



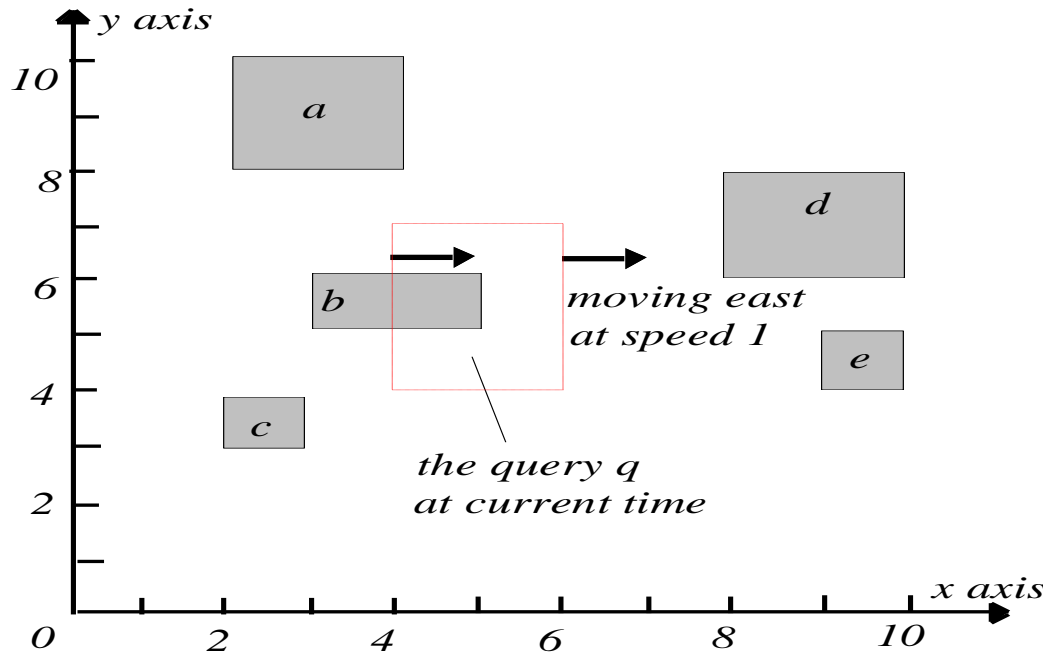
Spatio-temporal Databases

Time Parameterized Queries

Based on Slides by Prof. Yufei Tao

Intro

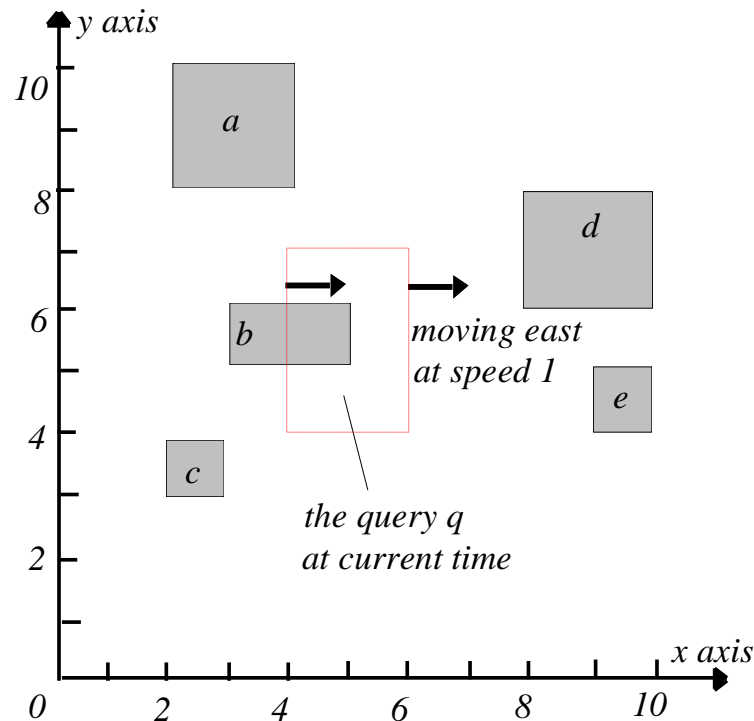
- The results of conventional spatial queries are not very useful in dynamic environments because they may be invalidated very soon due to the movements of objects and queries.



The Time Parameterized (TP) Window Query

■ Returns:

- The current query result **R**
- The validity period **T** of R
- The change of result **C** at the end of T



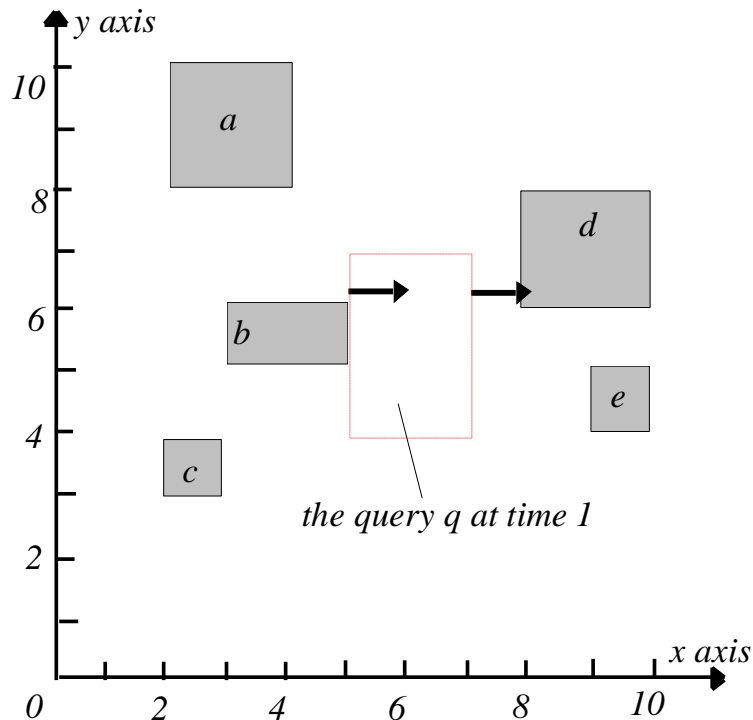
Result:

