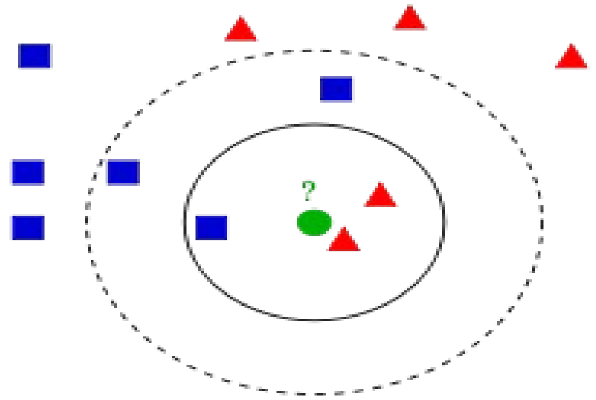
Эффективный поиск ближайших соседей в PostgreSQL

Oleg Bartunov, Teodor Sigaev
Sternberg Astronomical Institute,
Moscow University

Российские
интернет-технологии
2011

Knn-search: The problem

- What are interesting points near Royal Oak pub in Ottawa ?
- What are the closest events to the May 20, 2009 in Ottawa ?
- Similar images – feature extraction, Hamming distance
- Classification problem (major voting)
- ....
- GIS, Science (high-dimensional data)
- Data merging (de-duplication)



Knn-search: Existing solutions


```
knn=# select id, date, event from events order by date <-> '1957-10-04'::date asc limit 10;
   id   |    date    |                                 event
--------+------------+-----------------------------------------------------------------
  58137 | 1957-10-04 | U.S.S.R. launches Sputnik I, 1st artificial Earth satellite
  58136 | 1957-10-04 | "Leave It to Beaver," debuts on CBS
 117062 | 1957-10-04 | Gregory T Linteris, Demarest, New Jersey, astronaut, sk: STS 83
 117061 | 1957-10-04 | Christina Smith, born in Miami, Florida, playmate, Mar, 1978
 102670 | 1957-10-05 | Larry Saumell, jockey
  31456 | 1957-10-03 | Willy Brandt elected mayor of West Berlin
  58291 | 1957-10-05 | 12th Ryder Cup: Britain-Ireland, 7 -4 at Lindrick GC, England
  58290 | 1957-10-05 | 11th NHL All-Star Game: All-Stars beat Montreal 5-3 at Montreal
  58292 | 1957-10-05 | Yugoslav dissident Milovan Djilos sentenced to 7 years
 102669 | 1957-10-05 | Jeanne Evert, tennis player, Chris' sister
(10 rows)

Time: 115.548 ms
```

- Very inefficient:
  - Full table scan, btree index on date won't help.
  - Sort full table

Knn-search: Existing solutions

- Find 10 closest points in table T
  MS SQL the best solution: Number is a table with integers
  http://blogs.msdn.com/b/isaac/archive/2008/10/23/nearest-neighbors.aspx

```
DECLARE @start FLOAT = 1000;
WITH NearestPoints AS
(
   SELECT TOP(1) WITH TIES *,  T.g.STDistance(@x) AS dist
   FROM Numbers JOIN T WITH(INDEX(spatial_index))
   ON T.g.STDistance(@x) < @start*POWER(2,Numbers.n)
   ORDER BY n
)
SELECT TOP(10) * FROM NearestPoints
ORDER BY n, dist
```

Denali supports knn, but still lose the above trick :)
http://www.sqlskills.com/BLOGS/BOBB/post/The-nearest-neighbor-optimization-in-SQL-Server-Denali.aspx

Knn-search: Existing solutions

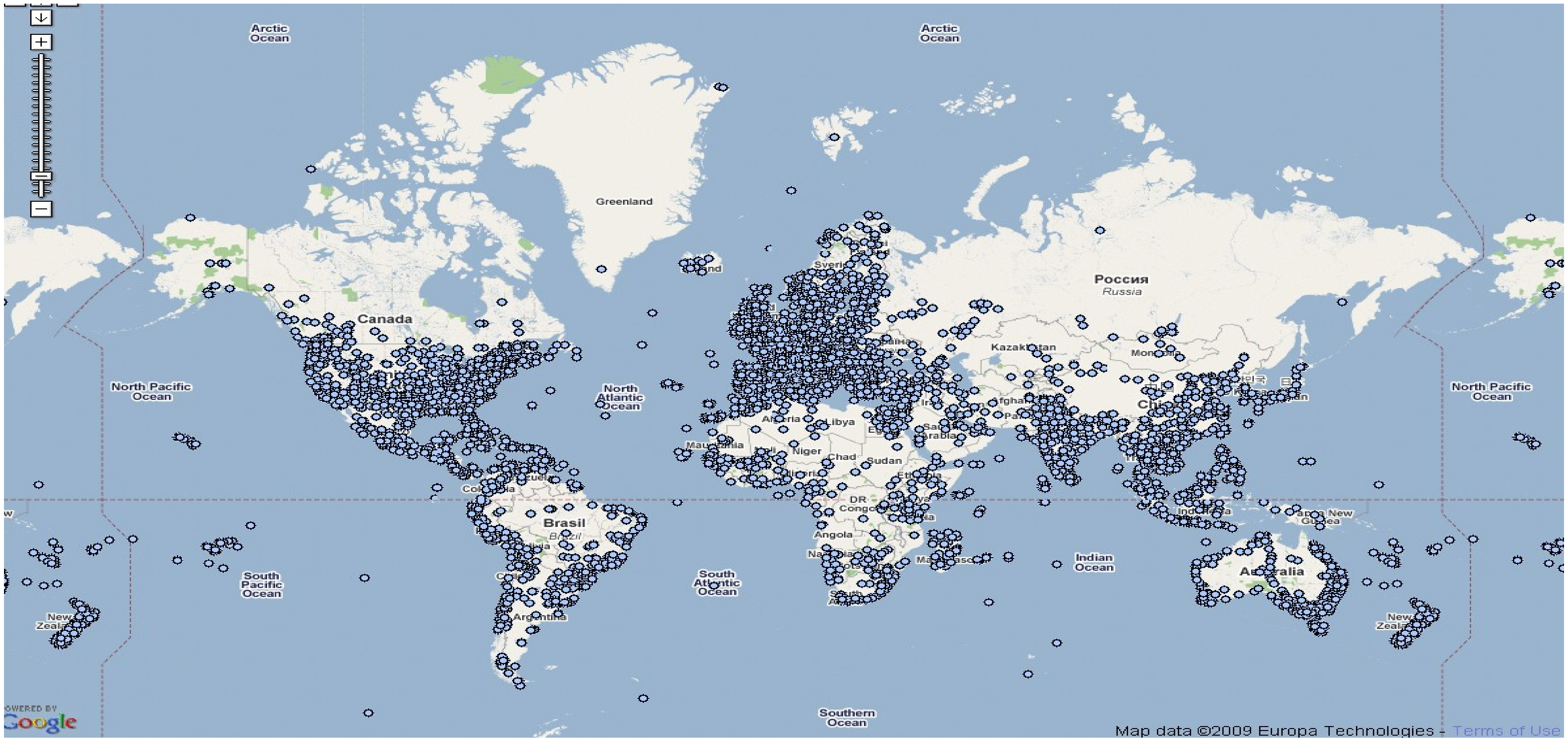
- Oracle Spatial has SDO_NN function (only 2 dimensions !) http://download.oracle.com/docs/cd/B19306_01/appdev.102/b14255/sdo_operat.htm#i78067

```
SELECT /*+ INDEX(c cola_spatial_idx) */
 c.mkt_id, c.name  FROM cola_markets c  WHERE SDO_NN(c.shape,
  sdo_geometry(2001, NULL, sdo_point_type(10,7,NULL), NULL,
  NULL),  'sdo_num_res=2') = 'TRUE';
```

