
<td>( n \log n )</td>
<td>( \log^3 \log n + k )</td>
</tr>
<tr>
<td>Afshani [ESA'08]</td>
<td>( n )</td>
<td>( \log^2 \log n + k )</td>
</tr>
<tr>
<td>Afshani, Chan [SODA'09]</td>
<td>( n )</td>
<td>( \log n + k )</td>
</tr>
<tr>
<td>C. [SODA'11]</td>
<td>( n )</td>
<td>( \log \log \log n + k )</td>
</tr>
</tbody>
</table>
A New 4D Dominance Existence Alg'm

- Divide-&-conquer \implies T_4(n) = 2T_4(n/2) + T_3(n) + O(n)

- To solve 3D red-blue dominance existence subproblem:
  not \exists pair (p,q): p is dominated by q
  \iff not \exists p,q: point p* lies in halfspace q**
  \iff (convex hull A of) all p* lie in
  (intersection B of) complement of all q**

  \iff A \cap B = A

  O(n) time by Chazelle's alg'm!

  ... provided that convex polyhedra A & B are given
4D Alg'm (Cont’d)

• but can pre-compute all A’s & B’s by bottom-up merging... by Chazelle’s alg’m again!

• pre-computation: \( T(n) = 2T(n/2) + O(n) \Rightarrow O(n \log n) \)

• rest: \( T(n) = 2T(n/2) + O(n) \Rightarrow O(n \log n) \) YES!

• Rmk: more complicated alg’ms [C., Larsen, Pătraşcu’11] to get \( O(n \log n) \) for 4D maxima (Problem 2) & \( O(n \log n + k) \) for 2D rectangle enclosure (Problem 4)

[need not Chazelle’s alg’m, but Clarkson-Shor random sampling, shallow cuttings, higher-deg. range trees, + bit packing tricks, ... (yikes!)]

3D
Rest of Talk:

I. Faster Worst-Case Alg'ms

II. Beyond Worst-Case Alg'ms
The 2D Maxima Problem, Revisited

- $\Omega(n \log n)$ worst-case lower bound under comparison/decision tree model, but...

- **Output-sensitive alg'ms:**
  - $O(nh)$ is easy, where $h = \text{output size}$
  - $\Theta(n \log h)$ [Kirkpatrick, Seidel, SoCG'85]

- "**Average-case**" alg'ms:
  - for uniformly distributed pts in a square, $O(n)$ expected [Bentley, Clarkson, Levine, SODA'90; Golin'94; Clarkson, FOCS'94; ...]
“Easy” vs. “Hard” Input
New Result [Afshani, Barbay, C., FOCS'09]

- ∃ alg'm for the 2D maxima problem that beats all other alg'ms on all point sets simultaneously!

an "instance-optimal" alg'm
Def’n of Instance Optimality (1st Attempt)

- Let $T_A(S) = \text{time of alg’m } A \text{ on input sequence } S$
- Let $OPT(S) = \min T_A(S) \text{ over all alg’ms } A$
- $A$ is instance-optimal if $\forall S, T_A(S) \leq O(1) \cdot OPT(S)$

... but not possible for 2D maxima!
[for every input sequence $S$, there is an alg’m with runtime $O(n)$ on $S$]
Our Def’n of “Instance Optimality”

average

• Let $T_A(S) = \max_{\text{time of alg'm A over all permutations of input set } S}$

• Let $\text{OPT}(S) = \min T_A(S)$ over all alg’ms $A$

• $A$ is instance-optimal in the order-oblivious setting if $\forall S$, $T_A(S) \leq O(1) \cdot \text{OPT}(S)$

[subsumes output-sensitive alg’ms, & all alg’ms that do not exploit input order, …]

[& average-case alg’ms for all point distributions as well!]
Related Work on Instance Optimality

• Fagin, Lotem, Naor’03 [finding the top k elements under a monotone aggregate scoring function]

• Sleator, Tarjan’85’s “dynamic optimality conjecture“ for binary search trees

• **Competitive** analysis of on-line alg’ms

• Various **adaptive** alg’ms, e.g.,
  Demaine, Lopez-Ortiz, Munro’00 [set union/intersection],
  Baran, Demaine’04 [approx. distance from pt to black-box curve], ...
A Measure of Difficulty

- Given point set $S$ of size $n$
- Consider a partition $P$ of $S$ into subsets $S_i$ such that each subset $S_i$ can be enclosed in a rectangle $R_i$ that is below staircase$(S)$

Let $H(P) := \sum_i |S_i| \log (n/|S_i|)$

$H(P) \sim n + (h \log n)$

$H(P) \sim h \cdot (n/h) \log h = n \log h$
A Measure of Difficulty

- Given point set $S$ of size $n$
- Consider a partition $P$ of $S$ into subsets $S_i$ s.t.
  
  each subset $S_i$ can be enclosed in a rectangle $R_i$ that is below $\text{staircase}(S)$