$R = \{b\}$

The TP Window Query

Returns:

- The current query result **R**
- The validity period **T** of R
- The change of result **C** at the end of T



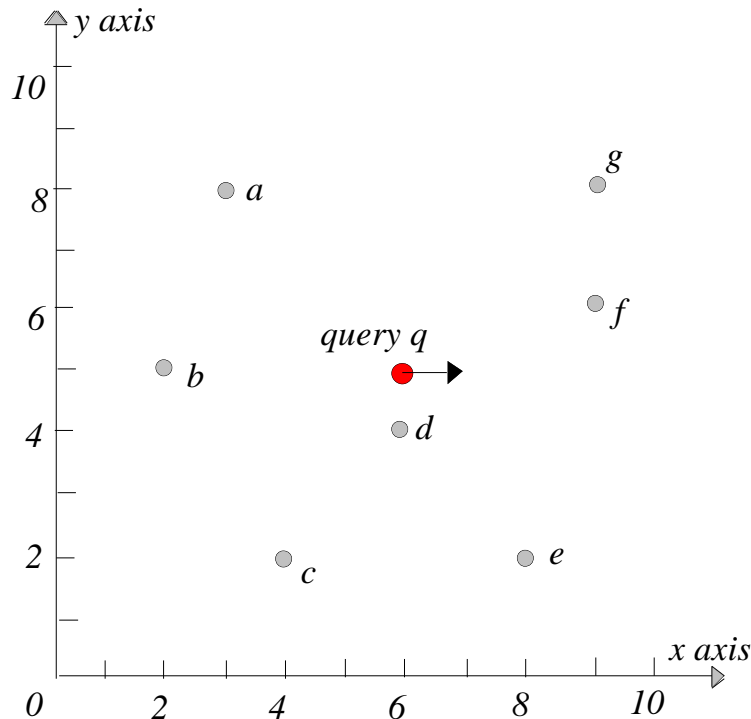
Result:

$R=\{b\}$, $T=1$, $C=\{-b\}$

The TP Nearest Neighbor Query

■ Returns:

- The current query result **R**
- The validity period **T** of R
- The change of result **C** at the end of T



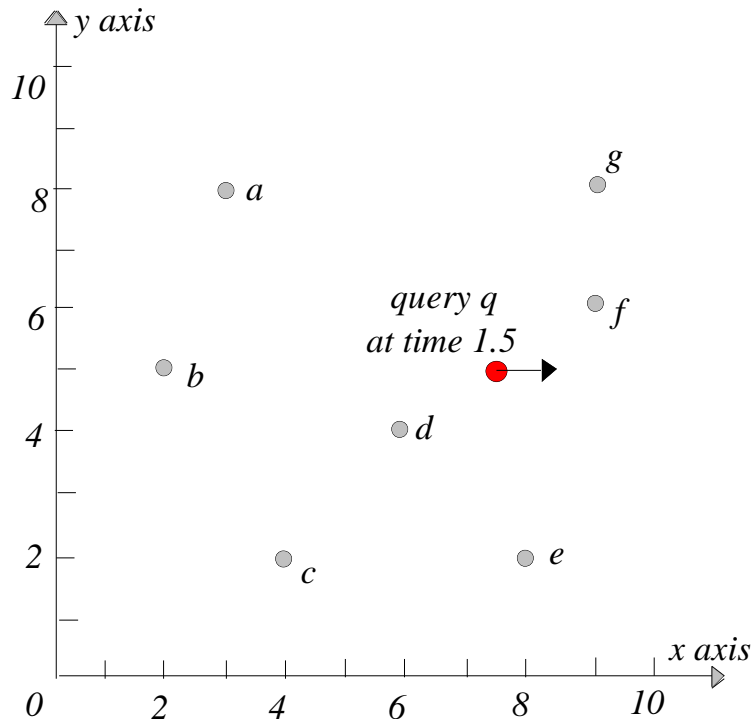
Result:

$R = \{d\}$

The TP Nearest Neighbor Query

Returns:

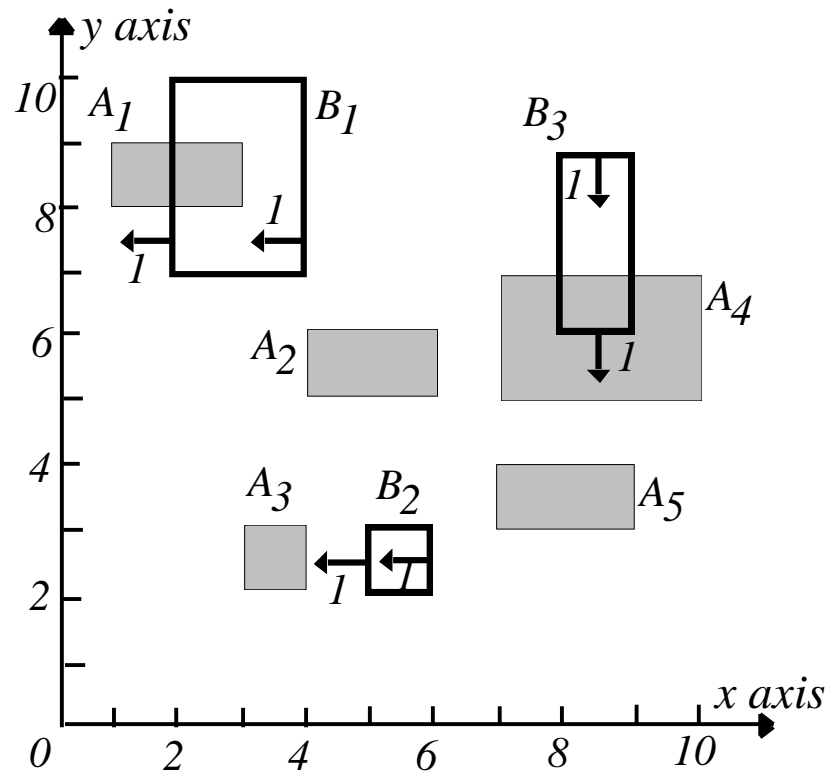
- The current query result **R**
- The validity period **T** of R
- The change of result **C** at the end of T



Result:

$R=\{d\}$, $T=1.5$, $C=\{f\}$

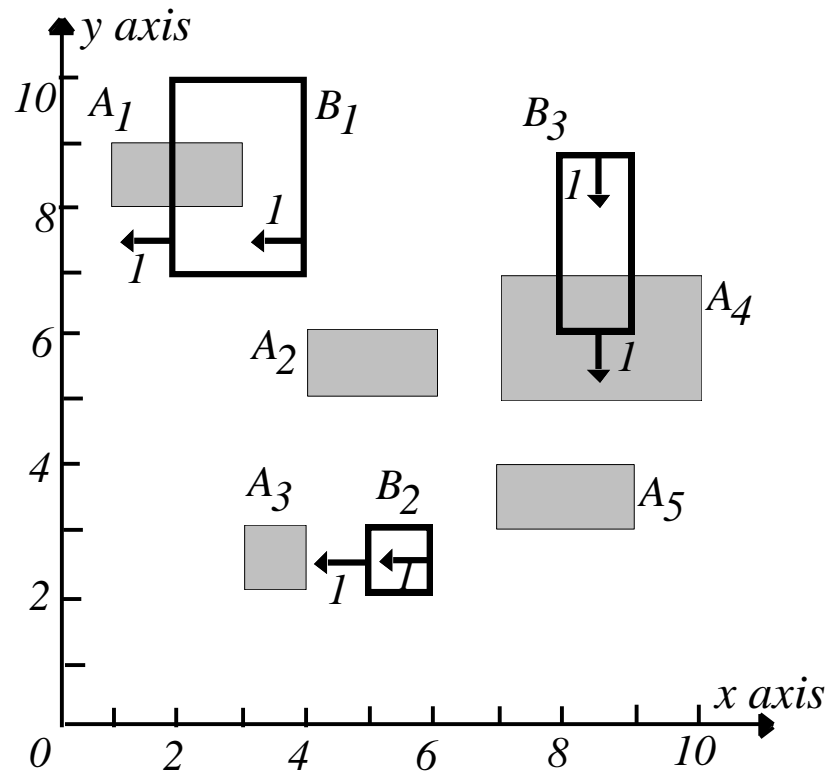
The TP Spatial Join Query



Result:

$R = \{(A1, B1), (B3, A4)\}$

The TP Spatial Join Query



Result:

$R = \{(A1, B1), (B3, A4)\}$

$T = \{1\}$

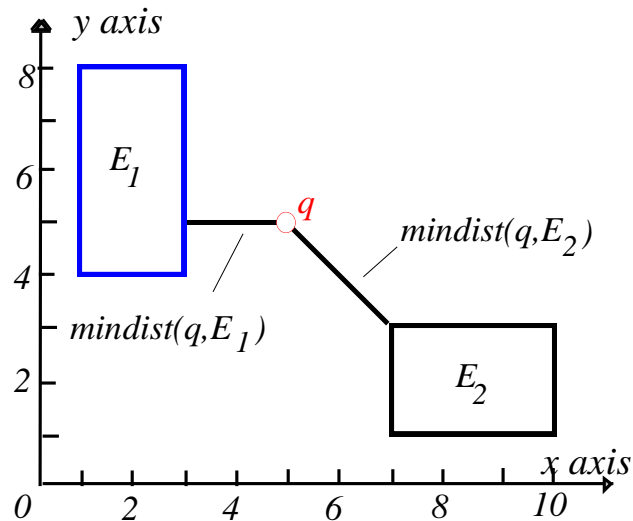
$C = \{A3, B2\}$

Point Nearest Neighbor (NN) Queries

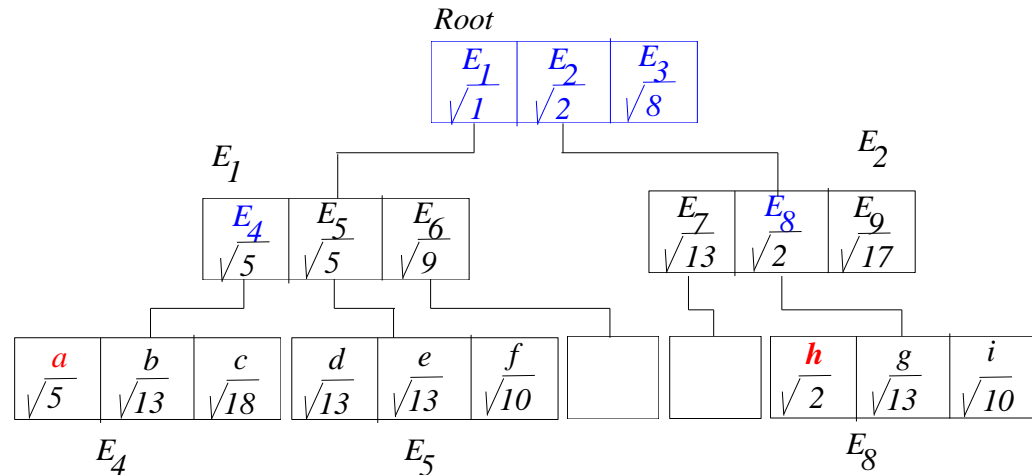
[Roussopoulos et al SIGMOD95, Hjalton and Samet TODS 99]

Branch and bound algorithms use *mindist* between the query point q and an R-tree entry E , to prune the search space:

- $\text{mindist}(E, q)$ = The minimum distance between E and q



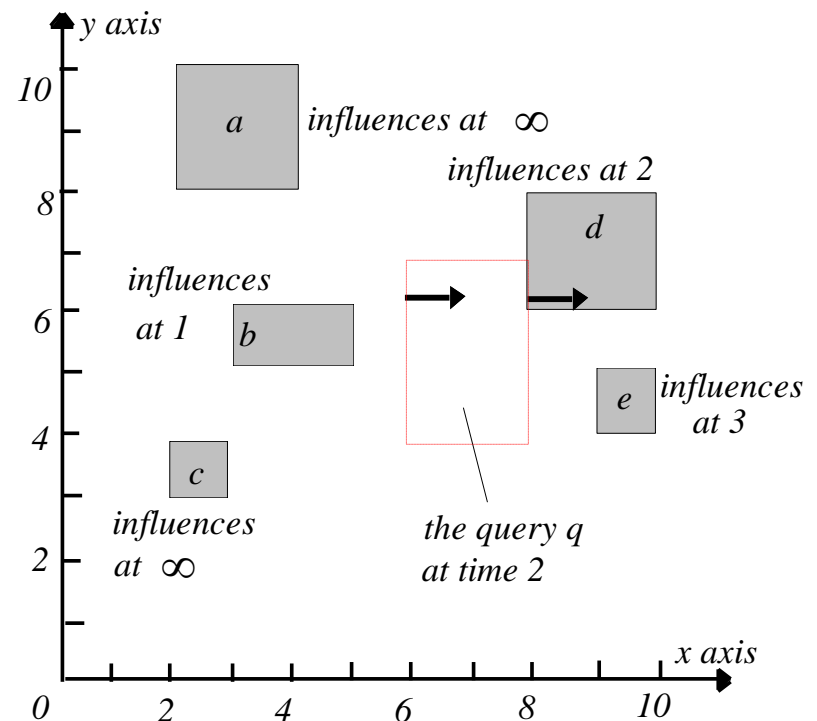
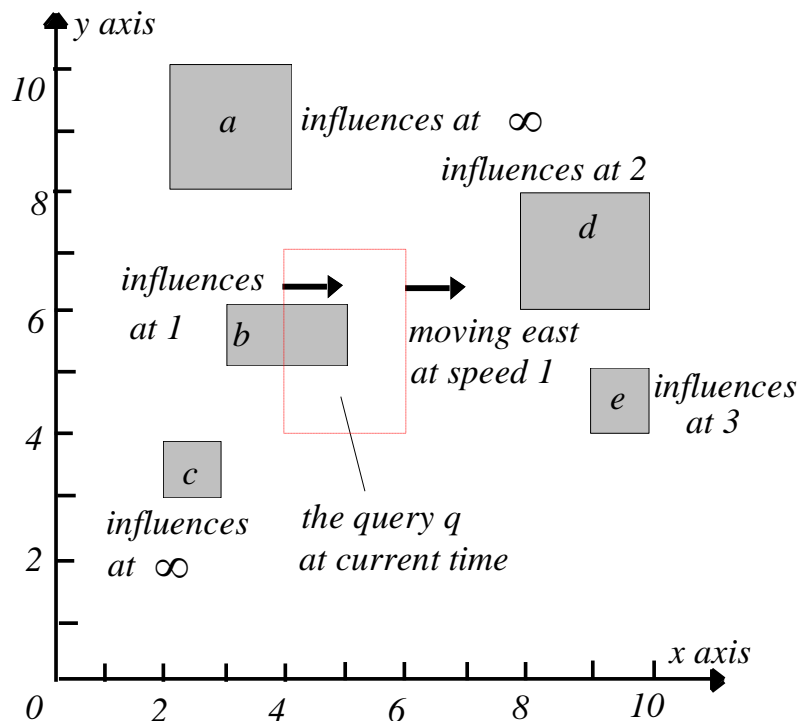
Depth-first (DF) and *Best-first* (BF) algorithms:



Report h and terminate

Reducing TP Window Queries to NN Search

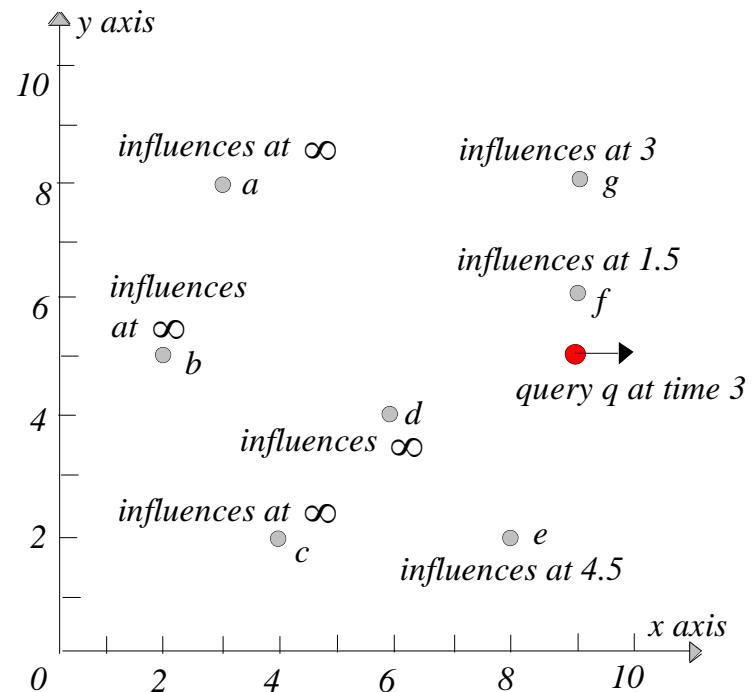
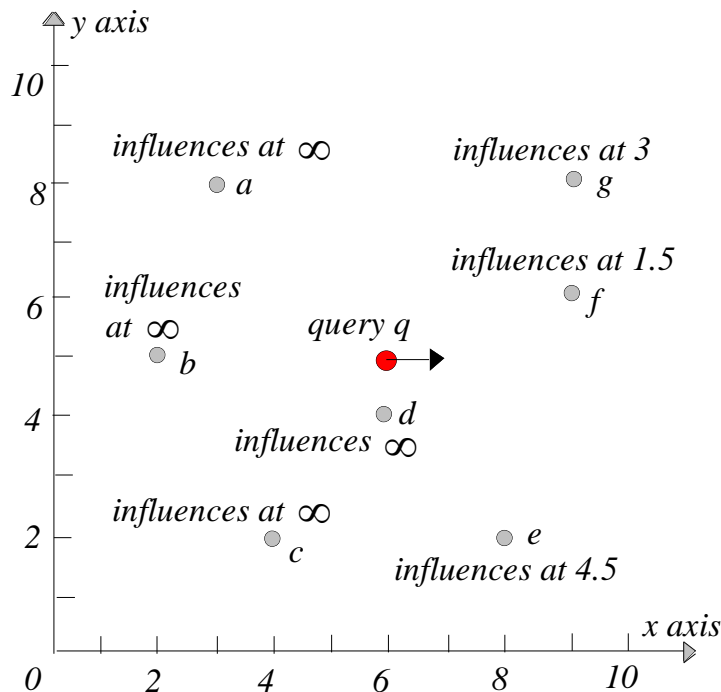
Definition: The influence time $T_{INF}(o, q)$ of a data object o indicates the time when o will change the current result of q .



- The object (**C** component) invalidating the current query result is the one with the smallest influence time (**T** component), i.e., a NN query using T_{INF} as the distance metric.

Reducing TP NN Queries to NN Search

Definition: The influence time $T_{INF}(o, q)$ of a data object o indicates the time when o will change the current result of q .