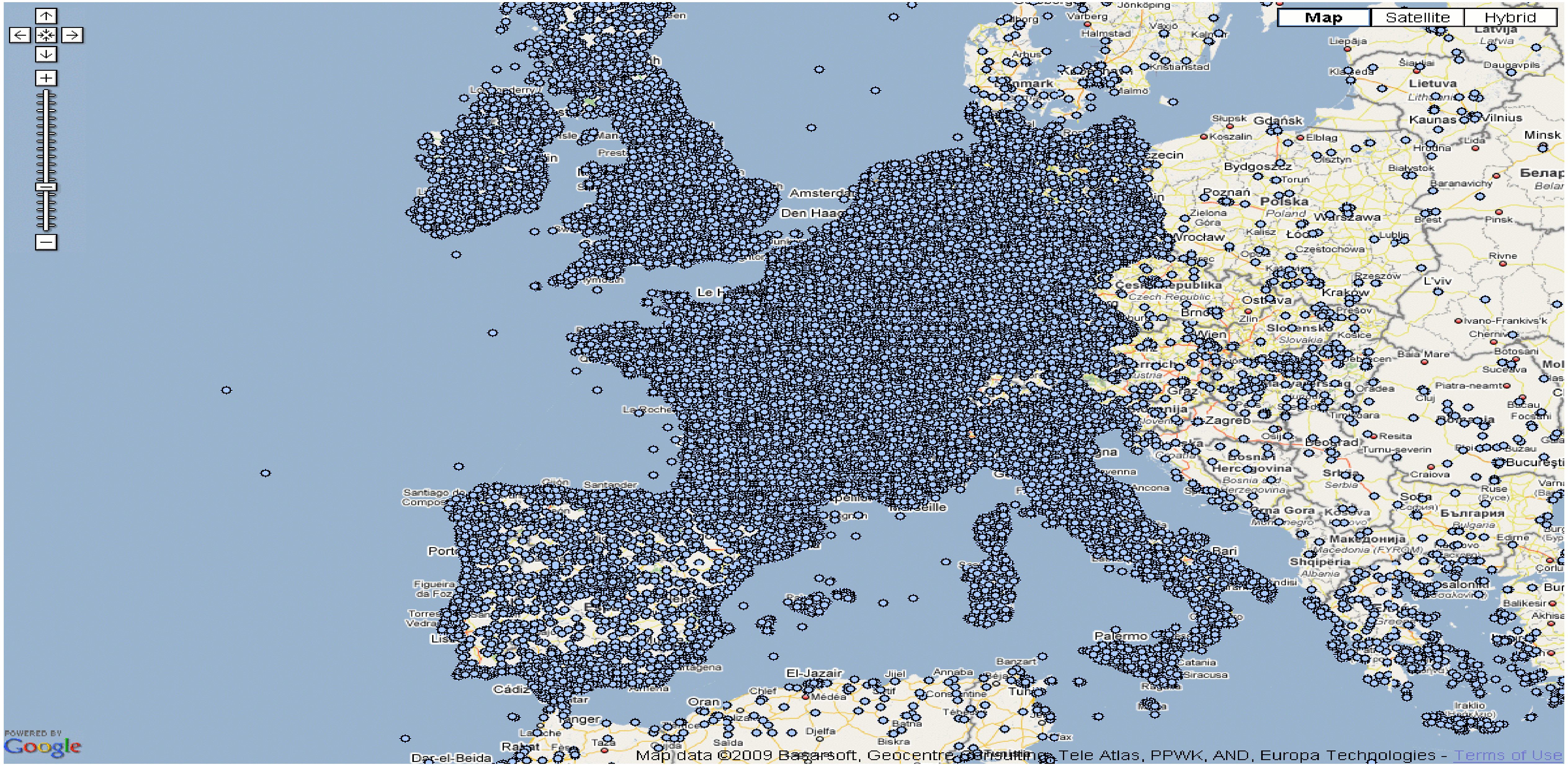
Knn-search: Existing solutions

- Traditional way to speedup query
  - Use indexes - very inefficient (no search query !)
    - Scan full index
    - Full table scan, but in random order !
    - Sort full table
    - Better not to use index at all !
  - Constrain data space (range search)
    - Incremental search → too many queries
    - Need to know in advance size of neighbourhood, how ? 1Km is ok for Paris, but too small for Siberia
    - Maintain 'density map' ?

What's a neighbourhood ?



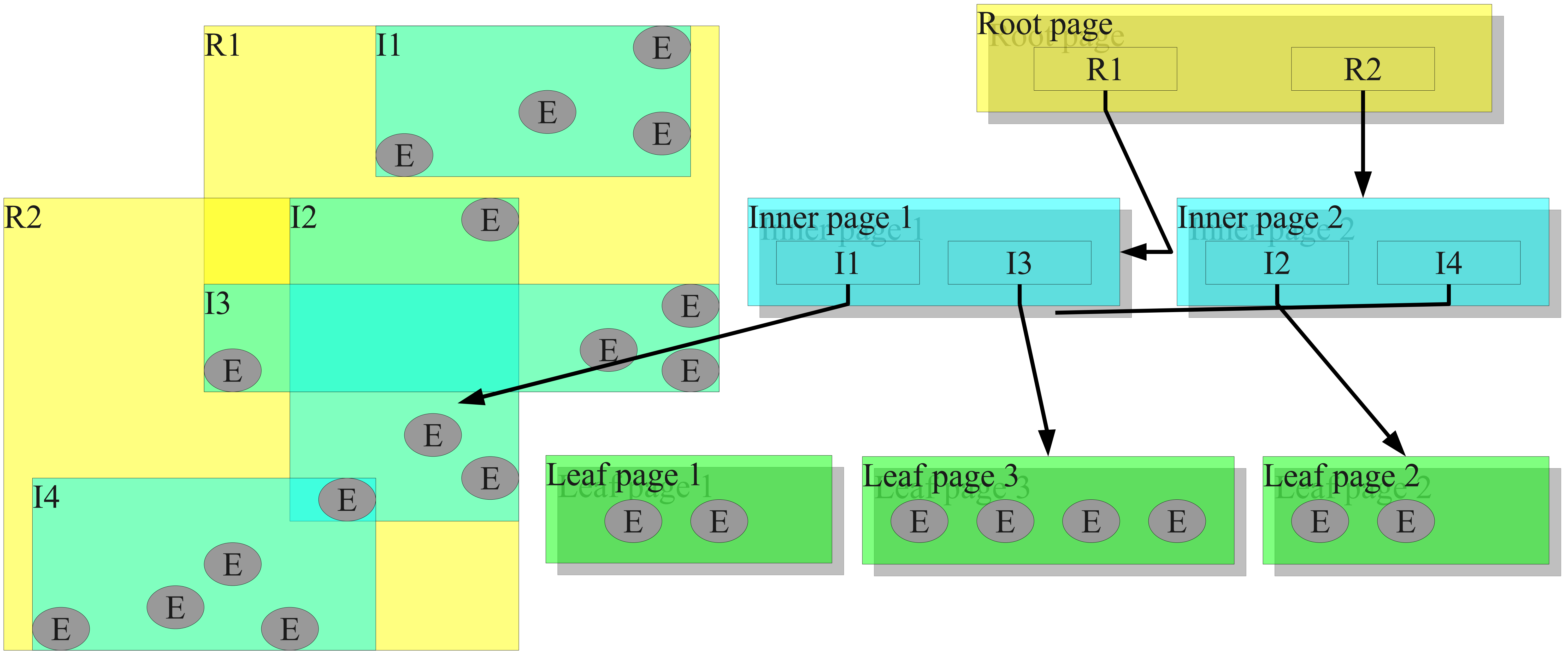
Zoom


Map
Satellite
Hybrid
Amsterdam
Den Haag
Gdańsk
Bydgoszcz
Poznań
Polska
Warszawa
Wrocław
Kraków
Lietuva
Kaunas
Vilnius
Minsk
Zagreb
Bosna i Hercegovina
Srbija
България
Shqipëria
Palermo
Bari
Marseille
Porto
Cádiz
Oran
El-Jazair
POWERED BY Google
Map data ©2009 Basarsoft, Geocentre Consulting, Tele Atlas, PPWK, AND, Europa Technologies - Terms of Use

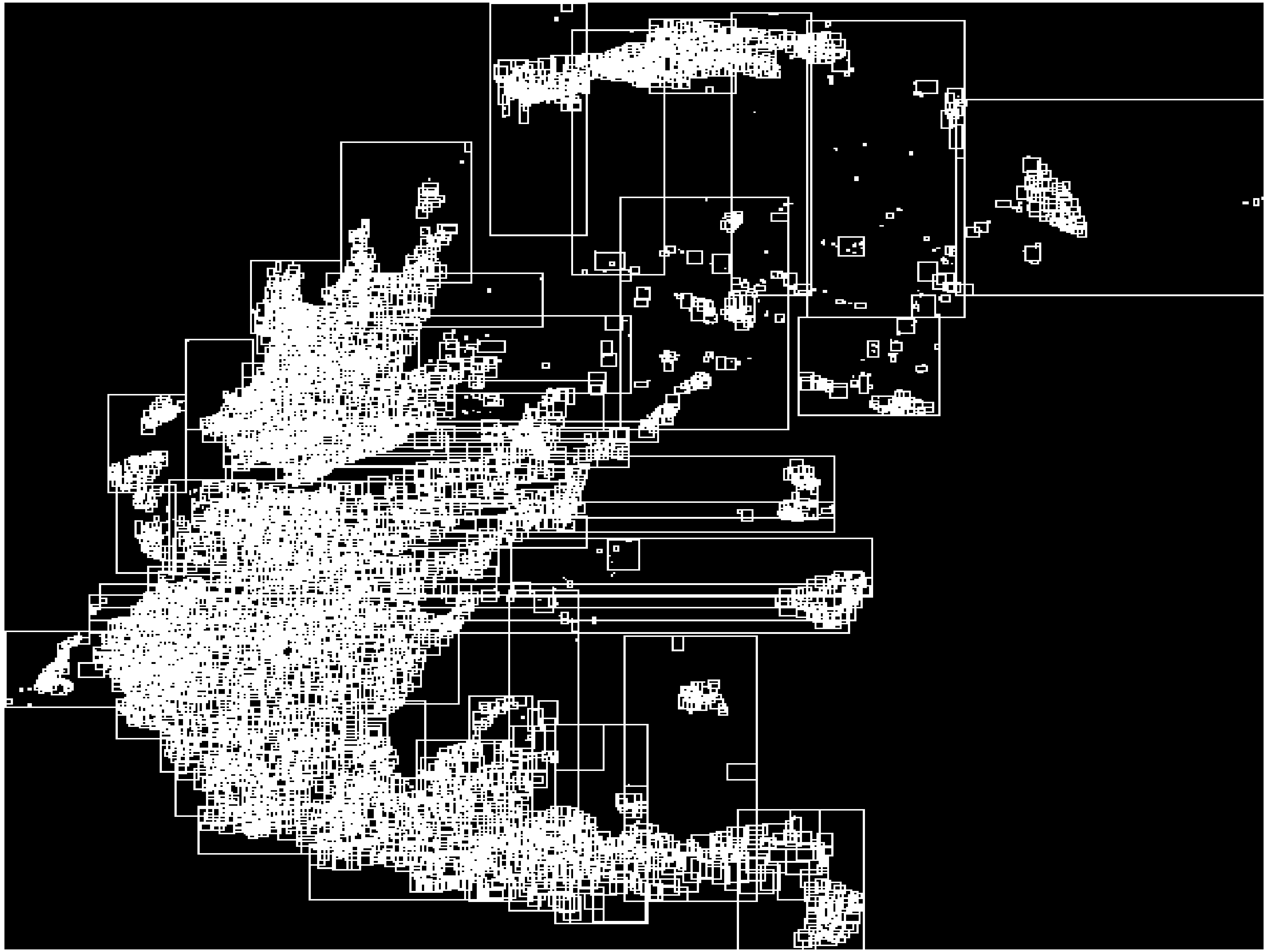
Knn-search: What do we want !

