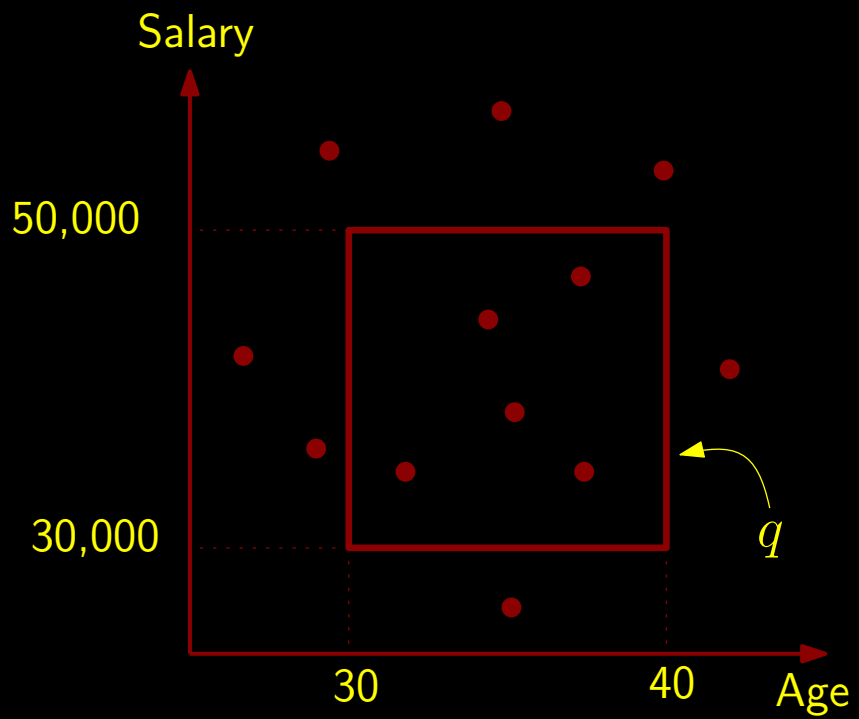
Data Structures for Geometric Intersection Query Problems

Saladi Rahul
Advisor: Prof. Ravi Janardan

Doctoral Candidate,

Dept. of Computer Science & Engg., University of Minnesota Twin-Cities

Range Searching

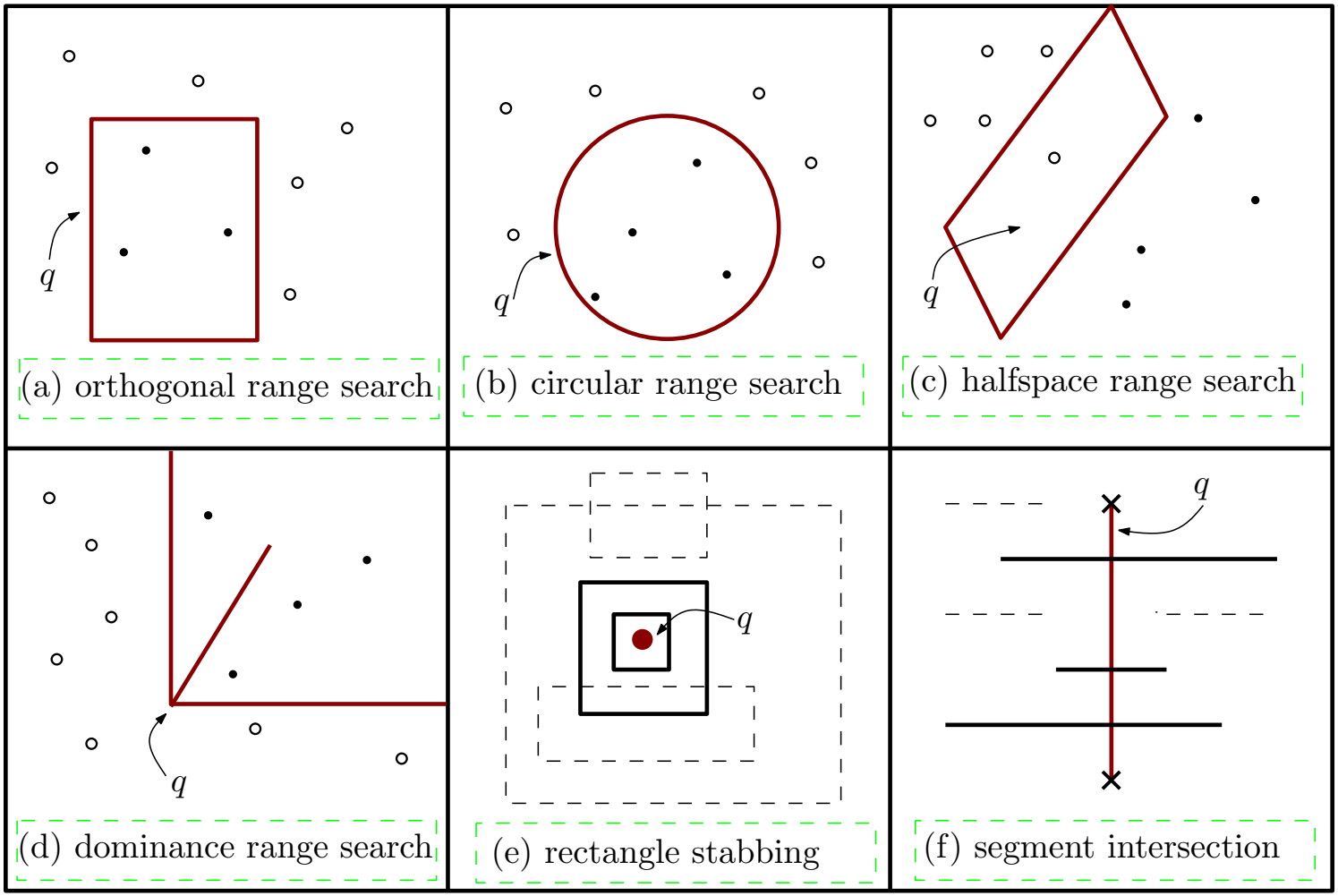


Performance Measures

- Size of the data structure
- Query time
- Update time
- Preprocessing time

Landscape of Geometric Intersection Queries (GIQ)

(1) Geometric Settings



(2) Aggregation Function

- reporting, counting.
- max, top-k, sum.
- convex hull, skyline.
- minimum spanning tree.
- closest pair.
- color (or group-by).

(3) Fundamental Structures and Techniques

- Balanced partition of objects. priority search tree, range trees, interval tree, segment tree, B-tree, R-tree, Kd-tree.
- More Sophisticated Tools. persistence, filtering search, fractional cascading.
- Randomization and Approximation Tools. ε-sample, ε-nets, moments technique.
- Integer Data. Van Emde Boas tree, fusion tree, *FindAny* structure
- Recent Discoveries. Buffer Trees, stronger version of filtering search, shallow cuttings for orthogonal problems.
- Very High Dimensional Space. Matrix multiplication, . . . *new ideas needed*

