M-TREES

Prof. Chris Jermaine
cmj4@cs.rice.edu

A Limitation of B-Trees

- Is that they only handle one-dimensional keys
  - — Or keys that can be ordered from first to last
- What if have other types of keys where range finds make sense?
  - — For example, text strings---find all strings that have a small ED to a query string
  - — Or vectors---find all vectors that have a small Euclidean distance to a query pt
  - — The latter ability is going to be important in our doc indexing system

M-Trees

- There are many, many generalizations to B-Trees
  - — But one that can handle both edit and Euclidean distance is an “M-Tree”
- In fact, can be used to handle any key...
  - — As long as you have a “distance metric” over the keys
- That is, you need a distance function $d(k_1, k_2)$ where:
  - — $d()$ is never negative
  - — $d()$ is zero if and only if the keys are equal
  - — $d()$ is reflexive
  - — and $d()$ obeys the triangle inequality
- Both edit and Euclidean distance are metric distances
  - — poof?

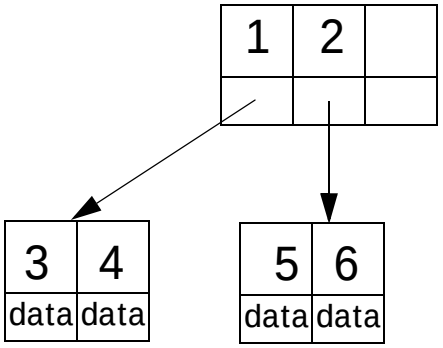
Proof ED Obeys Triangle Inequality

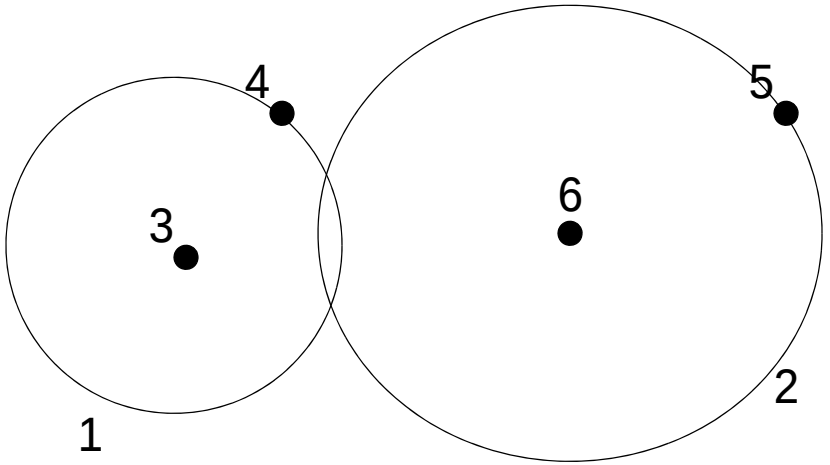
- Need to show that if
  - — $d(k_1, k_2)$ = c1
  - — $d(k_2, k_3)$ = c2
  - — then $d(k_1, k_3)$ <= c1 + c2
- How?
  - — Need to argue that we can get from k_1 to k_3 in at most c1 + c2 edt ops
  - — But this is easy!
  - — xform k_1 to k_2 in c1 ops
  - — xform k_2 to k_3 in c2 ops
  - — we have a c1 + c2 op xformation of k_1 to k_3

How Does an M-Tree Work?

- Rather than using (key, ptr) pairs in internal nodes...
  - — You use ((key, distance), ptr) pairs
  - — (key, distance) can be thought of as sort of a sphere
- The B-Tree “ordered” invariant is replaced in the M-Tree
  - — Rather than “Consider the (key_i, ptr_i) pair at position *i* in an internal node... For every key in the tree referred to by ptr_j (for $j <= i$), key <= key_i”
  - — We have: “Consider a ((key_i, $distance_i$), ptr_i) pair at position *i* in an internal node. For every key in the tree referred to by ptr_i, d(key, key_i) <= $distance_i$”
  - — That is, every data item in the subtree must be “close to” key_i