- We want to avoid full table scan – read only <right> tuples
  - So, we need index
- We want to avoid sorting – read <right> tuples in <right> order
  - So, we need special strategy to traverse index
- We want to support tuples visibility
  - So, we should be able to resume index traverse

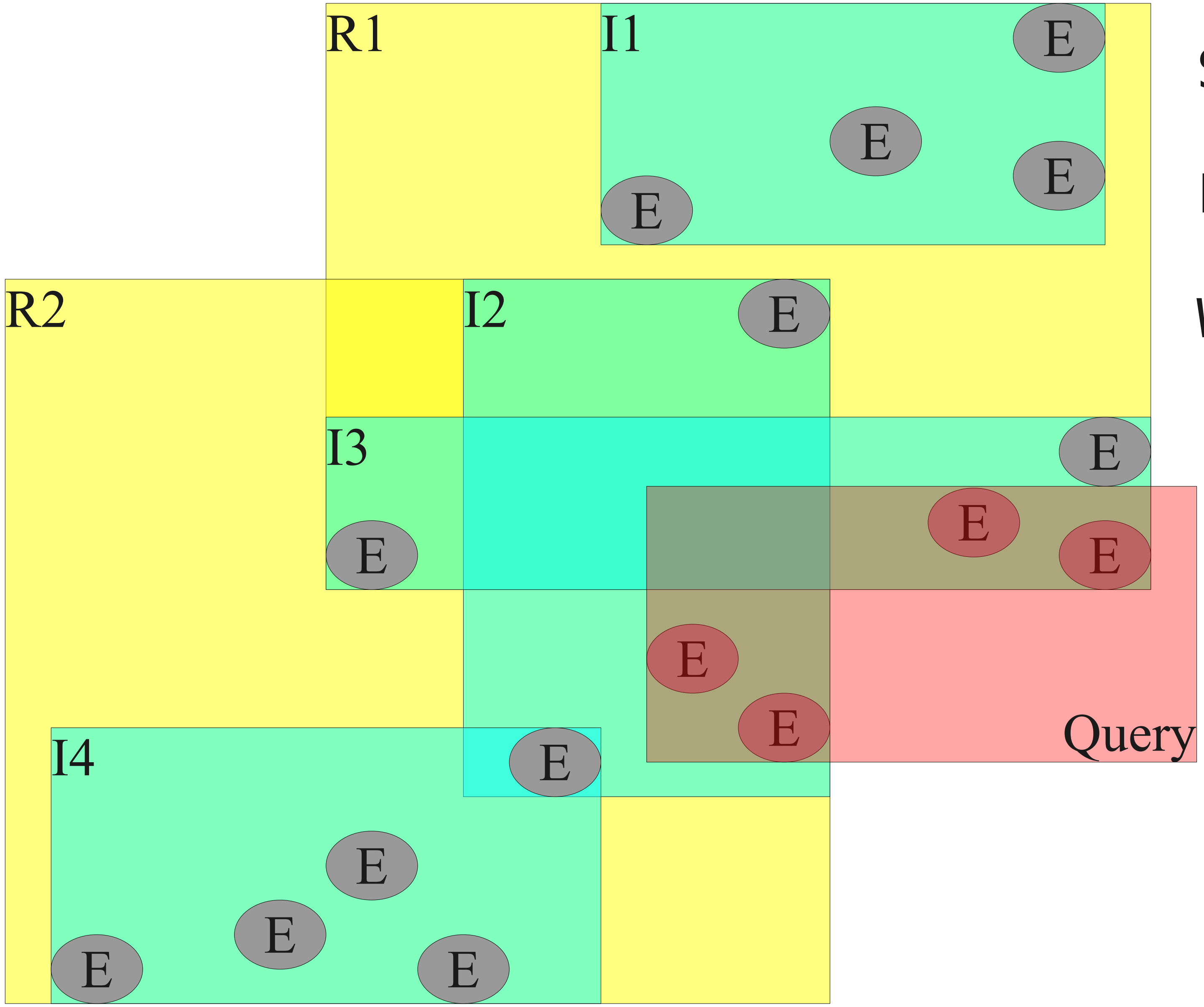
R-tree index


R1
I1
R2
I2
I3
I4
E
Root page
R1
R2
Inner page 1
I1
I3
Inner page 2
I2
I4
Leaf page 1
Leaf page 3
Leaf page 2

- Visualization of R-tree index using Gevel.
- Greece
  (data from rtreeportal.org)



R-tree index




```
SELECT
   *
FROM
   events
WHERE
   events.coord <@ 'QUERY';
```

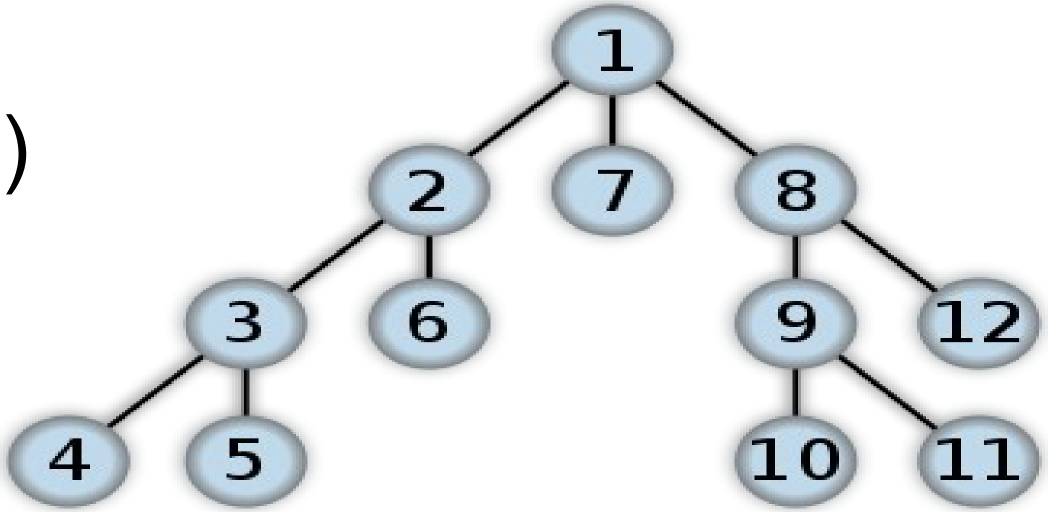
- Root page: R1, R2 keys
- Inner pages: I3, I2 keys
- Leaf pages: 4 points

- Very efficient for Search !

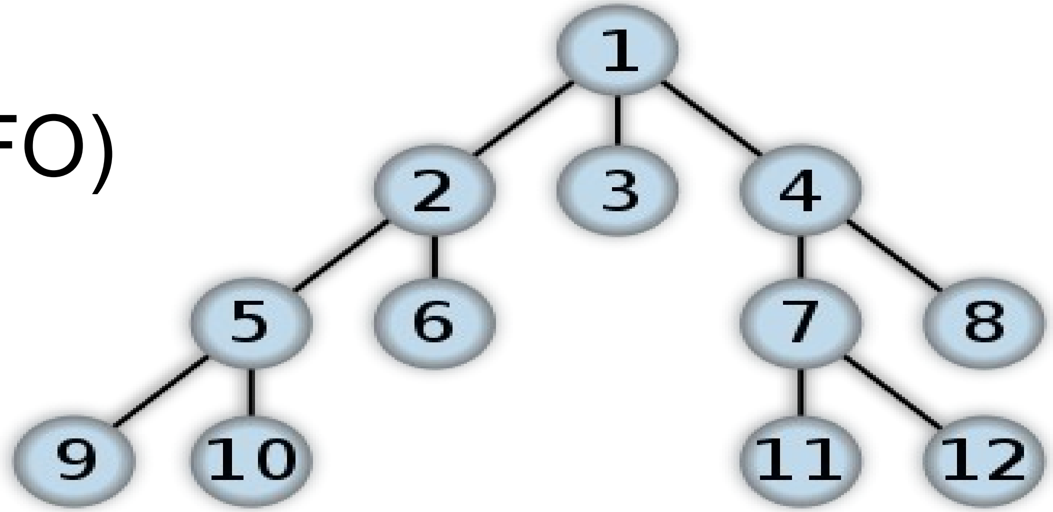
Knn-search: Index traverse

- Depth First Search (stack, LIFO)

  R-tree search



- Breadth First Search (queue, FIFO)