- Let $H(P) := \sum_i |S_i| \log (n/|S_i|)$

- Define the difficulty of $S$ to be
  
  $H(S) := \min H(P)$ over all valid partitions $P$ satisfying (*)
An Instance-Optimal 2D Maxima Alg’m

Maxima(S):
1. if \(|S| \leq 2\) then return ...
2. \(m = x\text{-median}\)
3. \(q = \text{highest pt right of } x=m\)
4. prune all pts dominated by \(q\)
5. Maxima({all pts left of \(q\} \cup \{q\})
6. Maxima({all pts right of \(q\} \cup \{q\})

• Rmk: this is not new — same as Kirkpatrick, Seidel’85’s output-sensitive alg’m !!
Analysis

• At level $k$ of recursion:

  • Let $P$ be any valid partition
  • Let $S_i$ be any subset of $P$, enclosed in rectangle $R_i$

  $\Rightarrow$ # pts in $S_i$ that survive level $k \leq \min \{n/2^k, |S_i|\}$

  $\Rightarrow$ total # pts that survive level $k \leq O(\sum_i \min \{n/2^k, |S_i|\})$
Analysis (Cont’d)

⇒ total # pts that survive level k ≤ \( O(\sum_i \min \{n/2^k, |S_i|\}) \)

⇒ runtime ≤ \( O(\sum_k \sum_i \min \{n/2^k, |S_i|\}) \)

= \( O(\sum_i \sum_k \min \{n/2^k, |S_i|\}) \)

= \( O(\sum_i (|S_i| + \ldots + |S_i| + |S_i|/2 + |S_i|/4 + \ldots )) \)

log \( (n/|S_i|) \) times

= \( O(\sum_i |S_i| \log (n/|S_i|)) \) = \( O(H(P)) \)

⇒ runtime ≤ \( O(\min_P H(P)) \) = \( O(H(S)) \)  GOOD!
Lower Bound

• Standard $\Omega(n \log n)$ proofs can't show instance-specific lower bounds...

• $2 \Omega(H(S))$ Proofs [Afshani, Barbay, C.'09]
  - An encoding-based argument
  - An adversary-based argument
Lower Bound Proof

• Build k-d tree
  cell at depth $k$
  contains $n/2^k$ pts
  i.e., depth of cell $R$
  $= \log \left( \frac{n}{|S \cap R|} \right)$

• Make cell $R$ a leaf if $R$ is below staircase($S$)
  $\Rightarrow$ leaf cells yield a valid partition $P^*$

• Adversary simulates alg'm on unknown input
• Maintain a cell $R_p$ for each input pt $p$ (initially, $R_p = \text{root}$)
Lower Bound Proof (Cont’d)

• When alg'm makes, say, x-comp. betw’n p & q:
  if depth(R_p) is odd then R_p ← any child of R_p
  if depth(R_q) is odd then R_q ← any child of R_q
  if x-median(R_p) < x-median(R_q) then
    R_p ← left child of R_p & R_q ← right child of R_q
    declare “<”
  else symmetric

• When R_p becomes a leaf, fix p to an unassigned pt in S ∩ R_p
  [Note: don't let more than |S ∩ R| points go into cell R…]

⇒ At the end, get a permutation of S
• Let $D = \sum_{p \in S} \text{depth}(R_p)$
• Each comp. increases $D$ by $O(1) \implies D \leq O(\# \text{ comps})$
• At the end, each $R_p$ must be a leaf
  (otherwise staircase could change)

$\implies \# \text{ comps} \geq \Omega(D) = \Omega\left(\sum_{\text{leaf } R} |S \cap R| \text{ depth}(R)\right)$

$= \Omega\left(\sum_{\text{leaf } R} |S \cap R| \log (n/|S \cap R|)\right)$

$= \Omega(H(P^*)) \geq \Omega(H(S)) \quad \text{Q.E.D.}$!
Other Instance-Optimal Results

• 3D/4D maxima: need a new alg'm this time, explicitly using k-d trees...

• 2D orthogonal segment intersection

• 2D red-blue rectangle enclosure

• & classical non-orthogonal problems too!

  [2D/3D convex hull, 2D point location, 2D/3D halfspace range reporting, ... under a multilinear decision tree model]
Conclusions

• find more instance-optimal results?

• worst-case complexity of Problems 1 & 3??
Problem 1: Inversion Counting

• Given permutation $\pi$ of $\{1, \ldots, n\}$, count # of pairs $(i, j)$ with $i < j$ & $\pi(i) > \pi(j)$

2 1 4 5 8 3 7 6 10 9

• Exercise: $O(n \log n)$ time

Current record: $O(n \log^{1/2} n)$ time [C., Pătrașcu, SODA'10]
Can you do better??
Problem 3: Orthogonal Segment Intersection

- Given $n$ horizontal/vertical line segments in 2D, report all intersections

- Exercise: $O(n \log n + k)$ time ($k = \text{output size}$)

For pre-sorted input, $O(n \log \log n + k)$ by vEB trees

But is $O(n + k)$ time possible? [Buttello, Bentley, Overmars'87,...]
The End