So Our New Picture Is This





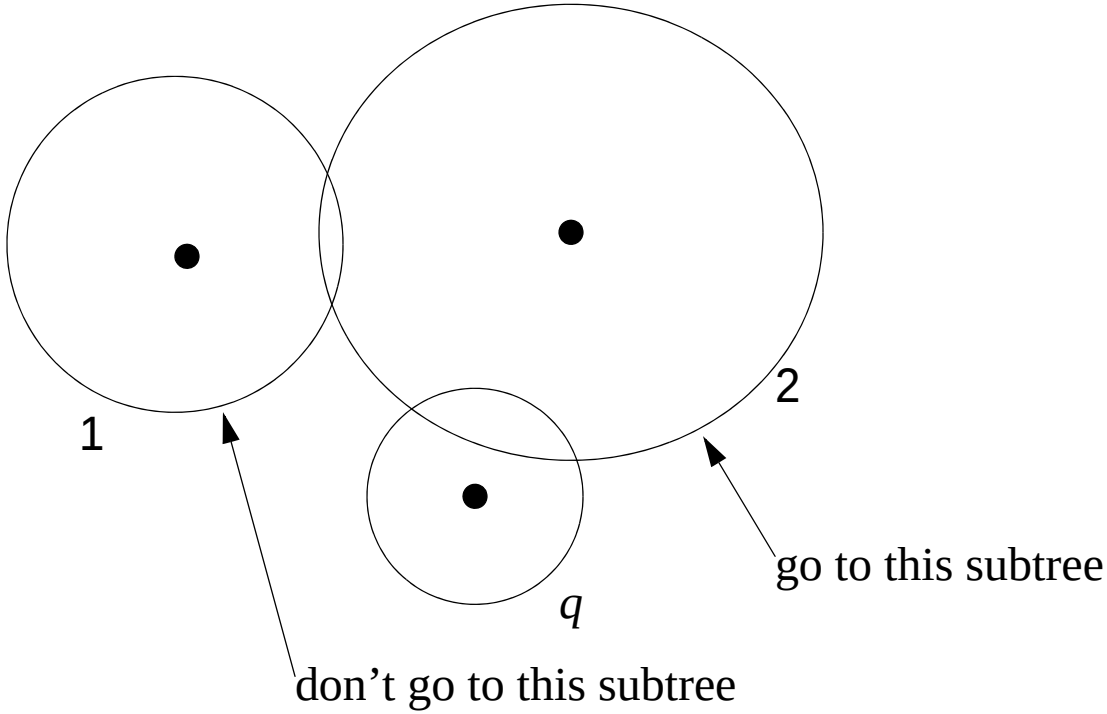
Key values 1 and 2 in internal nodes are now “spheres” that contain all data in subtree!

How Do You Query an M-Tree?

- Say you have a “range” find
  - That is, you have a (key_q, $distance_q$) pair
  - Where you want all (key, data) pairs in any leaf node
  - Such that d(key, key_q) <= $distance_q$
- Easy
  - In an internal node...
  - Just go to tree ptr_i if d(key_i, key_q) <= $distance_i$ + $distance_q$

In Other Words

If an internal node has spheres 1 and 2...