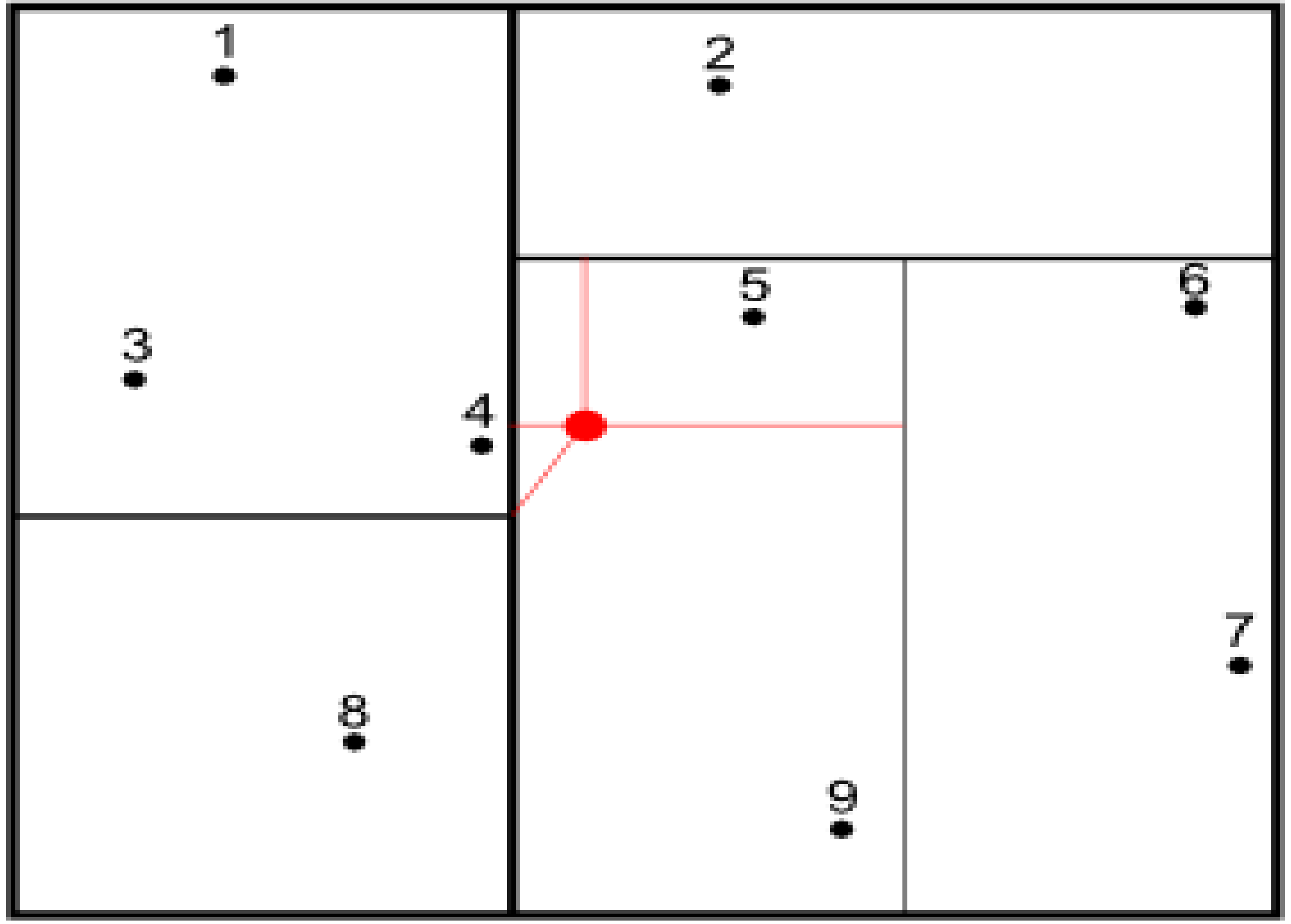
- Both strategies are not good for us – full index scan

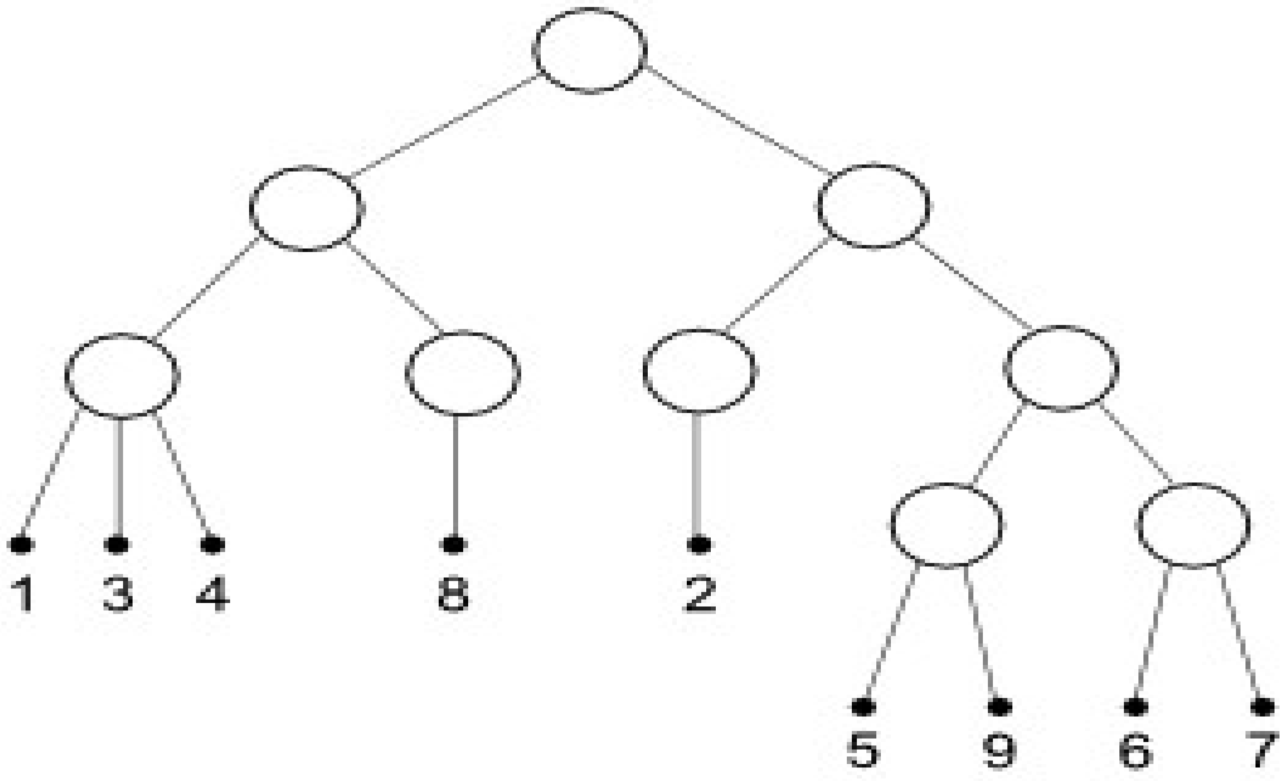
Knn-search: Index traverse

- Best First Search (PQ, priority queue). Maintain order of items in PQ according their distance from given point
  - Distance to MBR (rectangle for Rtree) for internal pages –   minimum distance of all items in that MBR
  - Distance = 0 for MBR with given point
  - Distance to point for leaf pages
- Each time we extract point from PQ we output it – it is next closest point ! If we extract rectangle, we expand it by pushing their children (rectangles and points) into the queue.
- We traverse index by visiting only interesting nodes !

Knn-search: Index traverse

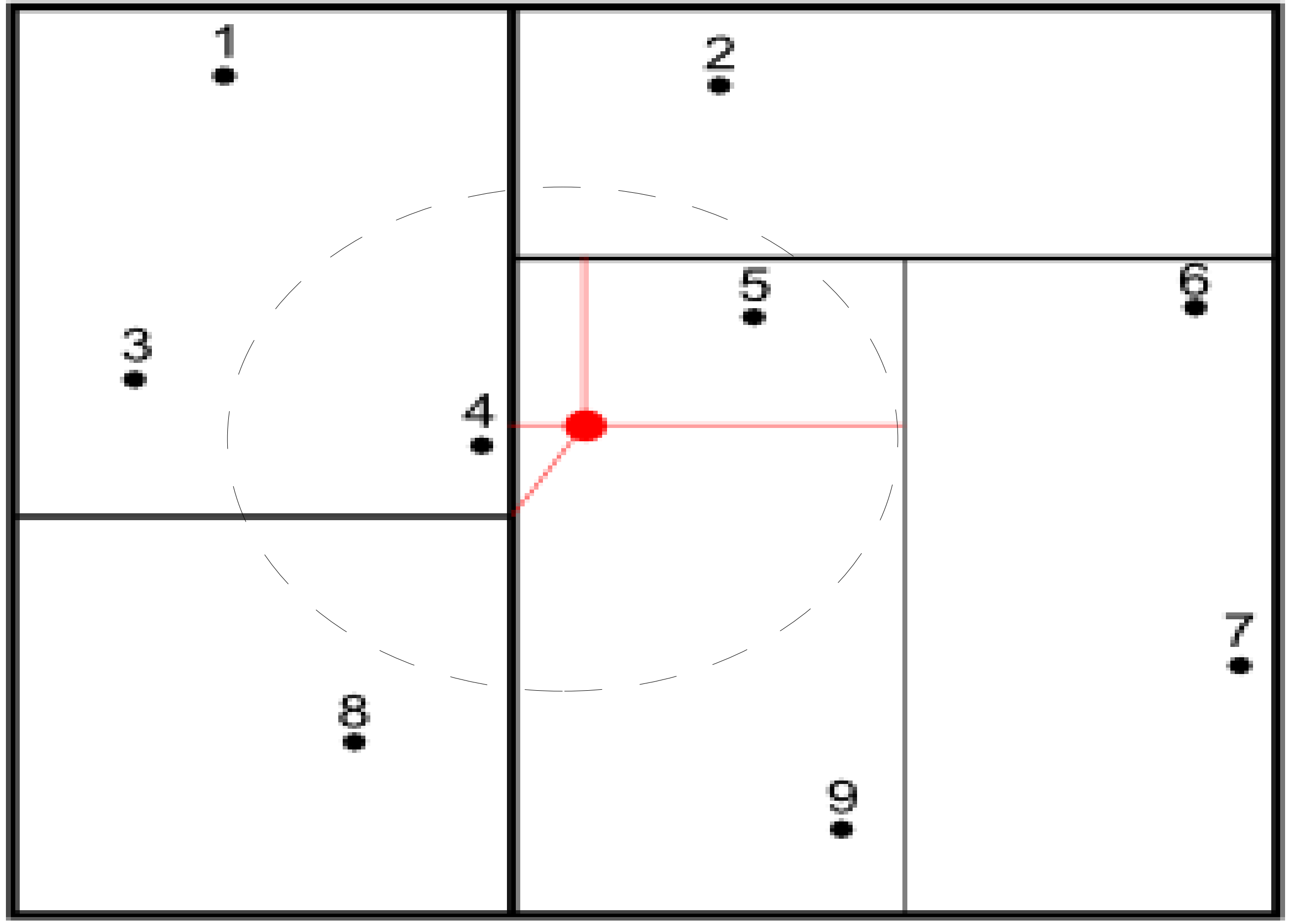
- Simple example – non-overlapped partitioning