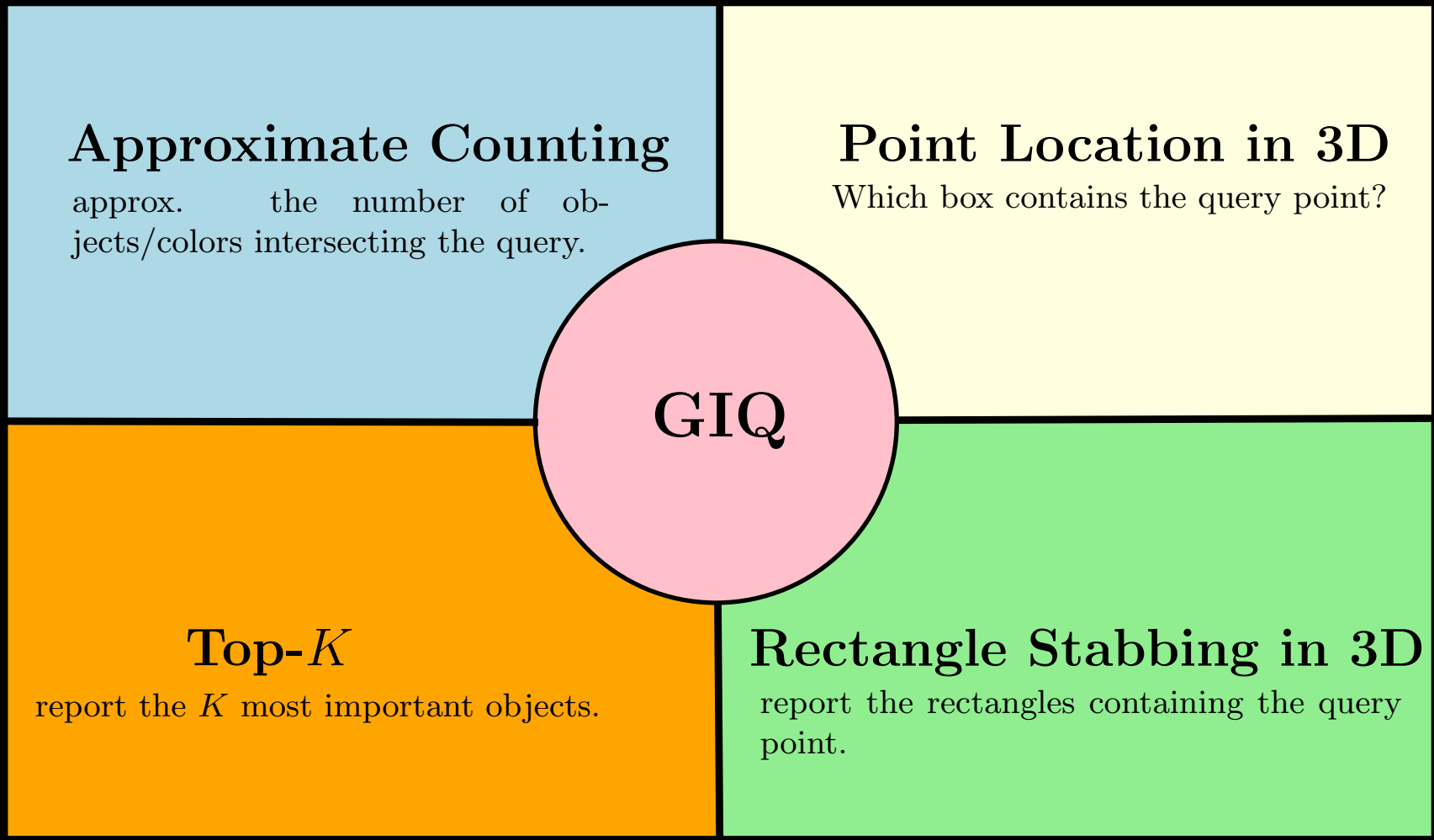
Philosophy of our research

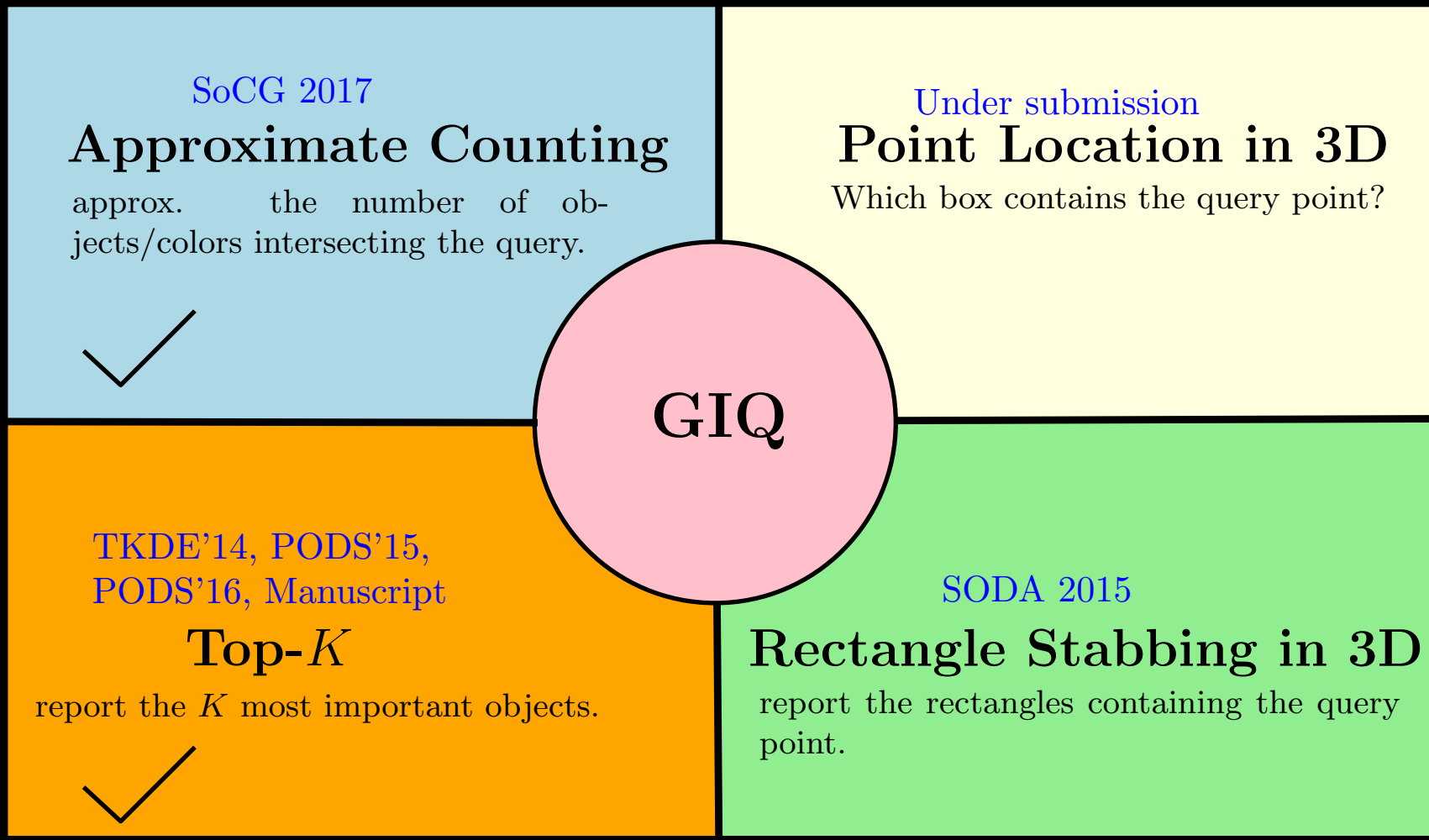
Design of geometric algorithms & data structures and their formal mathematical analysis.

Quest for optimality...

- How far can you push the space & query time bounds?
- (Curse of dimensionality) 1D vs 2D vs 3D vs ...

Scope of the thesis


Approximate Counting
approx. the number of objects/colors intersecting the query.
Point Location in 3D
Which box contains the query point?
GIQ
Top-K
report the K most important objects.
Rectangle Stabbing in 3D
report the rectangles containing the query point.

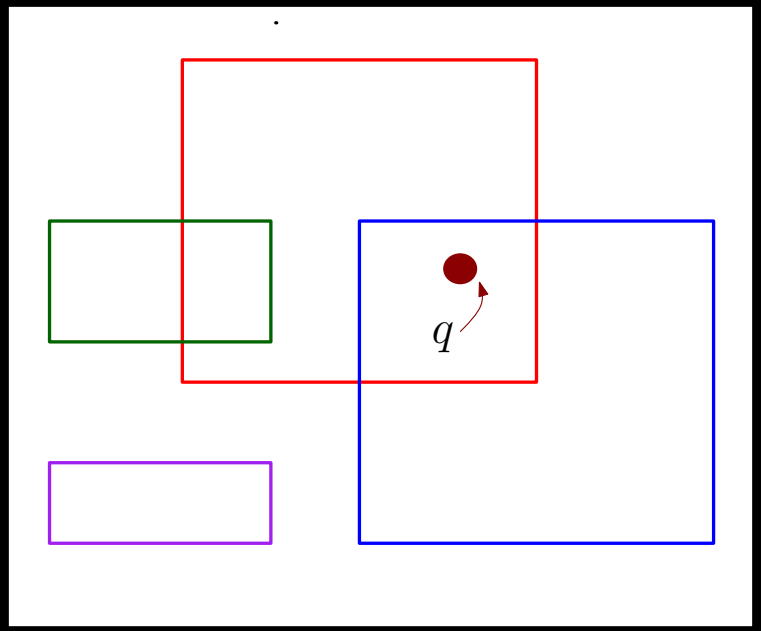

SoCG 2017
Approximate Counting
approx. the number of objects/colors intersecting the query.
Under submission
Point Location in 3D
Which box contains the query point?
GIQ
TKDE'14, PODS'15, PODS'16, Manuscript
Top-K
report the K most important objects.
SODA 2015
Rectangle Stabbing in 3D
report the rectangles containing the query point.

Rectangle Stabbing

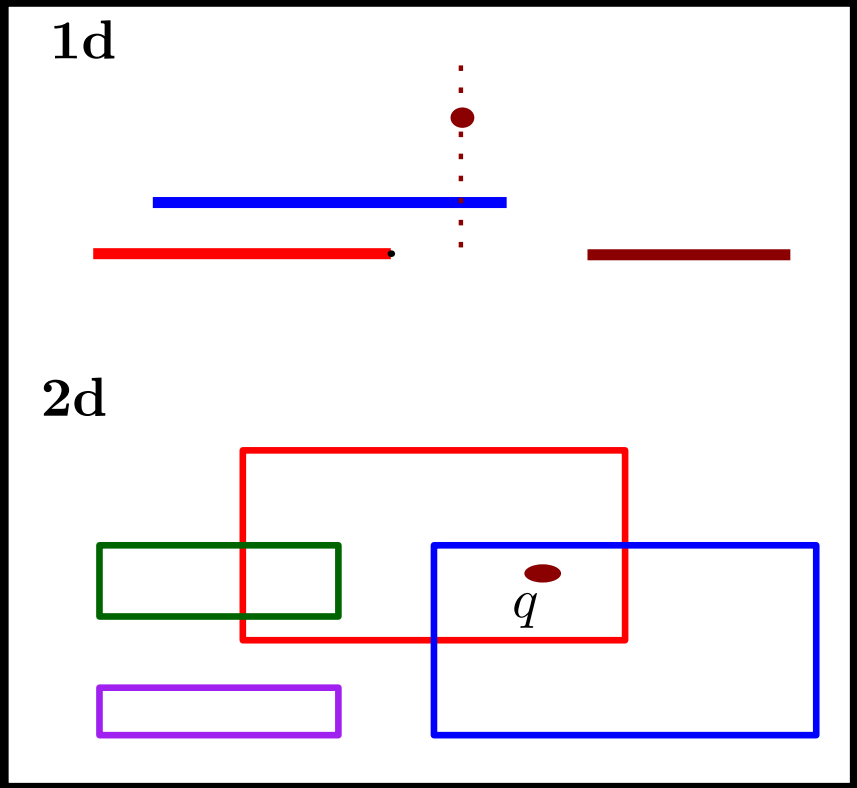
(Almost) resolved a three-decade old open problem.

Saladi Rahul. *Improved bounds for orthogonal point enclosure query and point location in orthogonal subdivisions in* $\mathbb{R}^3$. SODA 2015.