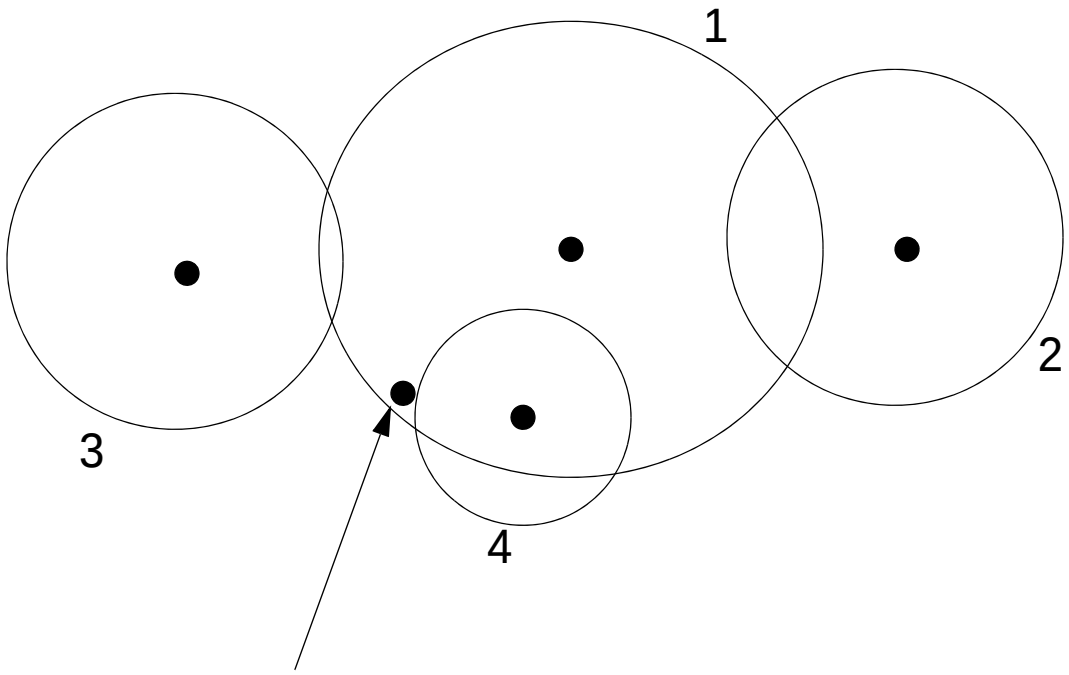


How Do You Insert Into an M-Tree?

- You want to insert a (key_{new}, $data_{new}$) pair into an internal node
  - — Just recursively add to subtree i if it minimizes $d(key_i, key_{new})$
  - — Note: you may have to increase $distance_i$ if $d(key_i, key_{new}) > distance_i$... why?

In Other Words

You have an internal node with spheres 1, 2, 3 and 4

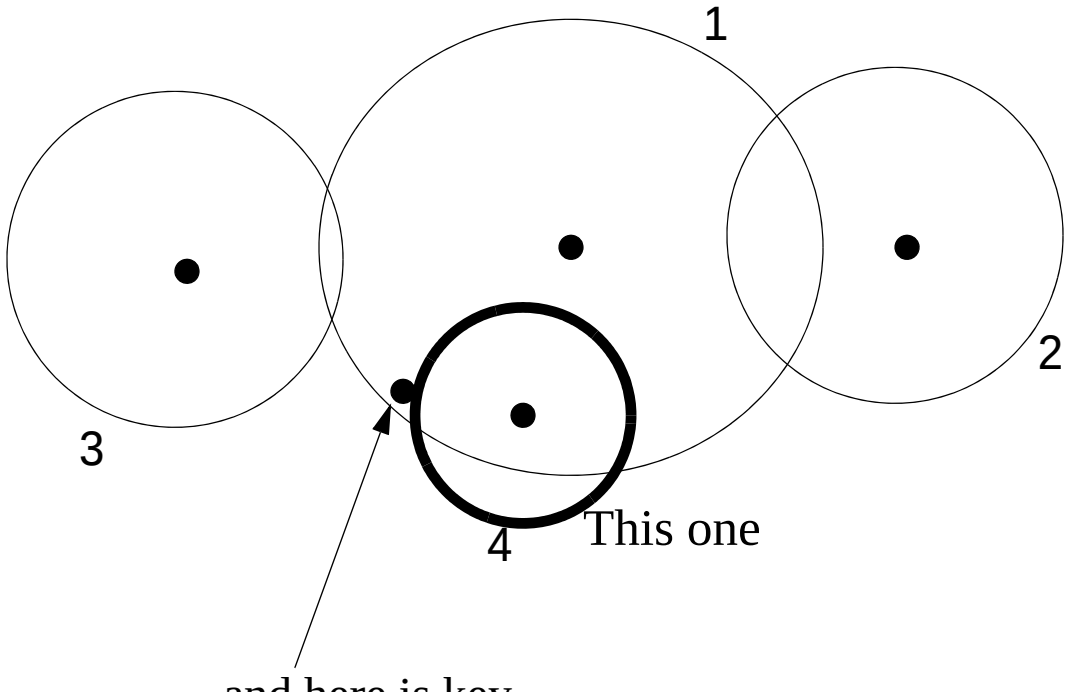


and here is key$_{new}$

Which subtree to insert into?