Knn-search: Index traverse

- Simple example – non-overlapped partitioning



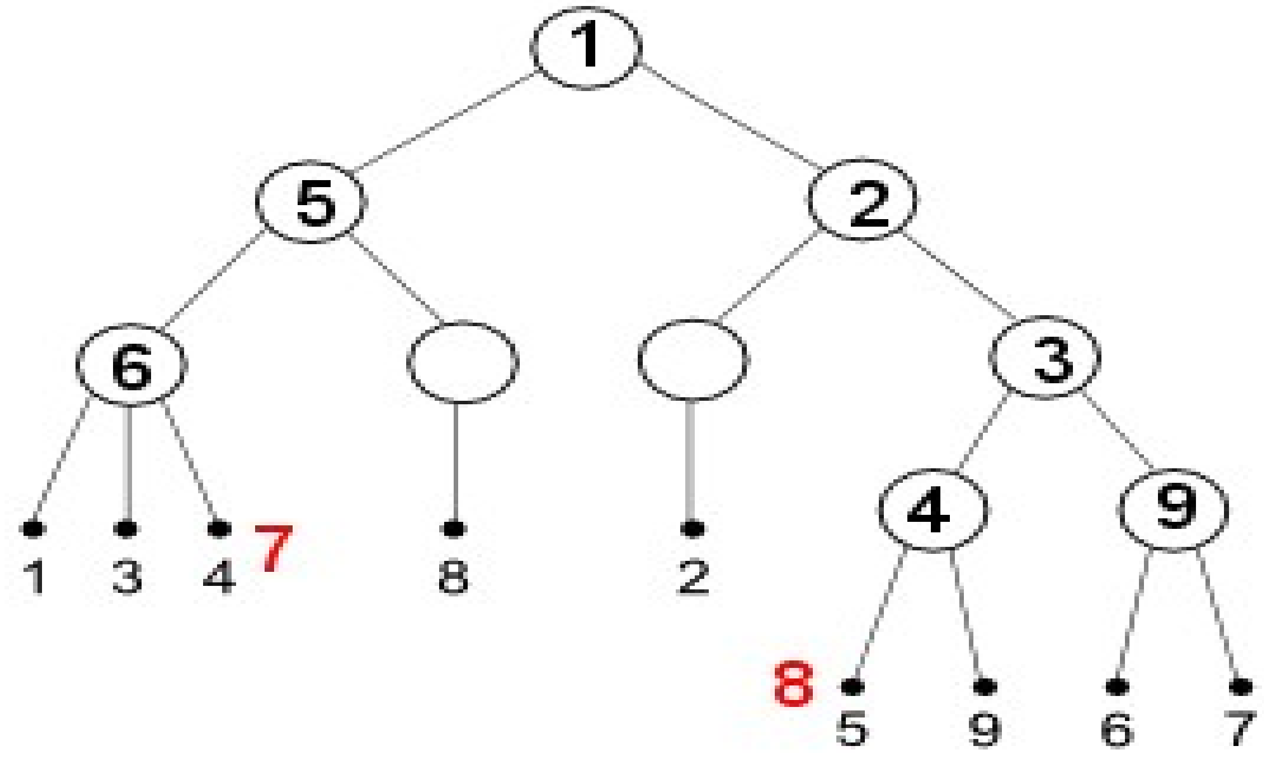
- Priority Queue
  - 1: {1,2,3,4,5,6,7,8,9}
  - 2: {2,5,6,7,9}, {1,3,4,8}
  - 3: {5,6,7,9}, {1,3,4,8}, {2}
  - 4: {5,9}, {1,3,4,8}, {2}, {6,7}
  - 5: {1,3,4,8}, 5, {2}, {6,7}, 9
  - 6: {1,3,4}, {8}, 5, {2}, {6,7}, 9
  - 7: 4, {8}, 5, {2}, {6,7}, 3, 1, 9

  we can output 4 without visit other rectangles !

  - 8: 5, {2}, {6,7}, 3, 8, 1, 9
  - 9: {6,7}, 3, 2, 8, 1, 9
  - 10: 3, 2, 8, 1, 9, 6, 7

Knn-search: Index traverse

- Simple example – non-overlapped partitioning



- Priority Queue
  - 1: {1,2,3,4,5,6,7,8,9}
  - 2: {2,5,6,7,9}, {1,3,4,8}
  - 3: {5,6,7,9}, {1,3,4,8}, {2}
  - 4: {5,9}, {1,3,4,8}, {2}, {6,7}
  - 5: {1,3,4,8}, 5, {2}, {6,7}, 9
  - 6: {1,3,4}, {8}, 5, {2}, {6,7}, 9
  - 7: 4, {8}, 5, {2}, {6,7}, 3, 1, 9
  - 8: 5, {2}, {6,7}, 3, 8, 1, 9

Knn-search: Performance

- SEQ (no index) – base performance
  - Sequentually read full table + Sort full table (can be very bad, sort_mem !)
- DFS – very bad !
  - Full index scan + Random read full table + Sort full table
- BFS – the best for small k !
  - Partial index scan + Random read k-records
    - T(index scan) ~ Height of Search tree ~ log(n)
  - Performance win BFS/SEQ ~ Nrelpages/k, for small k. The more rows, the more benefit !
  - Can still win even for k=n (for large tables) - no sort !

Knn-search: What do we want !

- + We want to avoid full table scan – read only <right> tuples
  - So, we need index
- + We want to avoid sorting – read <right> tuples in <right> order
  - So, we need special strategy to traverse index
- + We want to support tuples visibility
  - So, we should be able to resume index traverse
- We want to support many data types
  - So, we need to modify GiST

Knn-search: modify GiST

- GiST – Generalized Search Tree, provides
  - API to build custom disk-based search trees (any tree, where key of internal page is a Union of keys on children pages)
  - Recovery and Concurrency
  - Data type and query extendability
- GiST is widely used in GIS (PostGIS), text search,...
- Current strategy of search tree traverse is DFS
  - Not good for knn-search
  - We need to add Best First Search strategy for knn-search
  - Retain API compatibility

Knn-search: syntax

- Knn-query uses ORDER BY clause

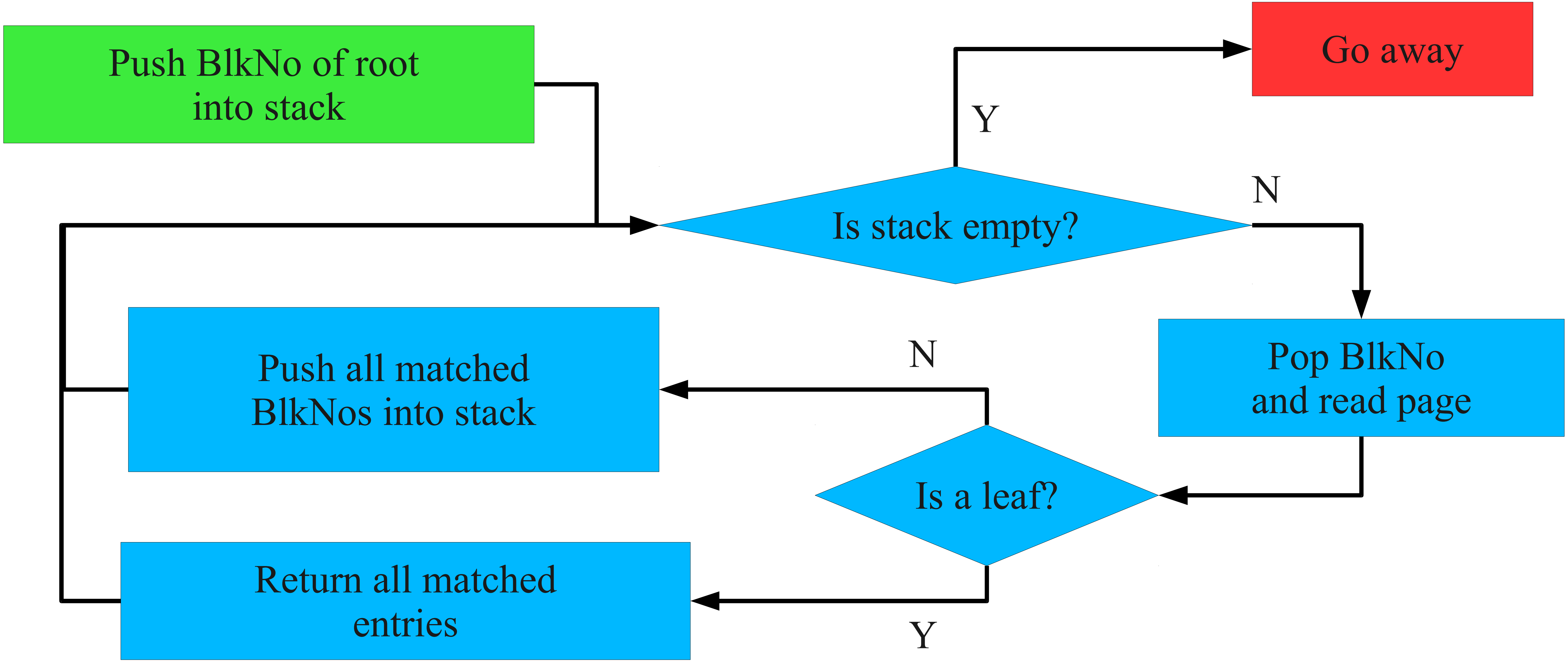
```
SELECT … FROM … WHERE …
ORDER BY p < - > '(0.,0.)'::point
 LIMIT k;
```