Problem



Optimality in $1d$ and $2d$



Space: $O(n)$

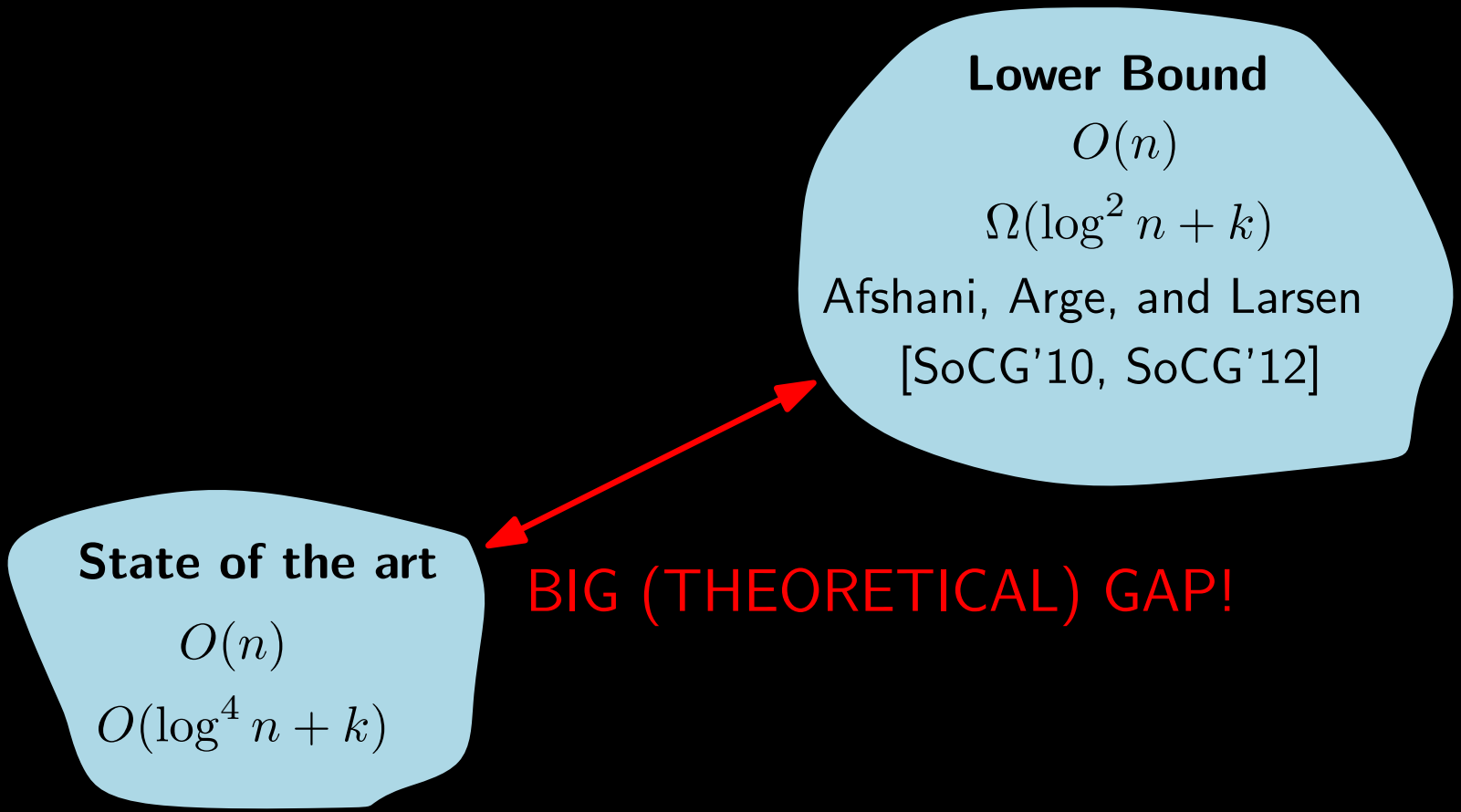
Query Time: $O(\log n + k)$

Space: $O(n)$

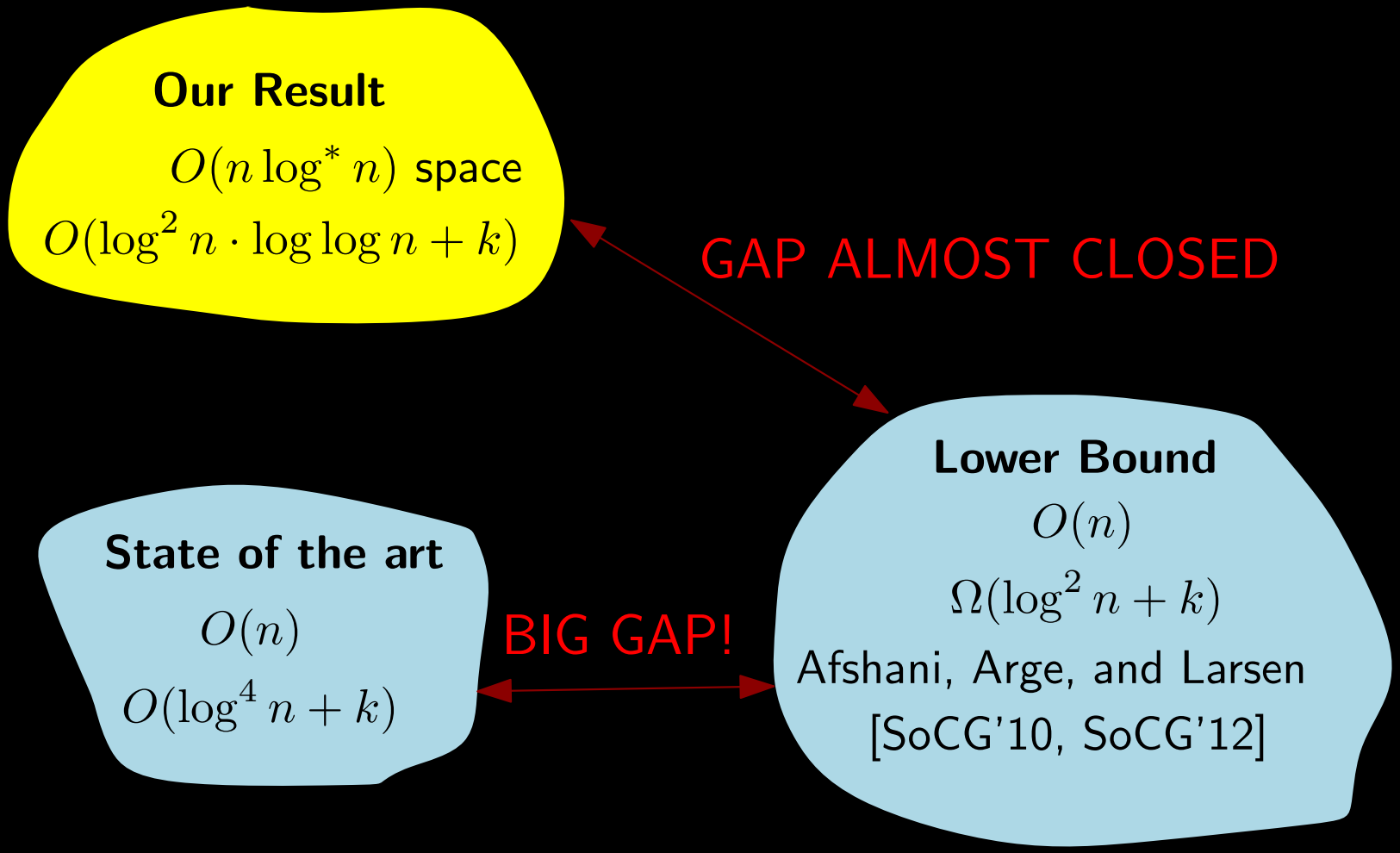
Query Time: $O(\log n + k)$

Comparison Model and Pointer Machine model: $\Omega(\log n + k)$

Rectangle stabbing in $3d$



Almost Optimal Result in $3d$

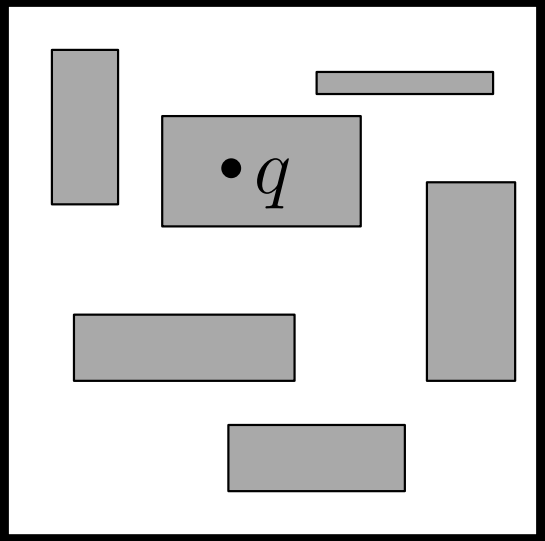


Orthogonal Point Location

(Designed the first optimal solution in 3D)

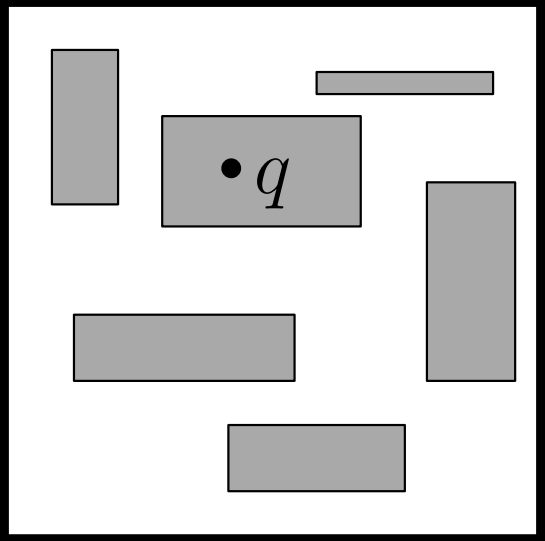
Under Submission.