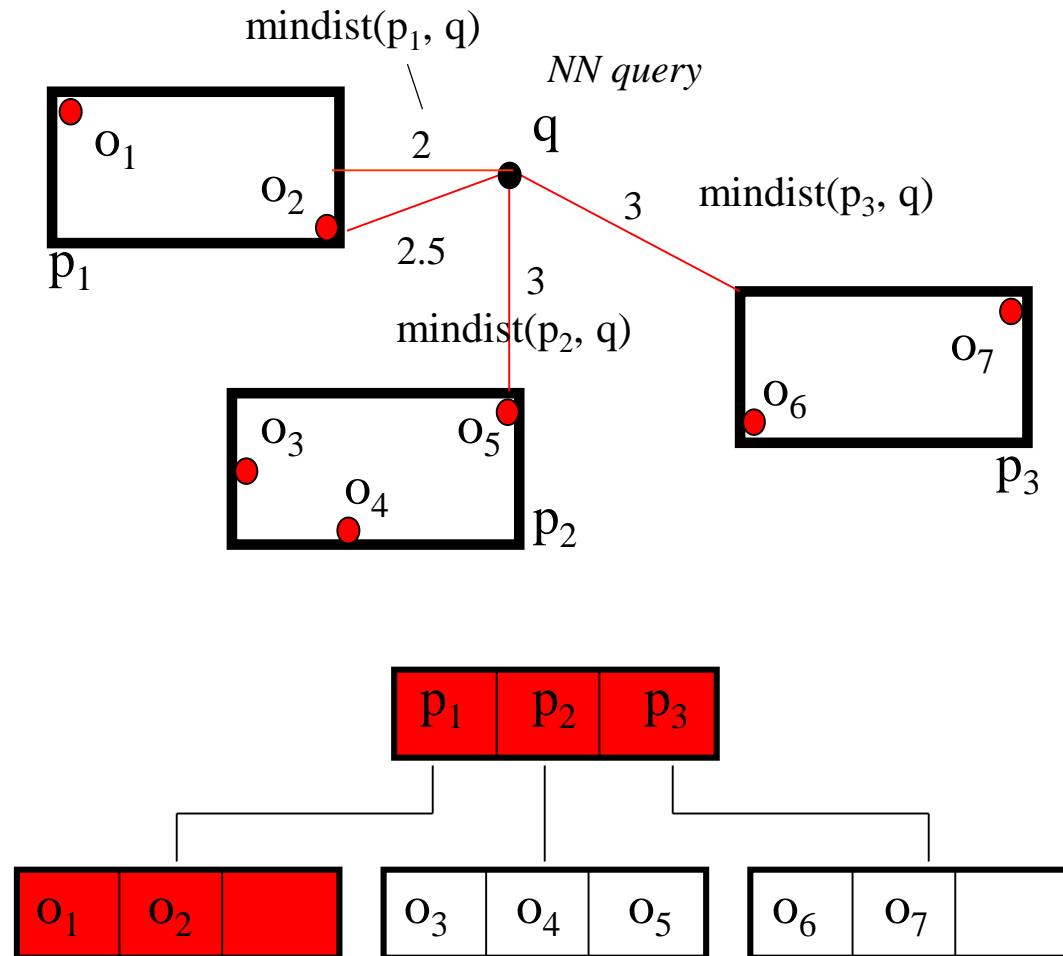


- The object (C component) invalidating the current query result is the one with the smallest influence time (T component), i.e., a NN query using T_{INF} as the distance metric.

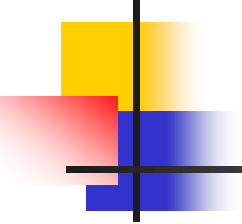
Nearest Neighbor (NN) Search with R-trees

The algorithm is based on the Branched and Bound framework.

We need 2 metrics: (i) $\text{dist}(o, q)$, and (ii) $\text{mindist}(E, q)$.

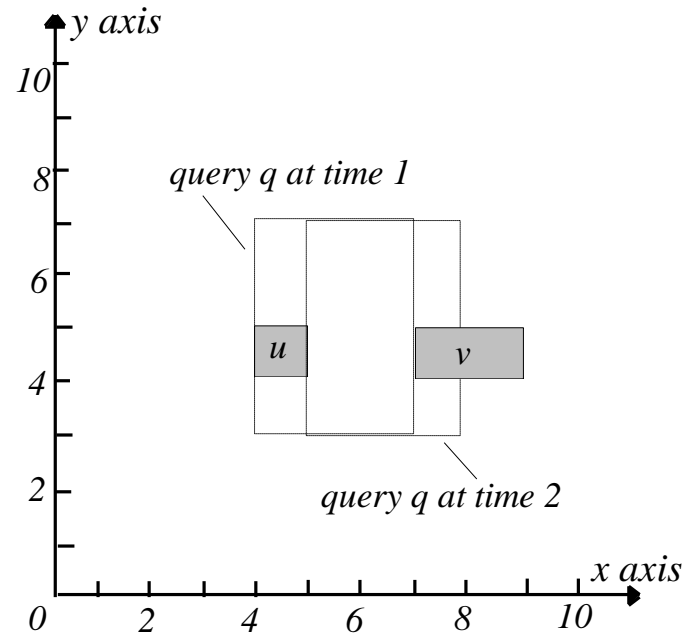
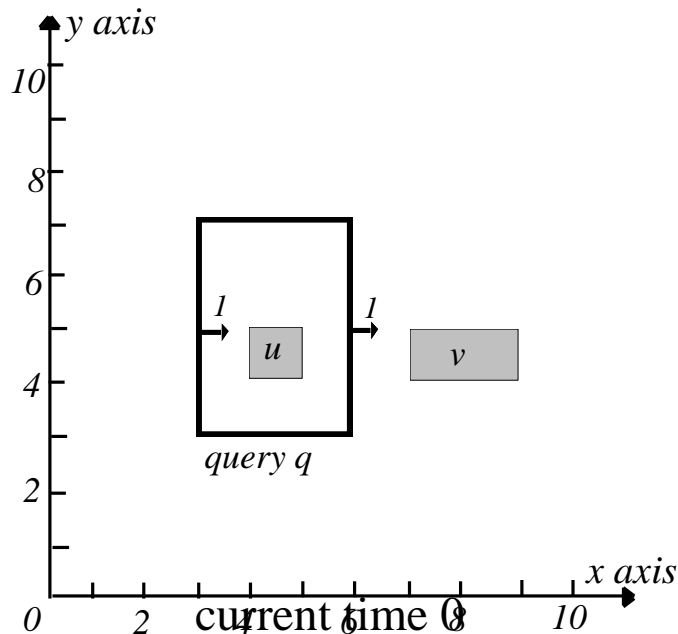


Processing TP Queries

- 
- Treating T_{INF} as the distance function, we may apply the branch and bound paradigm to answer TP queries.
 - Of course we must derive T_{INF} for specific query types.
 - Similar to $\text{mindist}(E, q)$ for the NN search, we also need $T_{\text{MININF}}(E, q)$, which is the minimum influence time $T_{INF}(o, q)$ among all objects o that can be in the subtree of E .