< - > - distance operator, should be provided for data type

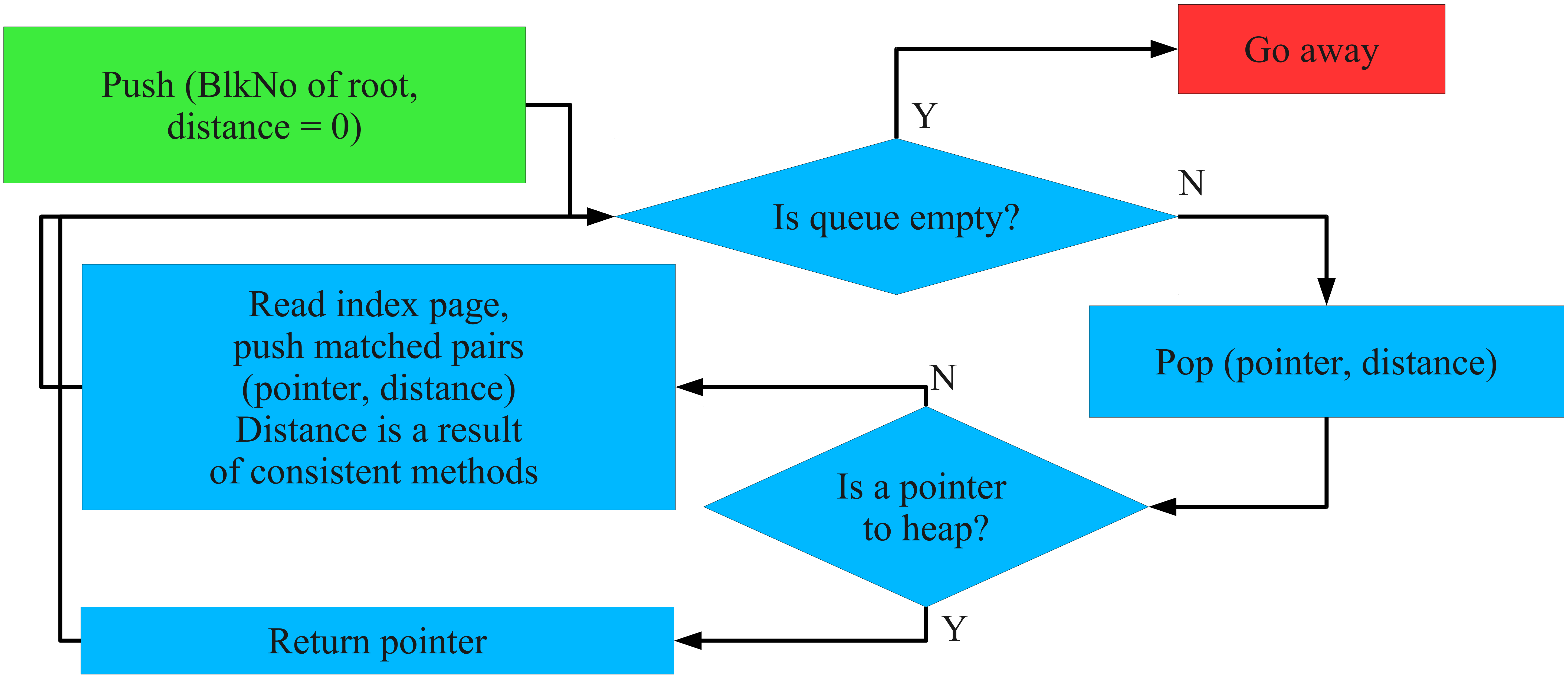
GiST interface

- compress/decompress
- same
- union
- penalty
- picksplit
- consistent
- **distance** - is needed if the operator class wishes to support ordered scans (nearest-neighbor searches)

Depth First Search

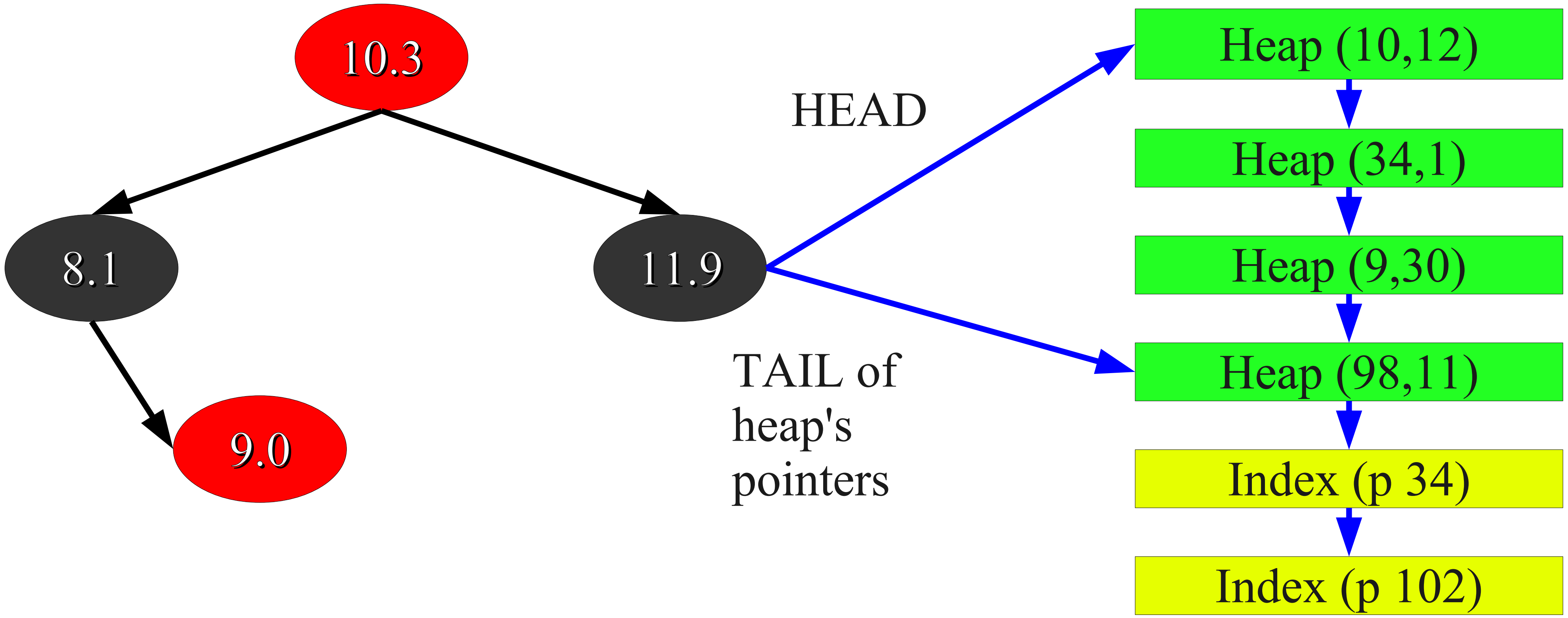


KNN-search: Priority Queue



GiST: Technical details

- Priority queue implemented as a RB-tree (Red-Black tree)
- Each node of RB-tree contains a list of pointers - pointers to internal pages follow pointers to heap.



GiST: Technical details

Depth First Search

```
push Stack, Root;
While Stack {
   If p is heap {
             output p;
   else {
             children = get_children(p);
             push Stack, children;
        }
}
```

Best First Search

```
push PQ, Root;
While PQ {
   If p is heap {
             output p;
   else {
             Children = get_children(p);
             push PQ, children;
        }
}
```

- For non-knn search all distances are zero, so PQ => Stack and BFS => DFS
- We can use only one strategy for both – normal search and knn-search !

Knn-search: What do we want !