Problem in 2D



Problem in 3D

Figure shown in 2D for convenience



History of point location in 3D

Reference	**Space**	**Query Time**
Edelsbrunner et al.	n	$\log^3 n$
Afshani et al.	n	$\frac{\log^2 n}{\log \log n}$
Rahul	n	$\log^{1.5} n$
Chan	n	$\log n \log \log n$
New	n	$\log_w n$
Nekrich	n/B	$\log_B^2 n$
New	n/B	$\log_B n$

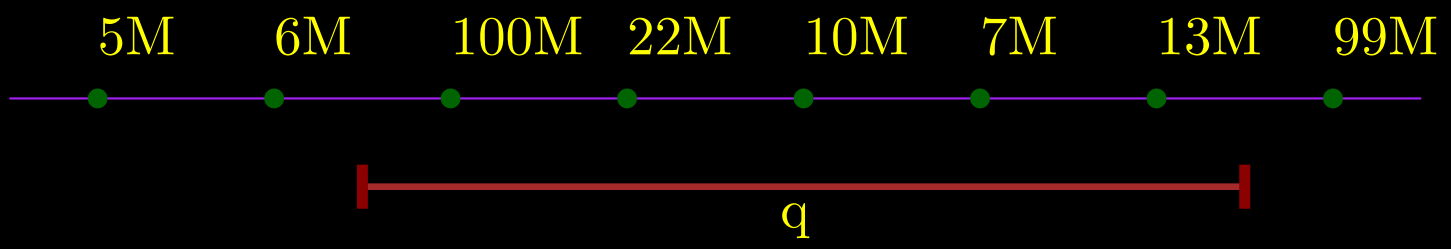
Top-k Geometric Intersection Queries (Top-k GIQ)

Why Top-k?

- Big Data. What happens if the database returns too many results?
- Reduce Cognitive Overload. "Enough Already!" [Carey and Kossmann'97]
- Smartphones. Limited screen size.

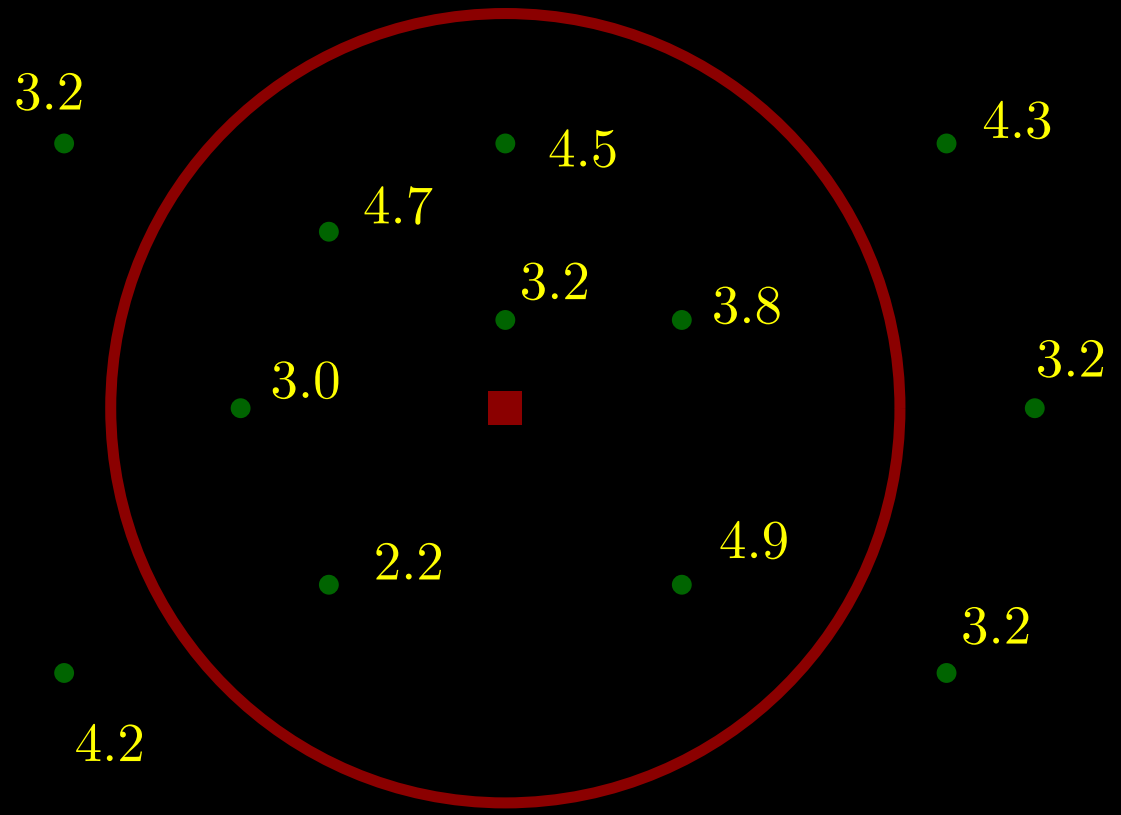
1D Top-k Range Search

Find the k most viewed youtube videos which were published between 1st June 2000 and 1st June 2005.



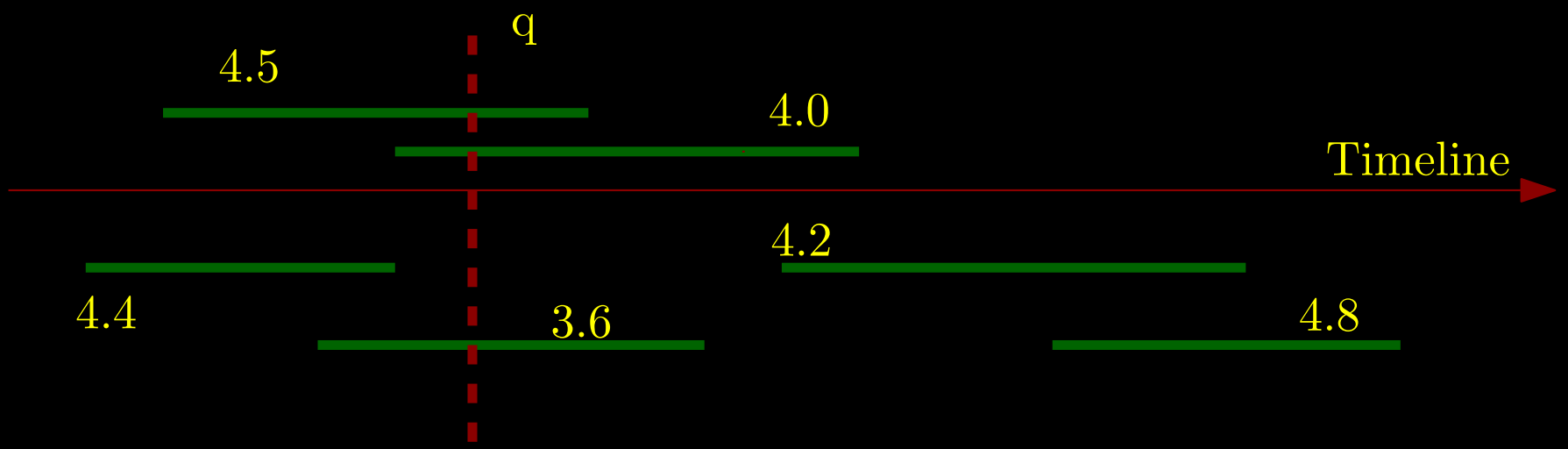
Top-k Circular Range Search

Find the k best-rated nearby restaurants.



Top-k Interval Stabbing

Report k best-rated hotels which have a vacancy on 13th Sept. 2016.



Our Contributions

Specific geometric settings.

Saladi Rahul and Yufei Tao. *On top-k range reporting in 2d space.* PODS 2015.

Yakov Nekrich, Saladi Rahul and Yufei Tao. *Optimal top-k planar rectangle stabbing and halfplane reporting.* Manuscript.

Generic reductions.

Saladi Rahul and Ravi Janardan. *A general technique for top-k geometric intersection query problems.* IEEE TKDE 2014.

Saladi Rahul and Yufei Tao. *Efficient top-k indexing via general reductions.* PODS 2016.

Specific Geometric Settings

Optimal worst-case solutions.

- Orthogonal range searching in 2D.
- Rectangle stabbing in 2D.
- Halfplane searching in 2D.