In Other Words

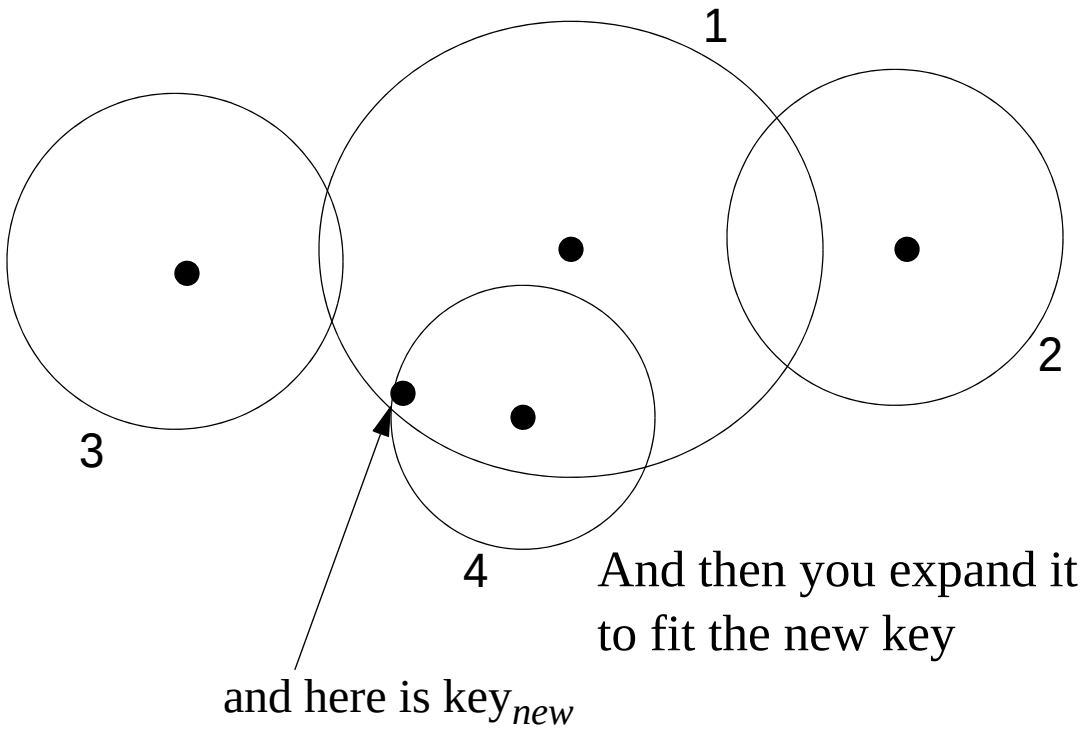
You have an internal node with spheres 1, 2, 3 and 4



Which subtree to insert into?

In Other Words

You have an internal node with spheres 1, 2, 3 and 4



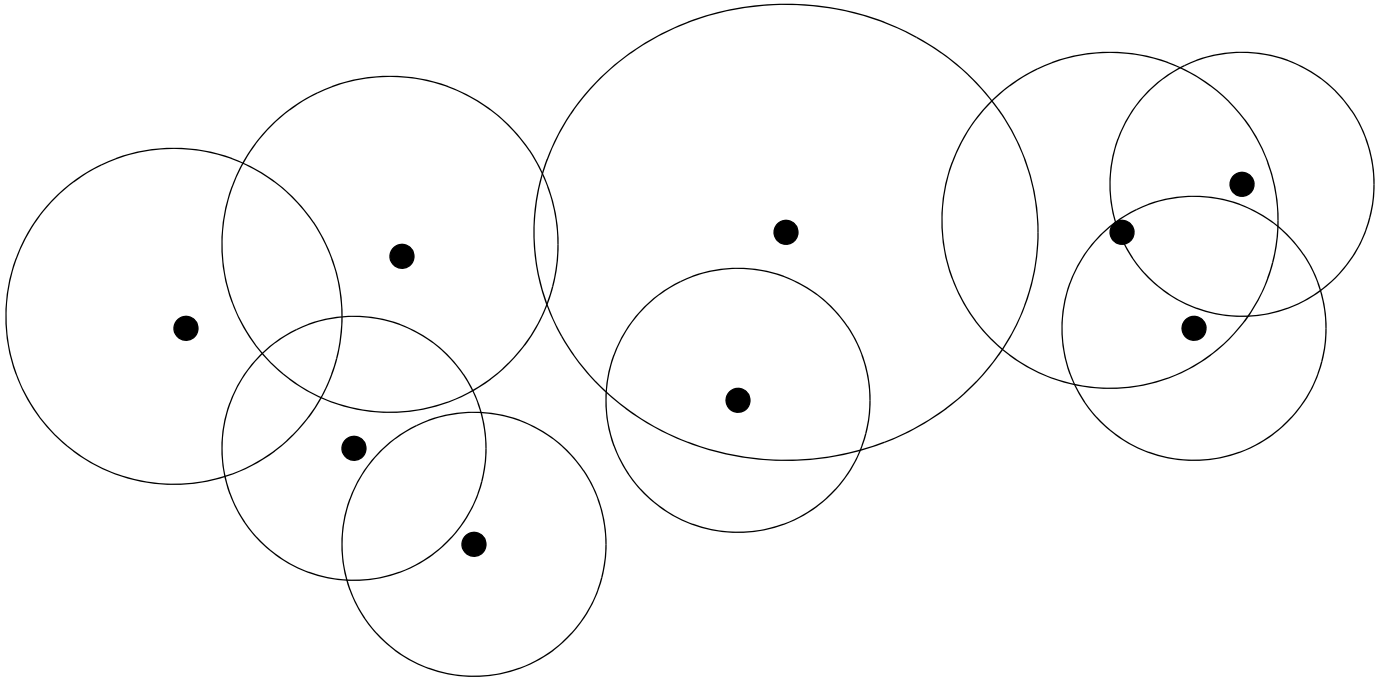
Which subtree to insert into?