- **+** We want to avoid full table scan – read only <right> tuples
  - So, we need index
- **+** We want to avoid sorting – read <right> tuples in <right> order
  - So, we need special strategy to traverse index
- **+** We want to support tuples visibility
  - So, we should be able to resume index traverse
- **+** We want to support many data types
  - So, we need to modify GiST

Knn-search: Examples

- Synthetic data – randomly distributed points

```
create table qq ( id serial, p point, s int4);

insert into qq (p,s)  select point( p.lat, p.long), (random()*1000)::int
from ( select  (0.5-random())*180 as lat, random()*360 as long
       from ( select generate_series(1,1000000) ) as t
) as p;
create index qq_p_s_idx on qq using gist(p);
analyze qq;
```

- Query – find k-closest points to (0,0)

```
set enable_indexscan=on|off;
explain (analyze on, buffers on)
  select * from qq order by (p <-> '(0,0)') asc limit 10;
```

Knn-search: Examples

- postgresql.conf:

```
shared_buffers = 512MB #32MB
work_mem = 32MB #1MB
maintenance_work_mem = 256MB #16MB
checkpoint_segments = 16
effective_cache_size = 1GB #128MB
```

- Index statistics (n=1000,000)

```
Number of levels:          3
Number of pages:           8787
Number of leaf pages:      8704
Number of tuples:          1008786
Number of invalid tuples:  0
Number of leaf tuples:     1000000
Total size of tuples:      44492028 bytes
Total size of leaf tuples: 44104448 bytes
Total size of index:       71983104 bytes
```

Knn-search: Examples

k=1, n=1,000,000

```
Limit  (cost=24853.00..24853.00 rows=1 width=24) (actual time=469.129..469.130
rows=1 loops=1)
   Buffers: shared hit=7353
   ->  Sort  (cost=24853.00..27353.00 rows=1000000 width=24) (actual
time=469.128..469.128 rows=1 loops=1)
         Sort Key: ((p <-> '(0,0)'::point))
         Sort Method:  top-N heapsort  Memory: 25kB
         Buffers: shared hit=7353
         ->  Seq Scan on qq  (cost=0.00..19853.00 rows=1000000 width=24)
(actual time=0.007..241.539 rows=1000000 loops=1)
               Buffers: shared hit=7353
 Total runtime: 469.150 ms

 Limit  (cost=0.00..0.08 rows=1 width=24) (actual time=0.104..0.104
rows=1 loops=1)
   Buffers: shared hit=4
   ->  Index Scan using qq_p_idx on qq  (cost=0.00..82060.60 rows=1000000
width=24) (actual time=0.104..0.104 rows=1 loops=1)
         Sort Cond: (p <-> '(0,0)'::point)
         Buffers: shared hit=4
 Total runtime: 0.117 ms       4000 times faster !
--------------------------
```

Knn-search: Examples

n=1000,000

k	:hit	:knn	: seq	:sortmem
1	:4	:0.117	:469.150	: 25
10	:17	:0.289	:471.735	: 25
100	:118	:0.872	:468.244	: 32
1000	:1099	:7.107	:473.840	: 127
10000	:10234	:31.629	:525.557	: 1550
100000	:101159	:321.182	:994.925	: 13957

Knn-search: Examples

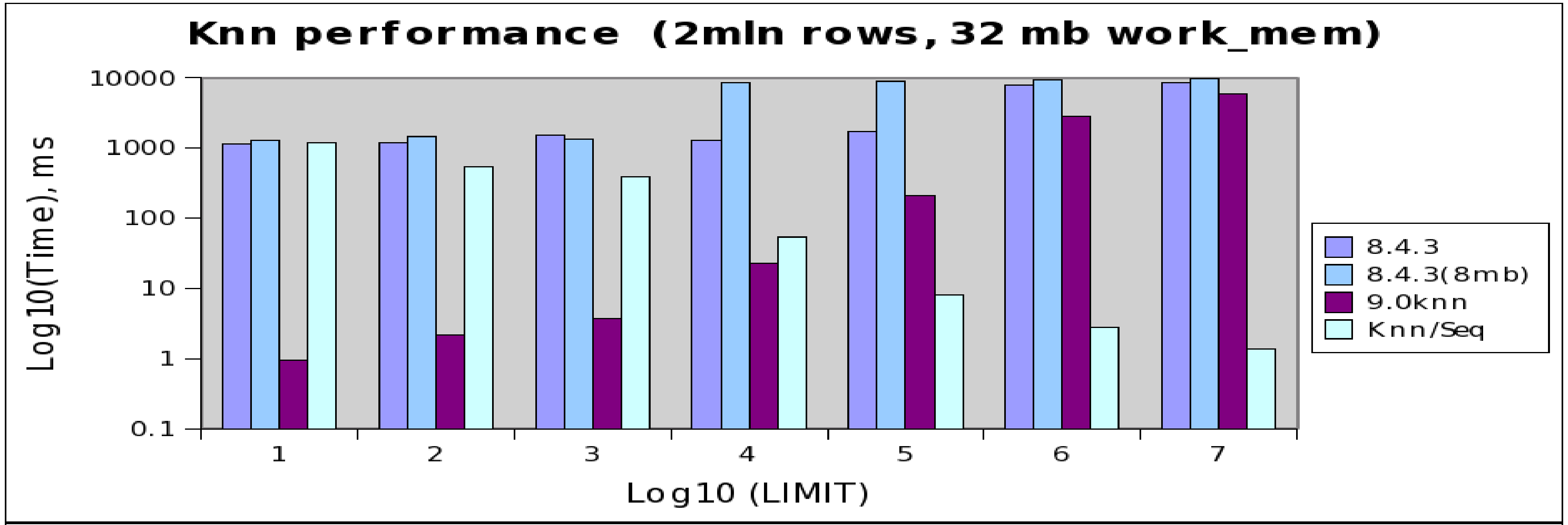
n=10,000

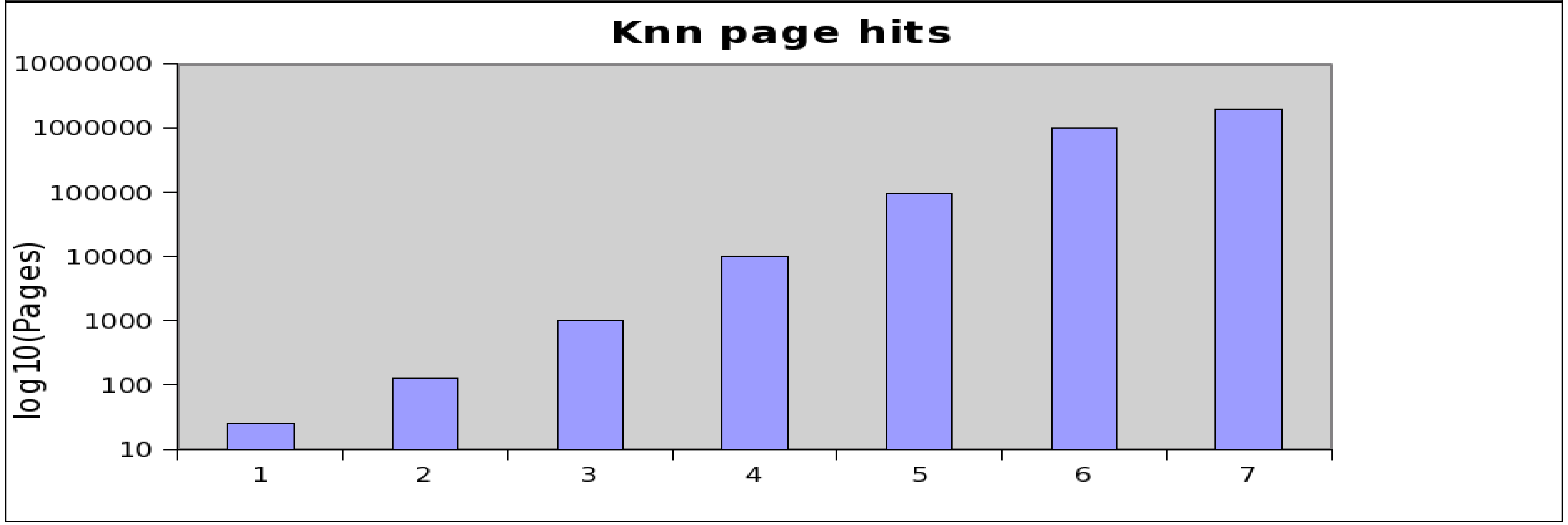
K	:hit	:knn	: seq
1	:3	:0.117	:6.072
10	:13	:0.247	:5.014
100	:103	:0.295	:6.381
1000	:996	:1.605	:8.670
10000	:9916	:16.487	:14.706

-> knn lose if k=n, n is small

Knn-search: Examples

- Real data
  2 mln points
  US, geonames





Knn-search: Examples

- Query: find 10 closest points in US with 'mars' in names to the point (5,5) - create composite index:

```
create index pt_fts_idx on geo using gist(point, to_tsvector('english',asciiname));

=# explain (analyze on, buffers on) select asciiname,point, (point <->
'5.0,5.0'::point) as dist from geo  where to_tsvector('english', asciiname)
@@ to_tsquery('english','mars') order by dist asc limit 10;
                                                           QUERY PLAN
-----------------------------------------------------------------------------------------
 Limit  (cost=0.00..33.55 rows=10 width=35) (actual time=0.452..0.597 rows=10 loops=1)
   Buffers: shared hit=56
   ->  Index Scan using pt_fts_idx on geo  (cost=0.00..34313.91 rows=10227 width=35)
(actual time=0.452..0.592 rows=10 loops=1)
         Index Cond: (to_tsvector('english'::regconfig, (asciiname)::text) @@
'''mar'''::tsquery)
         Sort Cond: (point <-> '(5,5)'::point)
         Buffers: shared hit=56
 Total runtime: 0.629 ms
(7 rows)
```