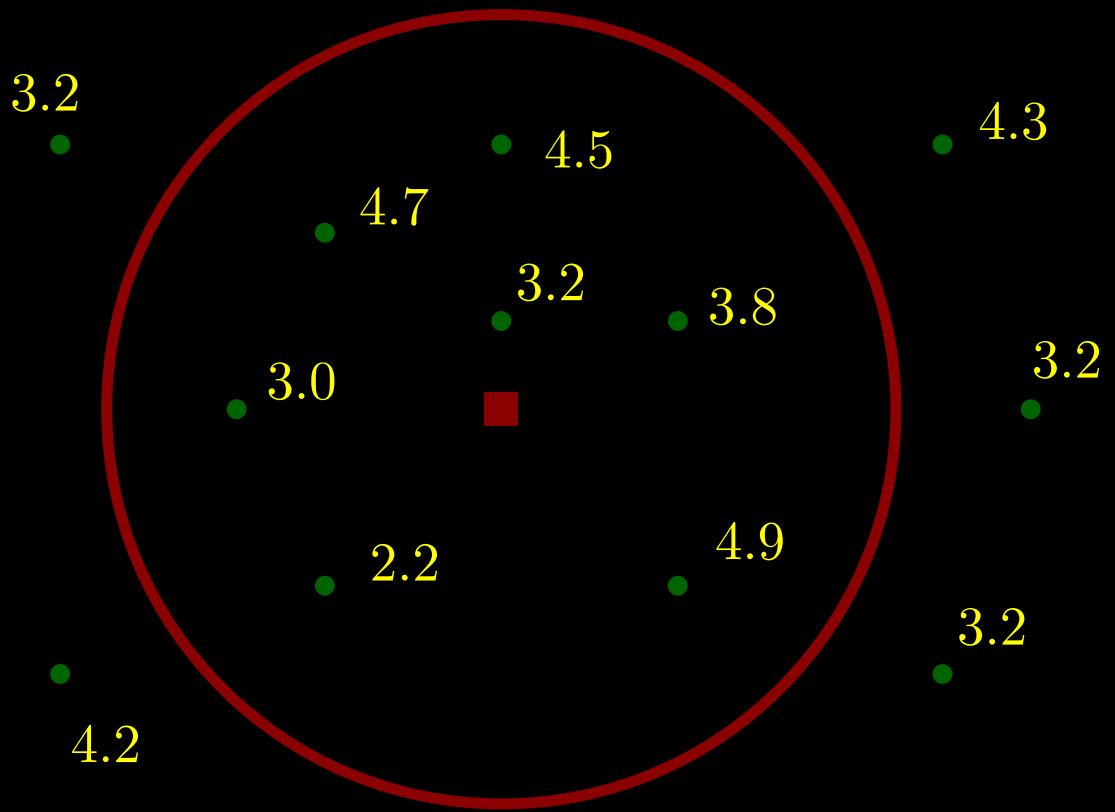
Generic Reductions (*Short and Sweet*)

- Short. Significantly simplify the design of top-k structures. Very little effort required.
- Sweet. Involves interesting and non-trivial theoretical analysis.

Techniques

Simple Approach-I (Naive Reporting)

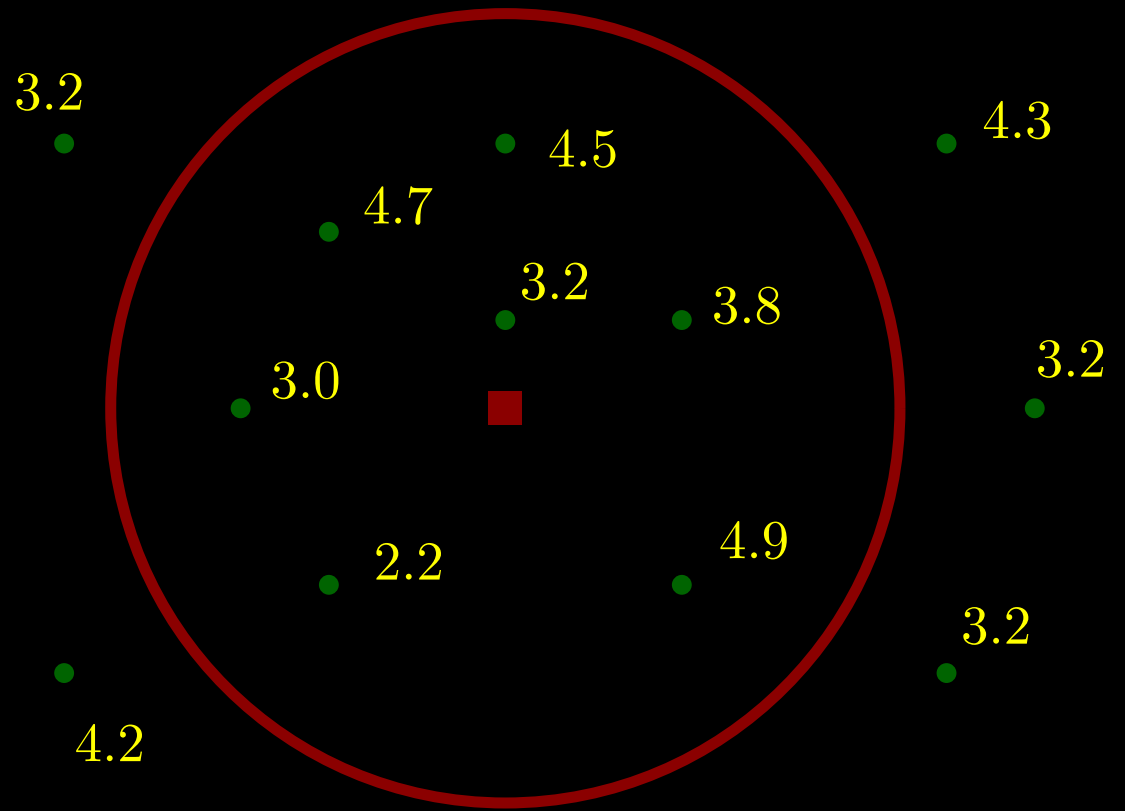
- Report all the objects intersecting the query, i.e., $\mathcal{A} \cap q$.
- Find the top-k objects in $\mathcal{A} \cap q$.
- Inefficient if $|\mathcal{A} \cap q| \gg k$.



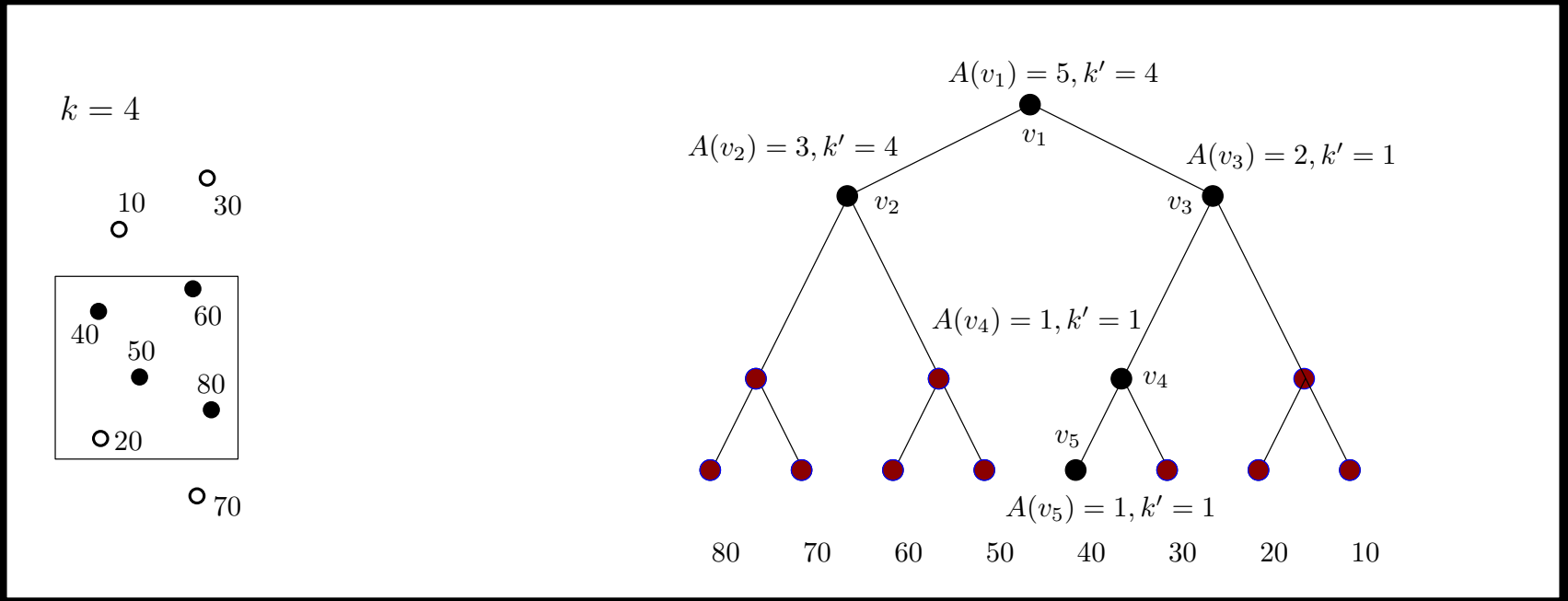
Answering a Top-k Query

Two Step Process

- Find the k-th largest weight in $\mathcal{A} \cap q$. Call it τ.
- Run a *prioritized reporting query*. Report objects with weight $\geq \tau$.