The Only Other Diff ‘Tween B- and M-Trees

- Splitting!
- In B-Trees, it was “sort and kick up the median”
- In M-Trees, it is “cluster, then kick up the two resulting spheres”

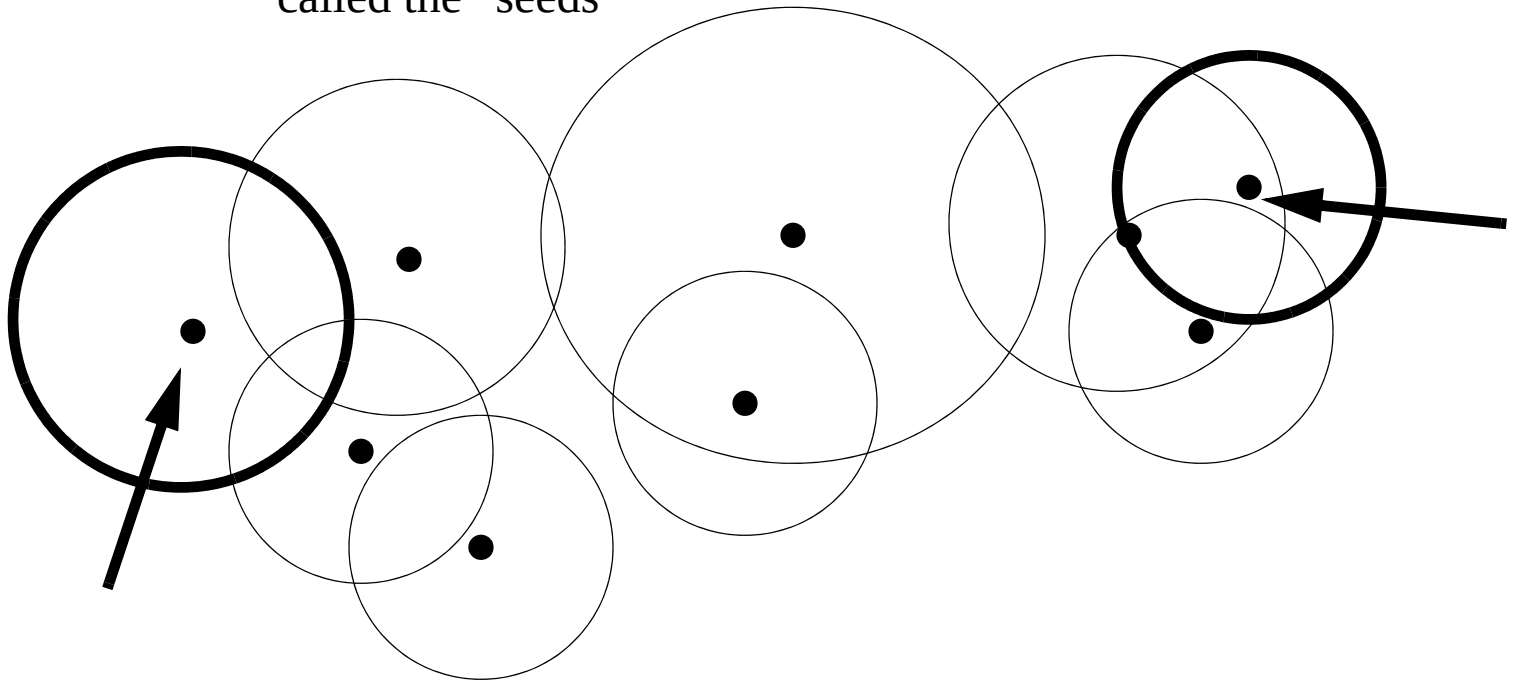