Knn-search: Existing solutions


```
knn=# select id, date, event from events order by date <-> '1957-10-04'::date asc limit 10;
   id   |    date    |                              event
--------+------------+-----------------------------------------------------------------
  58137 | 1957-10-04 | U.S.S.R. launches Sputnik I, 1st artificial Earth satellite
  58136 | 1957-10-04 | "Leave It to Beaver," debuts on CBS
 117062 | 1957-10-04 | Gregory T Linteris, Demarest, New Jersey, astronaut, sk: STS 83
 117061 | 1957-10-04 | Christina Smith, born in Miami, Florida, playmate, Mar, 1978
 102670 | 1957-10-05 | Larry Saumell, jockey
  31456 | 1957-10-03 | Willy Brandt elected mayor of West Berlin
  58291 | 1957-10-05 | 12th Ryder Cup: Britain-Ireland, 7 -4 at Lindrick GC, England
  58290 | 1957-10-05 | 11th NHL All-Star Game: All-Stars beat Montreal 5-3 at Montreal
  58292 | 1957-10-05 | Yugoslav dissident Milovan Djilos sentenced to 7 years
 102669 | 1957-10-05 | Jeanne Evert, tennis player, Chris' sister
(10 rows)

Time: 115.548 ms
```

- Very inefficient:
  - Full table scan, btree index on date won't help.
  - Sort full table

Knn-search: Existing solutions

contrib/btree_gist

```
knn=# select id, date, event from events order by date <-> '1957-10-04'::date asc limit 10;
   id   |    date    |                                 event
--------+------------+-----------------------------------------------------------------
  58137 | 1957-10-04 | U.S.S.R. launches Sputnik I, 1st artificial Earth satellite
  58136 | 1957-10-04 | "Leave It to Beaver," debuts on CBS
 117062 | 1957-10-04 | Gregory T Linteris, Demarest, New Jersey, astronaut, sk: STS 83
 117061 | 1957-10-04 | Christina Smith, born in Miami, Florida, playmate, Mar, 1978
 102670 | 1957-10-05 | Larry Saumell, jockey
  31456 | 1957-10-03 | Willy Brandt elected mayor of West Berlin
  58291 | 1957-10-05 | 12th Ryder Cup: Britain-Ireland, 7 -4 at Lindrick GC, England
  58290 | 1957-10-05 | 11th NHL All-Star Game: All-Stars beat Montreal 5-3 at Montreal
  58292 | 1957-10-05 | Yugoslav dissident Milovan Djilos sentenced to 7 years
 102669 | 1957-10-05 | Jeanne Evert, tennis player, Chris' sister
(10 rows)

Time: 0.590 ms
```

- Very inefficient:
  - 8 index pages read + 10 tuples read
  - NO sorting
  - About 200 times faster !

Knn-search: Examples

- pg_trgm support – distance = 1 – Similarity

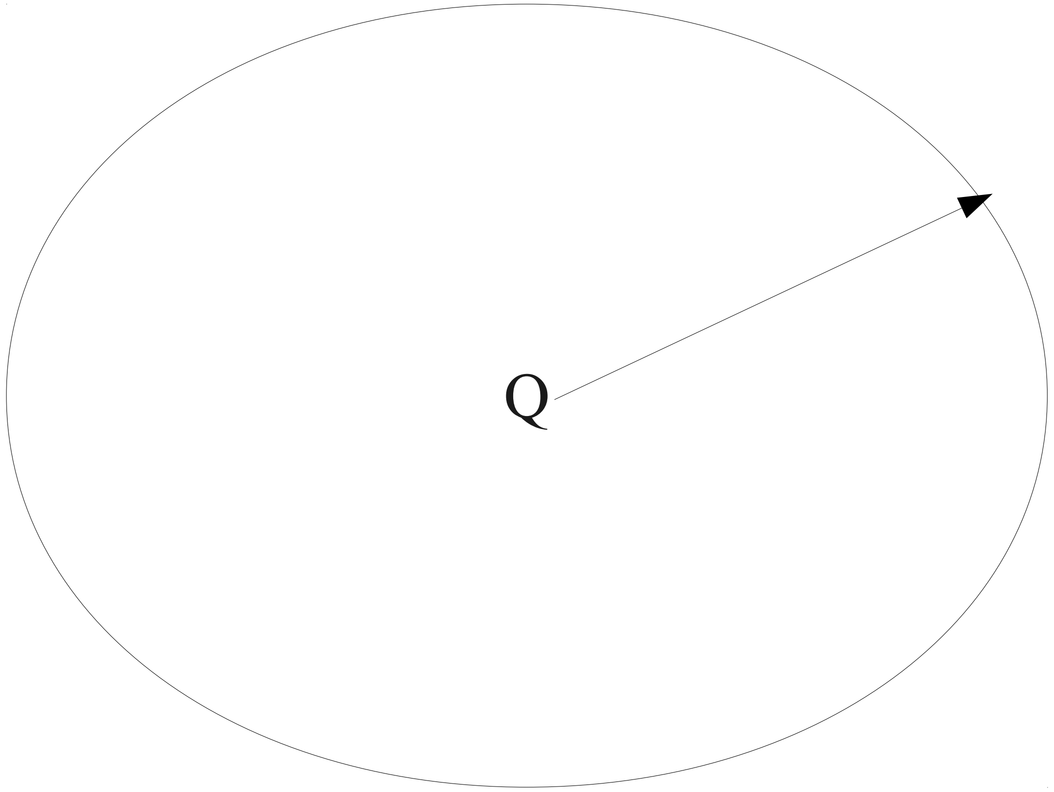
```
knn=# select date, event, ('jeorge ewashington'  <-> event ) as dist
from events order by dist asc limit 10;

    date    |                      event                        |   dist
------------+---------------------------------------------------+----------
 1732-02-11 | George Washington                                 | 0.458333
 1792-12-05 | George Washington re-elected U.S. pres            | 0.674419
 1811-02-23 | George Washington Hewitt, composer                |    0.675
 1753-08-04 | George Washington becomes a master mason          | 0.697674
 1941-07-19 | Jennifer Dunn, Rep-R-Washington                   | 0.710526
 1945-05-12 | Jayotis Washington, rocker                        | 0.714286
 1817-05-05 | George Washington Julian, MC, Union, died in 1899 |  0.72549
 1789-08-25 | Mary Ball Washington, mother of George, dies      | 0.729167
 1844-01-12 | George Washington Cable, American Novelist        | 0.729167
 1925-01-31 | George Washington Cable, American Novelist        | 0.729167
(10 rows)

Time: 187.604 ms
```

Knn-search: Examples

- Corner case for knn-search - all data are on the same distance from point Q !



Knn-search: Examples

- Corner case for Best First Strategy - all data are on the same distance from point Q !

```
create table circle (id serial, p point, s int4);
insert into circle (p,s)
    select point( p.x, p.y), (random()*1000)::int
    from ( select  t.x, sqrt(1- t.x*t.x)  as y
           from ( select random() as x, generate_series(1,1000000) ) as t
) as p;
create index circle_p_idx on circle using gist(p);
analyze circle;

Number of levels:          3
Number of pages:           8266
Number of leaf pages:      8201
```

Knn-search: Examples

- Corner case for knn-search - all data are on the same distance from point Q !

```
=# explain (analyze on, buffers on) select * from circle
                              order by (p <-> '(0,0)') asc limit 10;


 Limit  (cost=0.00..0.80 rows=10 width=24) (actual time=226.907..226.924
rows=10 loops=1)
   Buffers: shared hit=8276
   ->  Index Scan using circle_p_idx on circle  (cost=0.00..79976.58
rows=1000000 width=24) (actual time=226.905..226.921 rows=10 loops=1)
         Sort Cond: (p <-> '(0,0)'::point)
         Buffers: shared hit=8276 - read all index
 Total runtime: 230.885 ms
```

- Still 2 times faster than SEQ (454.331 ms) because of sorting

Knn-search: Status

Committed to 9.1

Knn: History of development

- Начало проекта - **Sep 8, 2007 at 7:54 PM**

```
date  Sat, Sep 8, 2007 at 7:54 PM
subject  Chat with Sergey V. Karpov

7:36 PM me: я тут knn-search занимаюсь, масса интересного. Все
думаю, как в постгресе это поиметь
 Sergey: а что это такое?
7:37 PM me: k-nearest соседей - супер важная задача
  найти 5 ближайших точек
7:38 PM Sergey: ближайших к чему?
 me: к заданной точке
7:39 PM Sergey: в какой системе координат?
 me: в любой, в n-мерном пространстве. В простом варианте - хотя
бы на земле/небе
7:40 PM это нужно для поиска похожих картинок, например.
  навиный вариант повторять запросы - не катит
```

Knn: History of development

- TODO (http://www.sai.msu.su/~megera/wiki/TODO)
  начало 2008 года, уже есть понимание что делать
- 25 июля 2009 года – письмо Paul Ramsey (POSTGIS)
- 10 июля 2009 года – контракт Open Planning Project Inc.
- 20 ноября 2009 года – патч KNNGiST v.0.1 (модуль расш)
- Commitfest nightmare
  - 22 июля 2010 – KNNGiST (v.0.8), commitfest
  - 13 сентября 2010 – KNNGiST (v.0.9)
  - 03 декабря 2010 – Tom Lane committed for 9.1 !
  - 21 января 2011 – contrib/btree_gist committed !

Knn: History of development

- Итого: На проект ушло больше 3 лет !
- Реальное программирование заняло несколько месяцев
- Основные причины:
  - Отсутствие поддержки
    - Слабая информационная связь разработчиков и заказчиков
  - Занятость разработчиков
  - Усложнение процедуры рассмотрения проектов в сообществе