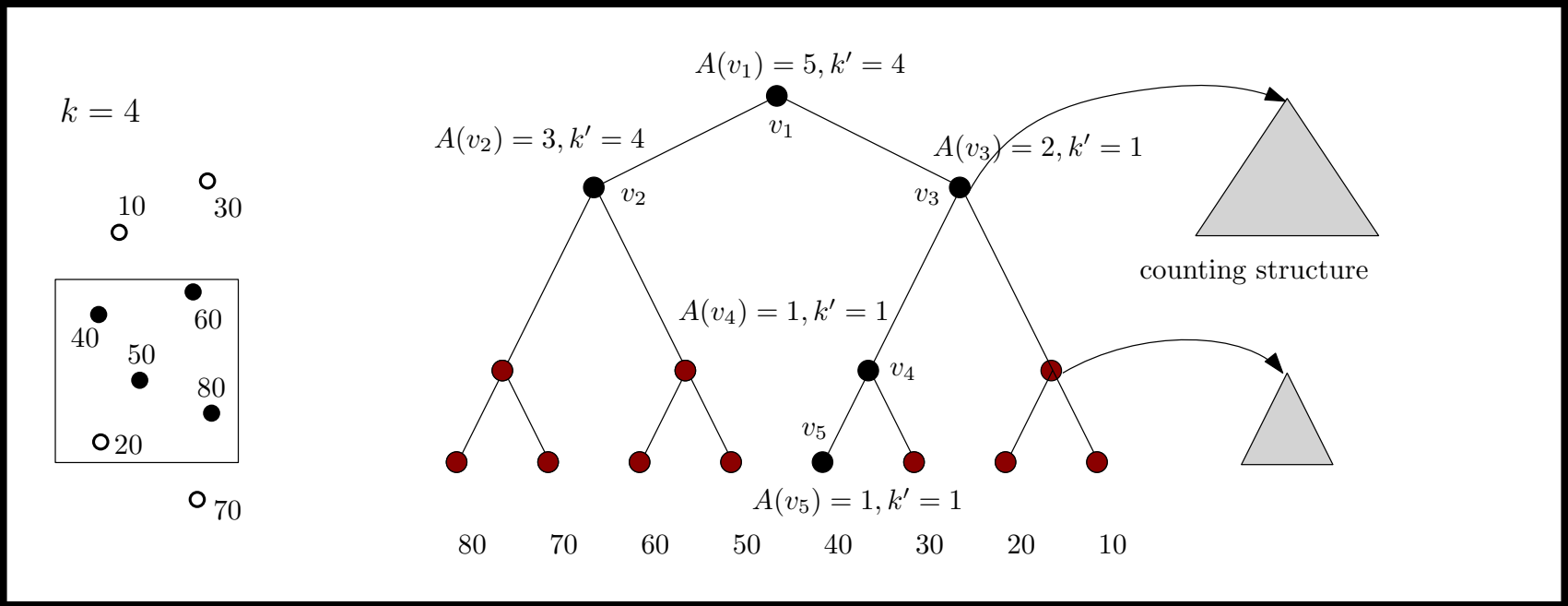


Our Approach (R & Janardan [TKDE'14])



1) Need to answer *counting* queries.
2) Only $O(\log n)$ nodes are visited.

Our Approach (R & Janardan [TKDE'14])



General Reduction-I

Given

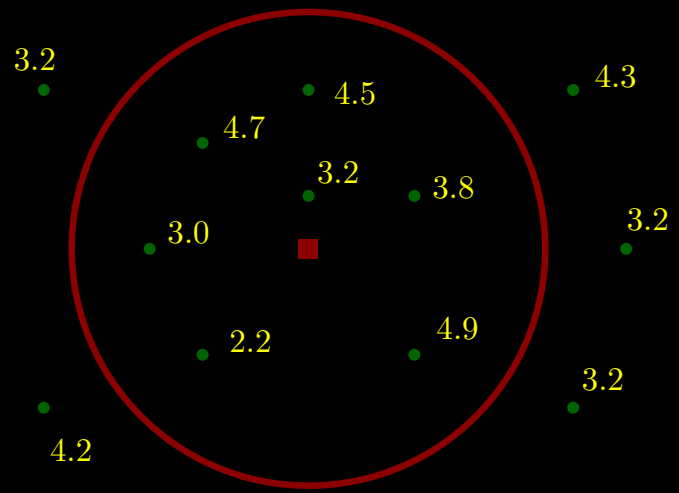
- A prioritized structure of $\mathcal{S}_{pri}(n)$ space that answers a query in $\mathcal{Q}_{pri}(n) + O(t)$ time;
- A counting structure of $\mathcal{S}_{cnt}(n)$ space that answers a query in $\mathcal{Q}_{cnt}(n)$ time.

Then there is a top-k structure with

- $\mathcal{S}_{top}(n) = O(\mathcal{S}_{cnt}(n) \cdot \log_2 n + \mathcal{S}_{pri}(n))$
- $\mathcal{Q}_{top}(n) = O(\mathcal{Q}_{cnt}(n) \cdot \log_2 n + \mathcal{Q}_{pri}(n) + k)$
- Updates handled efficiently.

Limitation

Expensive Counting Structures



Space: $O(n)$

Query time: $O(\sqrt{n})$

Can Other Aggregate Functions be Used to Solve Top-k GIQ?

Another Companion Problem

Max Query:

- Report the object with the largest weight.
- Easiest special case of Top-k query.

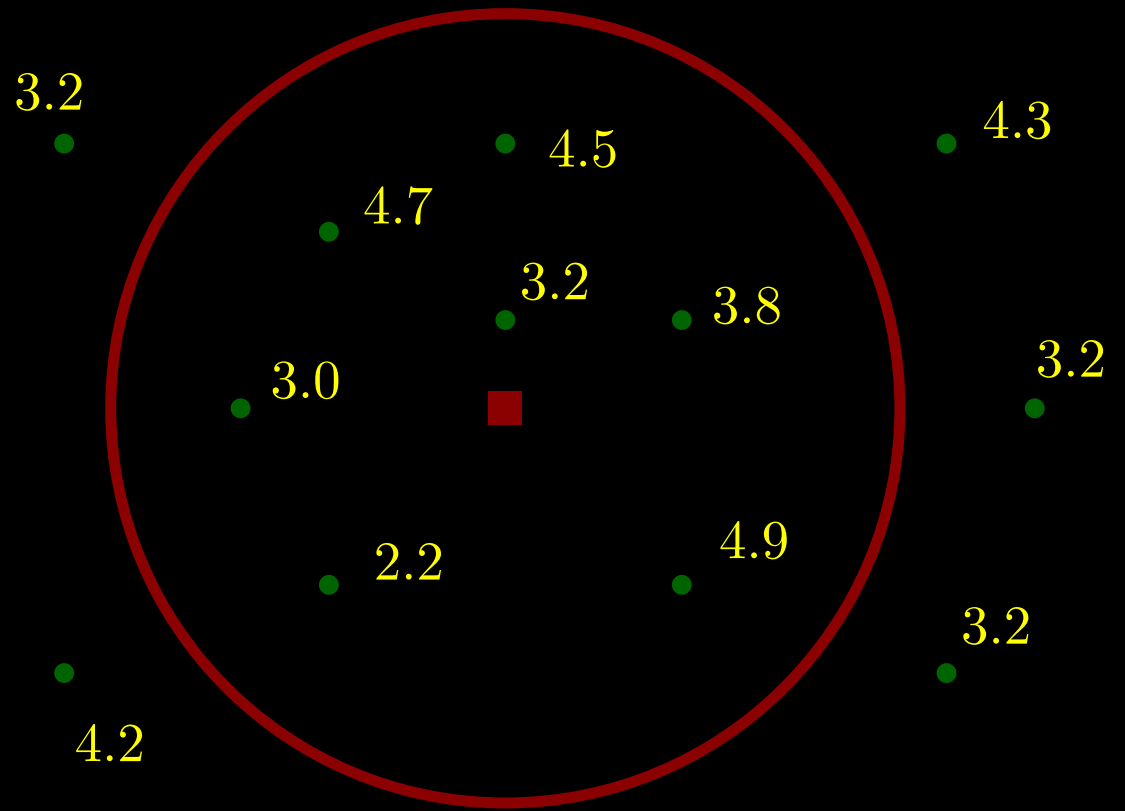
New Goal:

- Design a Top-k GIQ structure using the Max Structure.

Answering a Top-k Query

Two Step Process

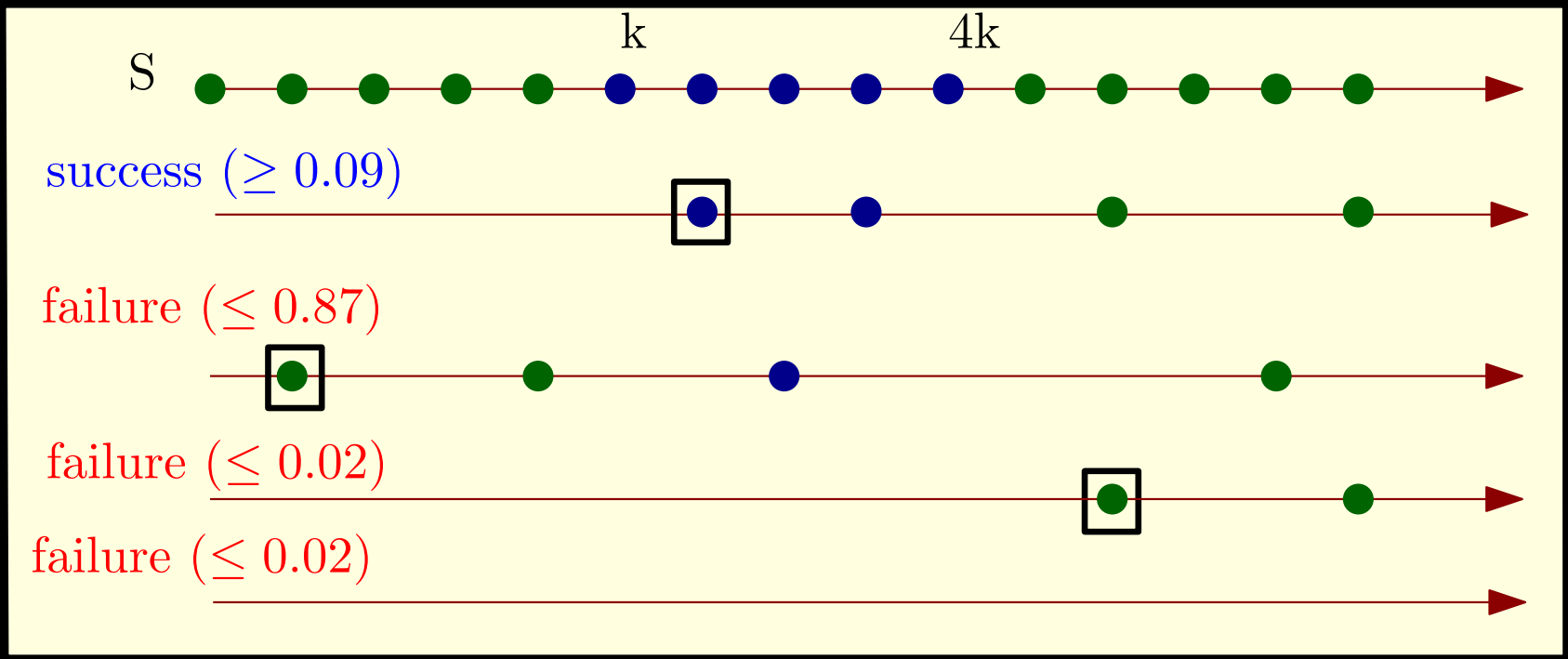
- Find the *approximate* k-th largest weight in $\mathcal{A} \cap q$. Call it τ.
- Run a prioritized reporting query. Report objects with weight $\geq \tau$.