T_{INF} for TP Window Query

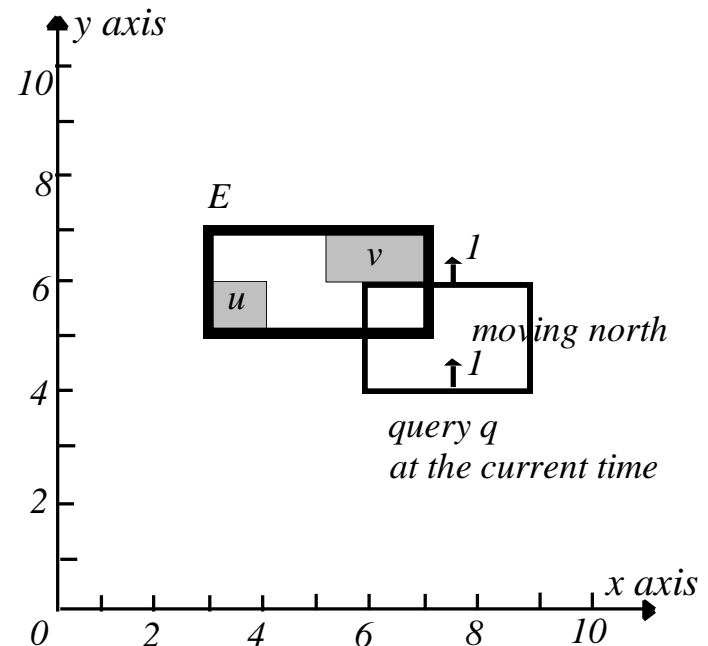
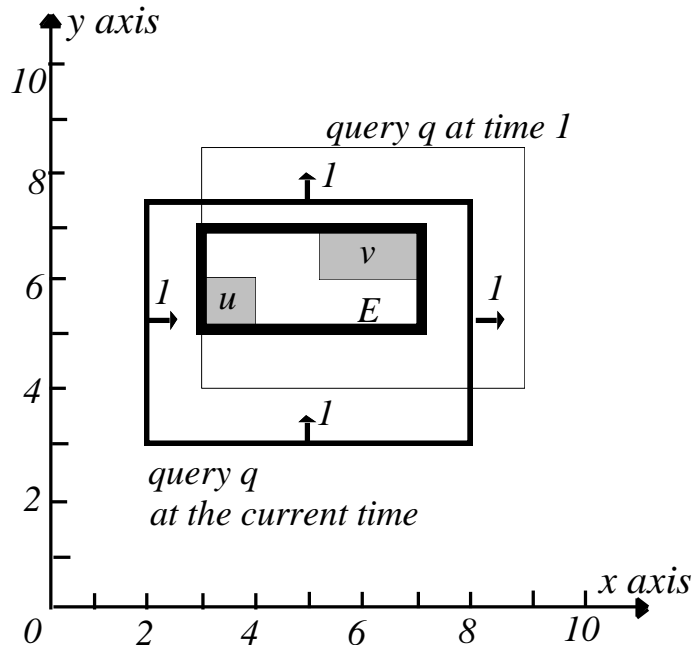
- If an object intersects query q now, its T_{INF} equals the earliest time it stops intersecting q in the future.
- If an object does not intersect query q now, its T_{INF} equals the earliest time it starts intersecting q in the future.



$$T_{INF}(u, q)=2, T_{INF}(v, q)=1$$

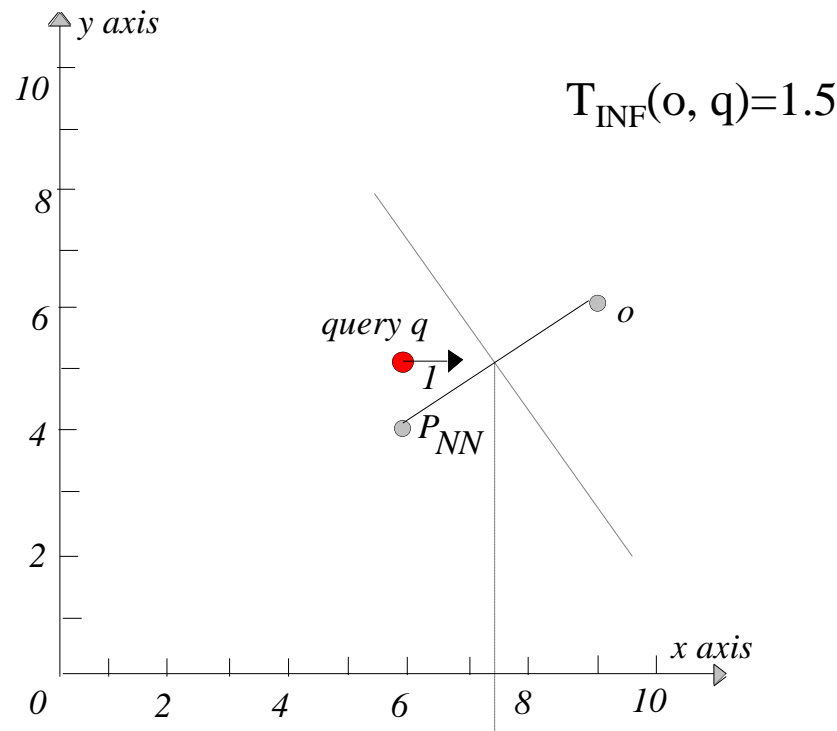
T_{MININF} for TP Window Query

- $T_{\text{MININF}}(E, q)$ equals the earliest future time E starts to intersect q if, at the current time
 - E does not intersect query q , or
 - E is contained query q
- $T_{\text{MININF}}(E, q)=0$ if E intersects (but is not contained in) q



T_{INF} for TP NN Query

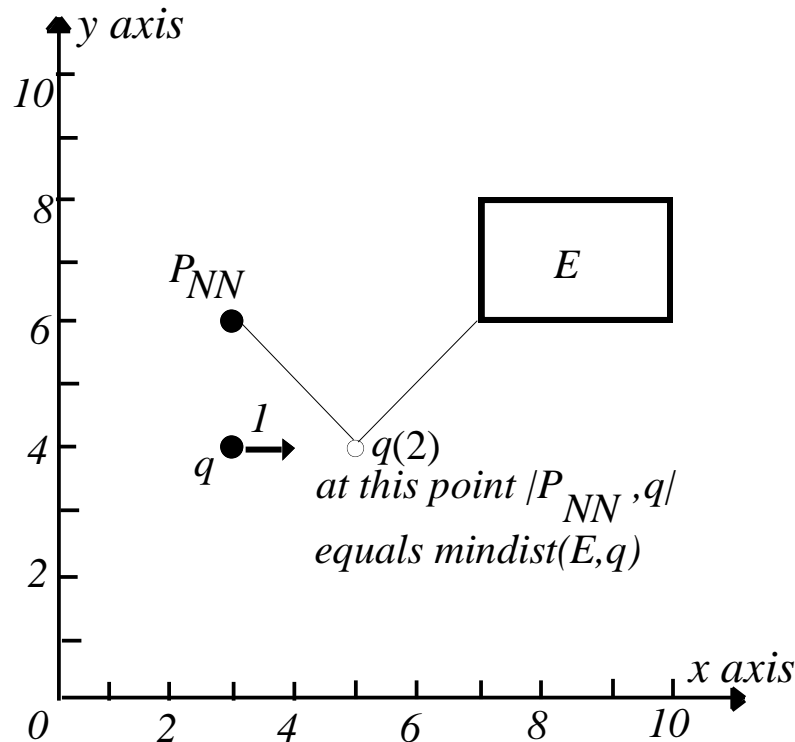
- Assume P_{NN} be the current nearest neighbor of query q ; T_{INF} of a data point o equals the time q crosses the perpendicular bisector of line segment $P_{NN}o$



- Note that T_{INF} for a TP NN query relies on the current result (i.e., the current NN), while T_{INF} for a TP window query does not.

T_{MININF} for TP NN Query

- $T_{\text{MININF}}(E, q)$ of a non-leaf entry E equals the time $\text{mindist}(E, q) = \text{dist}(q, P_{\text{NN}})$. As with T_{INF} , it depends on the current query result.



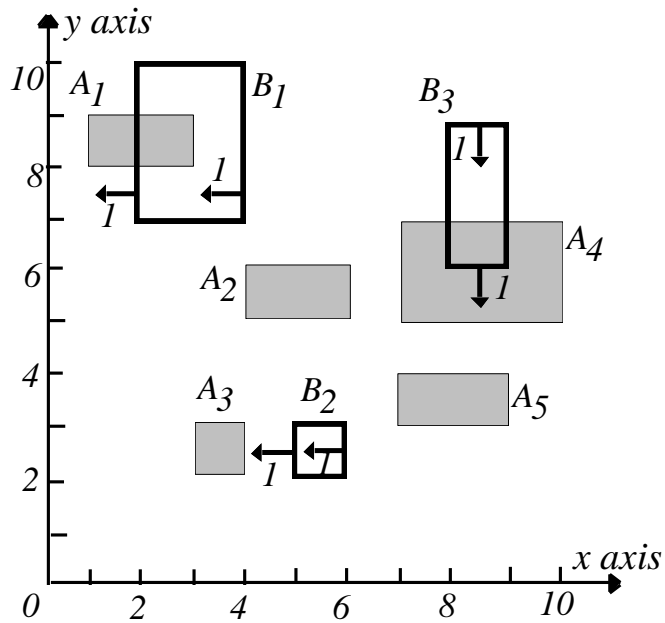


BaB Algorithms for TP Queries

- For those queries (e.g., TP window) where T_{MIN} and T_{MININF} do not depend on the current query result, the **T** and **C** components can be retrieved together with the **R** component in a *single* traversal of the index structure.
- For other queries (e.g., TP K-NN) where T_{MIN} and T_{MININF} depend on the current query result, the **T** and **C** components can be retrieved together with the **R** component in *separate* traversals of the index structure.

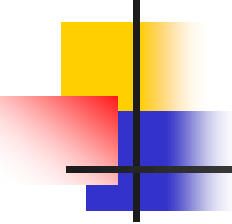
TP Spatial Join

- The TP spatial join is reduced to a closest pair query following the similar idea.



	A_1	A_2	A_3	A_4	A_5
B_1	2	∞	∞	∞	∞
B_2	∞	∞	1	∞	∞
B_3	∞	∞	∞	4	2

Processing Complex Queries

- 
- The proposed algorithms apply to other mobility combination of objects and queries as well (i.e., mobile objects and static queries, mobile objects and mobile queries).
 - TP queries are preliminary components for
 - Continuous queries
 - Earliest event queries



Conclusions

- The time-parameterized query can be integrated with any spatial query type to retrieve predictive information.
- Processing of TP queries can be reduced to NN search by defining appropriate distance functions.
- TP queries are preliminary building blocks for more complex queries.