Reducing top-k to top-1 (R & Tao [PODS'16])

Let S be a set of m elements. For a $(1/k)$-sample set R of S

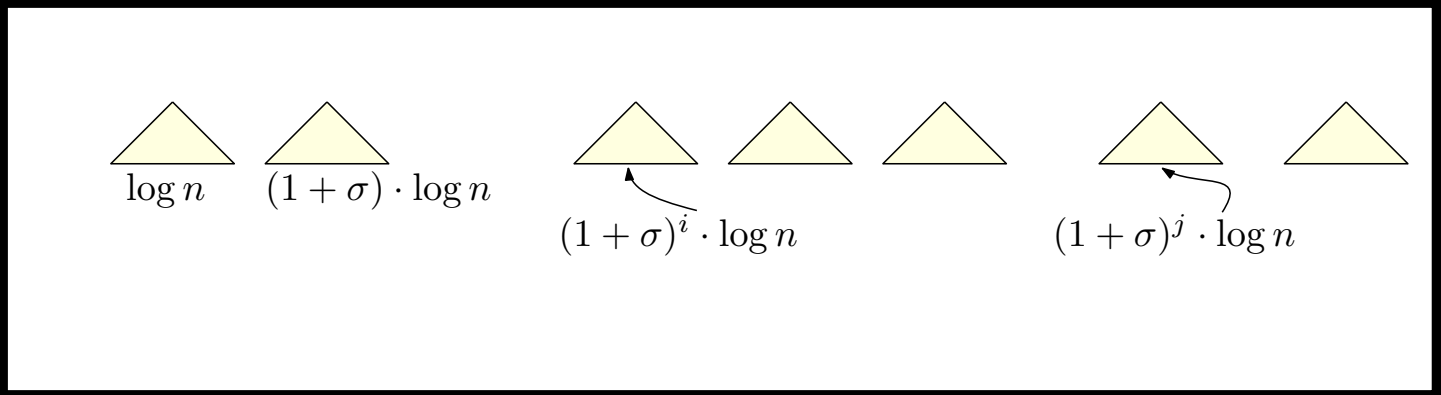
- The rank-1 element in R has rank in S in the range $[k, 4k]$, with probability at least 0.09.



$$(1 - \tfrac{1}{k})^{4k} < e^{-4} \approx 0.02$$

Build several Top-1 structures

- If you fail, go to the next structure.
- Intuition.Will visit very few structures.



$k\sum_{h=i}^{j}(0.91)^{h-i}\cdot(1+\sigma)^{h-i}) \leq k\sum(0.99)^{h-i} = O(k)$

$0.91\cdot(1+\sigma) < 1$. Pick $\sigma = 0.09$.

General Reduction-II: NO Deterioration!

Given

- A max structure of $\mathcal{S}_{max}(n)$ space, $\mathcal{Q}_{max}(n)$ query time, and $\mathcal{U}_{max}(n)$ update time.
- A prioritized reporting structure of $\mathcal{S}_{pri}(n)$ space, $\mathcal{Q}_{pri}(n)$ query time, and $\mathcal{U}_{pri}(n)$ update time.

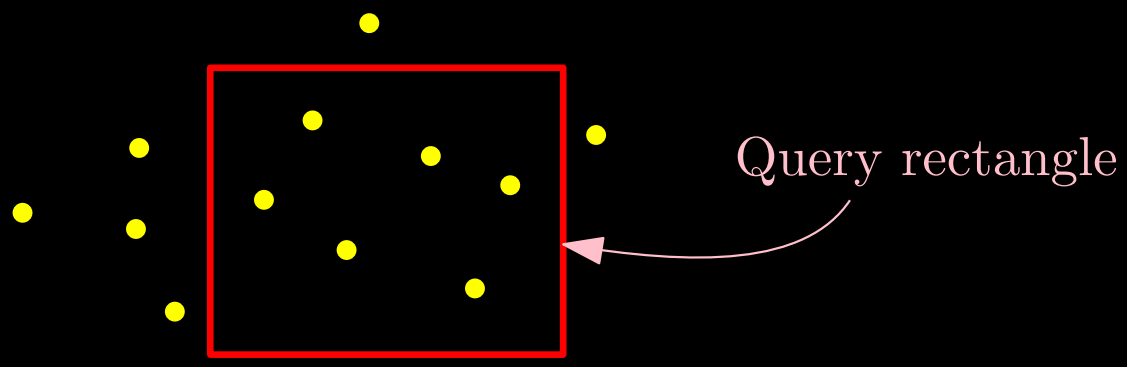
[R & Tao, PODS'16]: In expectation, there is an optimal top-k structure with:

- $\mathcal{S}_{top}(n) = O(\mathcal{S}_{max}(n) + \mathcal{S}_{pri}(n))$
- $\mathcal{U}_{top}(n) = O(\mathcal{U}_{max}(n) + \mathcal{U}_{pri}(n))$
- $\mathcal{Q}_{top}(n) = O(\mathcal{Q}_{max}(n) + \mathcal{Q}_{pri}(n))$

Approximate Counting

Saladi Rahul. *Approximate Range Counting Revisited.*
SoCG 2017.

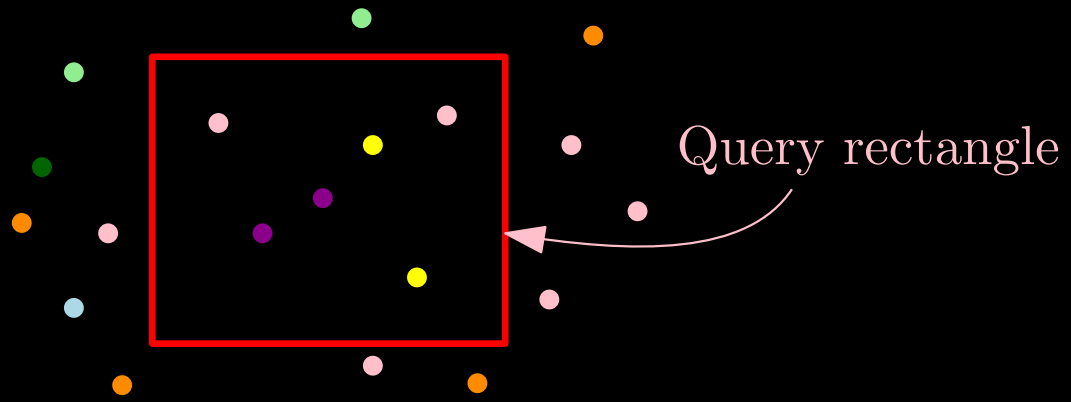
Problem-I



K= # objects intersecting the query

Approximate range counting:
Report a value in the range $[(1-\varepsilon)K, (1+\varepsilon)K]$

Problem-II (Enter the Colors...)



K= # **colors** intersecting the query

Colored approximate range counting:
Report a value in the range $[(1-\varepsilon)K, (1+\varepsilon)K]$

Previous Work

(1) ε-approximations Vapnik and Chervonenkis ['71]

(2) Relative (p, ε)-approximations Har-Peled and Sharir ['11], Aronov and Sharir ['10], Sharir and Shaul ['11]

(3) General Reductions via Sampling Aronov & Har-Peled ['08], Kaplan, Ramos and Sharir ['11]

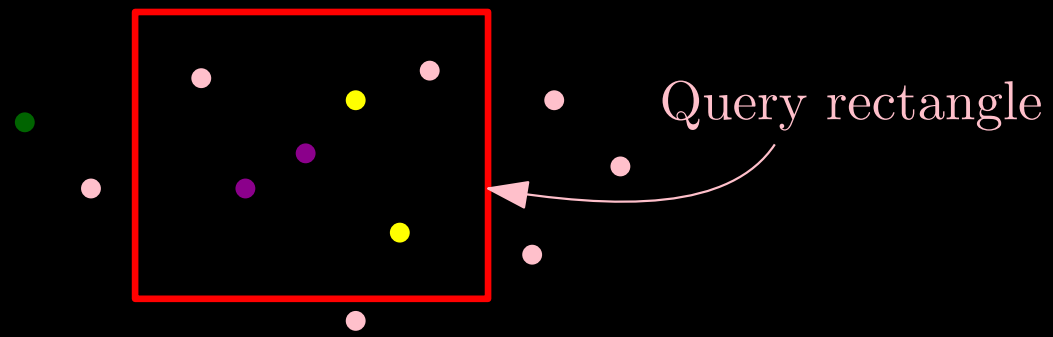
(4) Shallow Cuttings Afshani and Chan ['09], Afshani, Hamilton and Zeh ['10]

(5) Word-RAM Model Chan and Wilkinson ['13], Nekrich ['14]

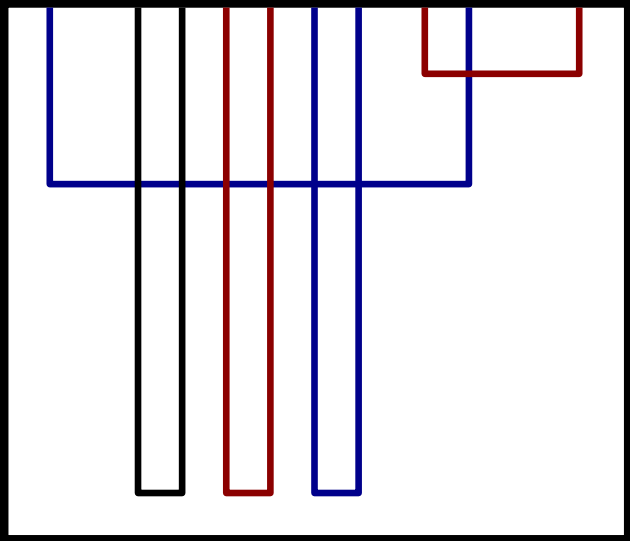
Why?

(1) Colored Orthogonal Range Search in 2D

	Space	**Query Time**
Exact	$O(n^2)$	$O(\log n)$
Reporting	$O(n \log n)$	$O(\log n + K)$
$(1+\varepsilon)$-Approximation	$O(n \log n)$	$O\left(\frac{\log n}{\varepsilon^2}\right)$



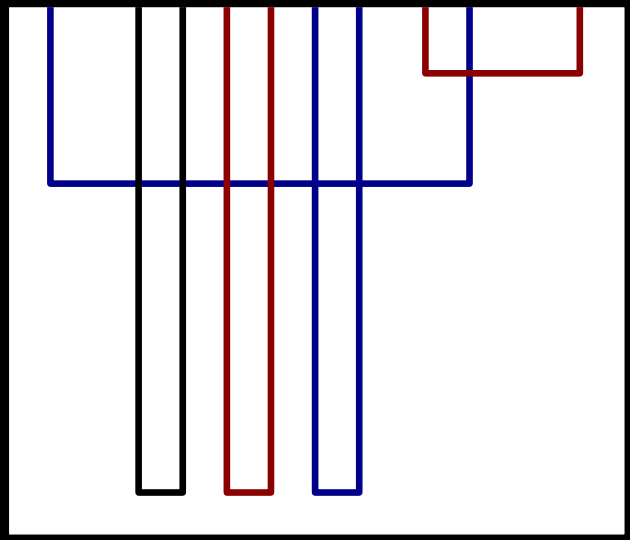
(2) 3-sided rectangle stabbing in 2D



Space: $O(n)$

Query Time: $O(\log\log U + (\log\log n)^2)$

(2) Optimal 3-sided rectangle stabbing in 2D



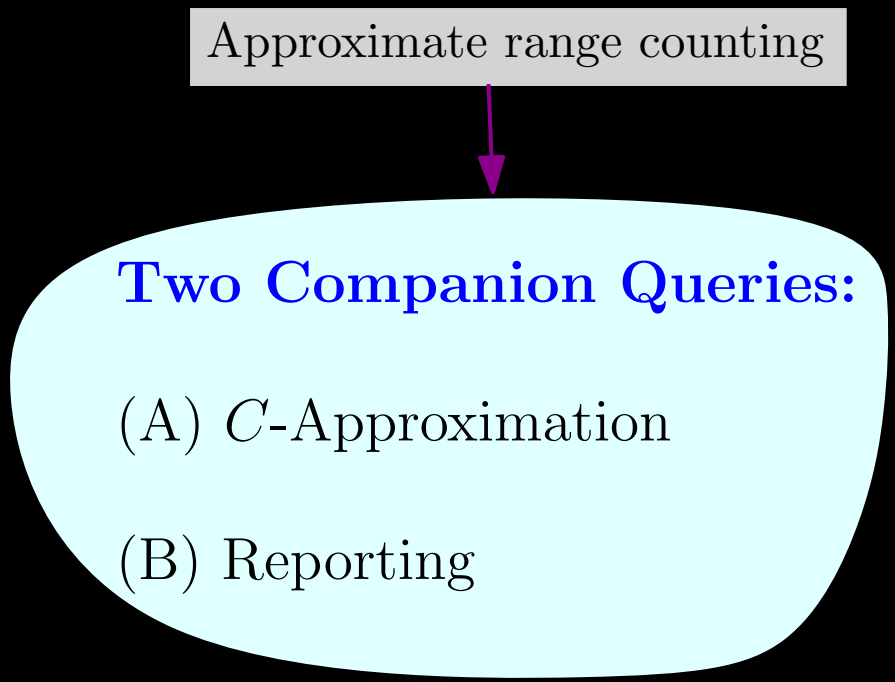
Space: $O(n)$

$O(1)$

Query Time: $O(\log\log U + (\log\log n)^2)$

A General Reduction

A General Reduction

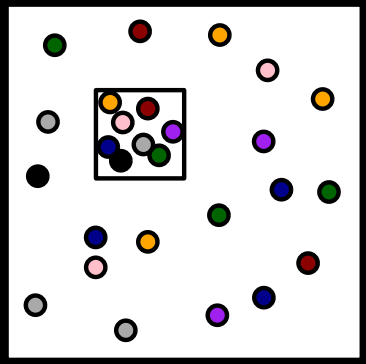


Space: $O(\mathcal{S}_{capp}(n) + \mathcal{S}_{rep}(n))$

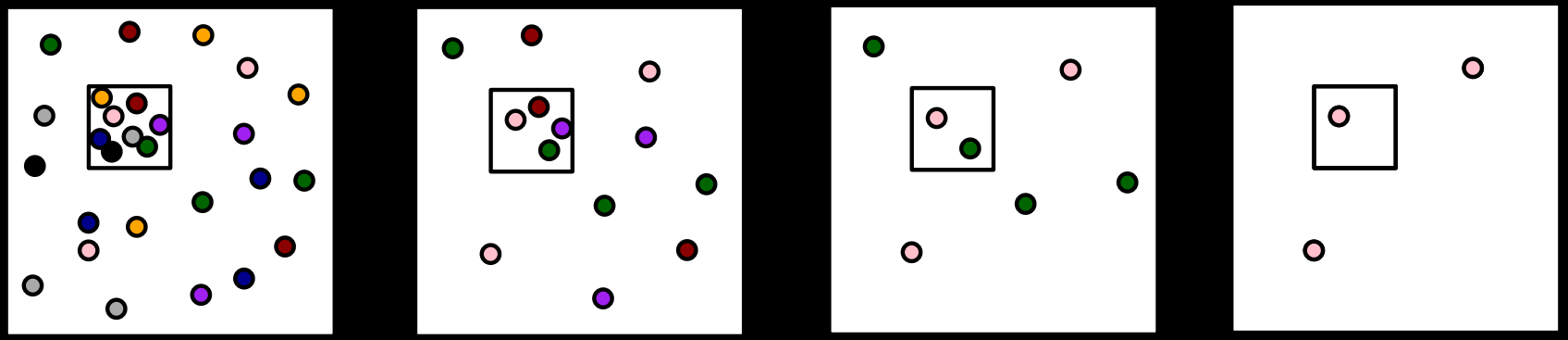
Query time: $O(\mathcal{Q}_{capp}(n) + \mathcal{Q}_{rep}(n) + \varepsilon^{-2} \log n)$

Approximate Counting via Random Sampling

Reporting Structure to Count Colors



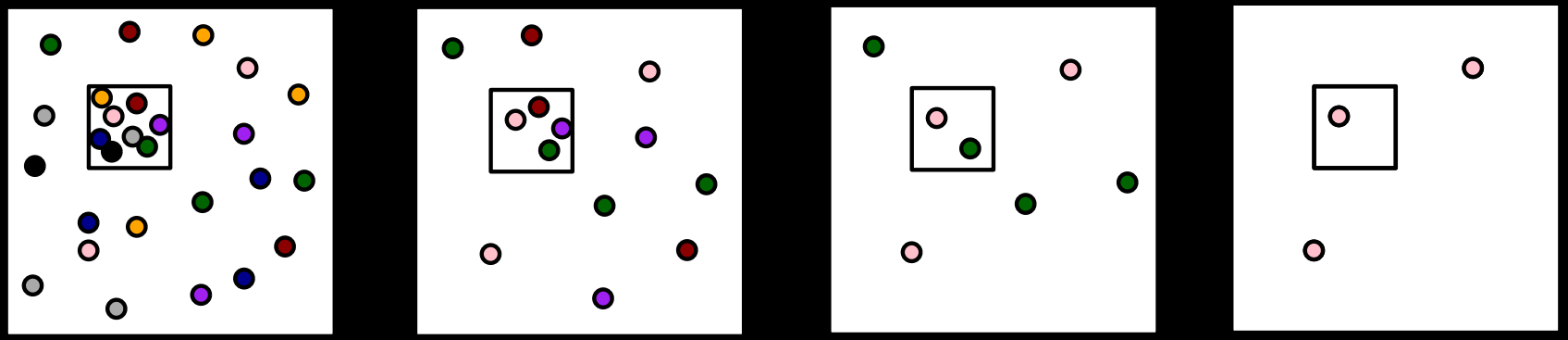
Sample, Sample, Sample....Colors



Find a structure with output-size

$$[C_1 \log n, C_2 \log n]$$

C-approximation is the saviour



Directly jump to the structure with output-size

$$[C_1 \log n, C_2 \log n]$$

Final Comments

Main techniques used

- Van Emde Boas + K-d Tree
- BITology
- Filtering search
- "Parallel" point location in 2D
- Shallow Cuttings
- Finding exact threshold
- Random sampling on objects
- Hardness result
- Transformation from colored to uncolored
- Random sampling on colors

Two Open Problems

- Orthogonal point location in $d \geq 4$.
  - Is it affected by the curse of dimensionality?
- Is top-k equivalent to prioritized reporting?
  - Conjecture: Yes.

Thank You!!