How To Cluster

Say you want to split an internal node with the following spheres



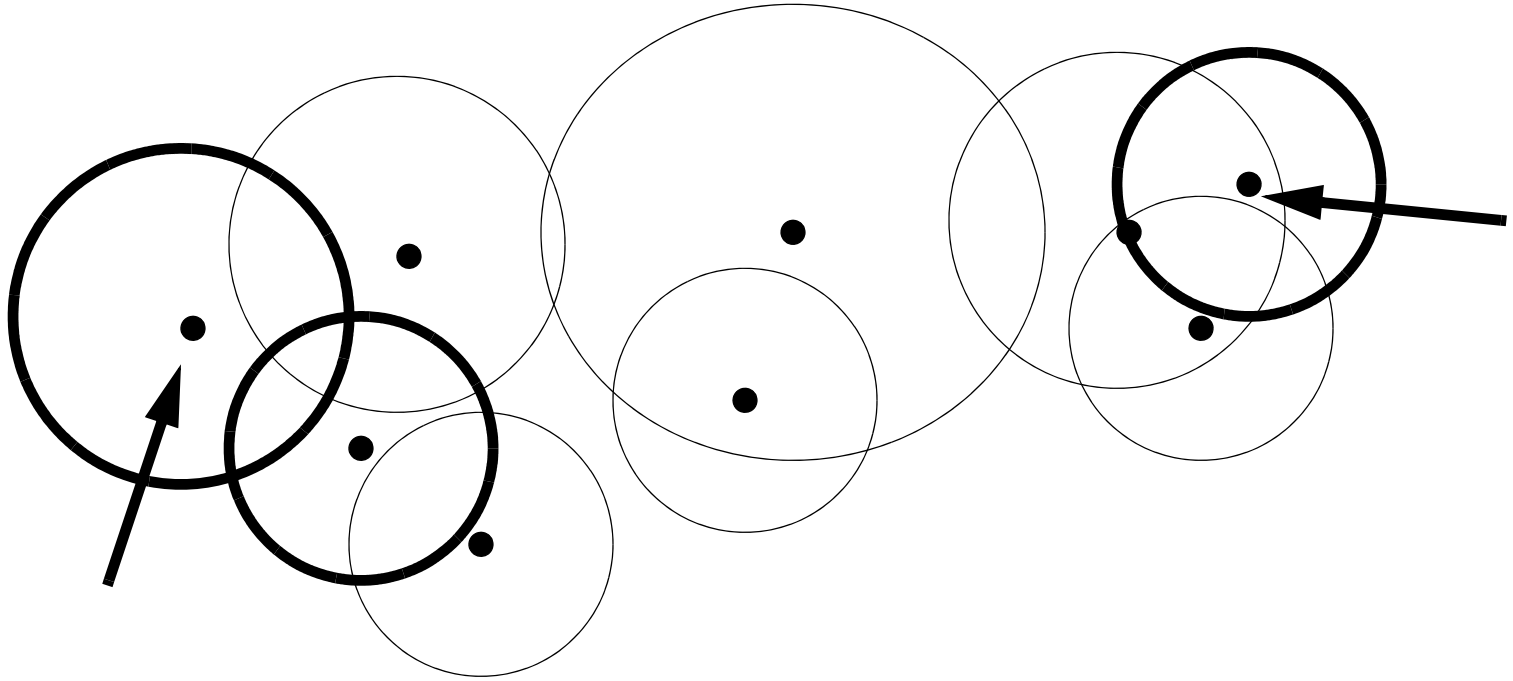
Step One

Find the most distant pair of keys (use a simple nested loop)... called the “seeds”



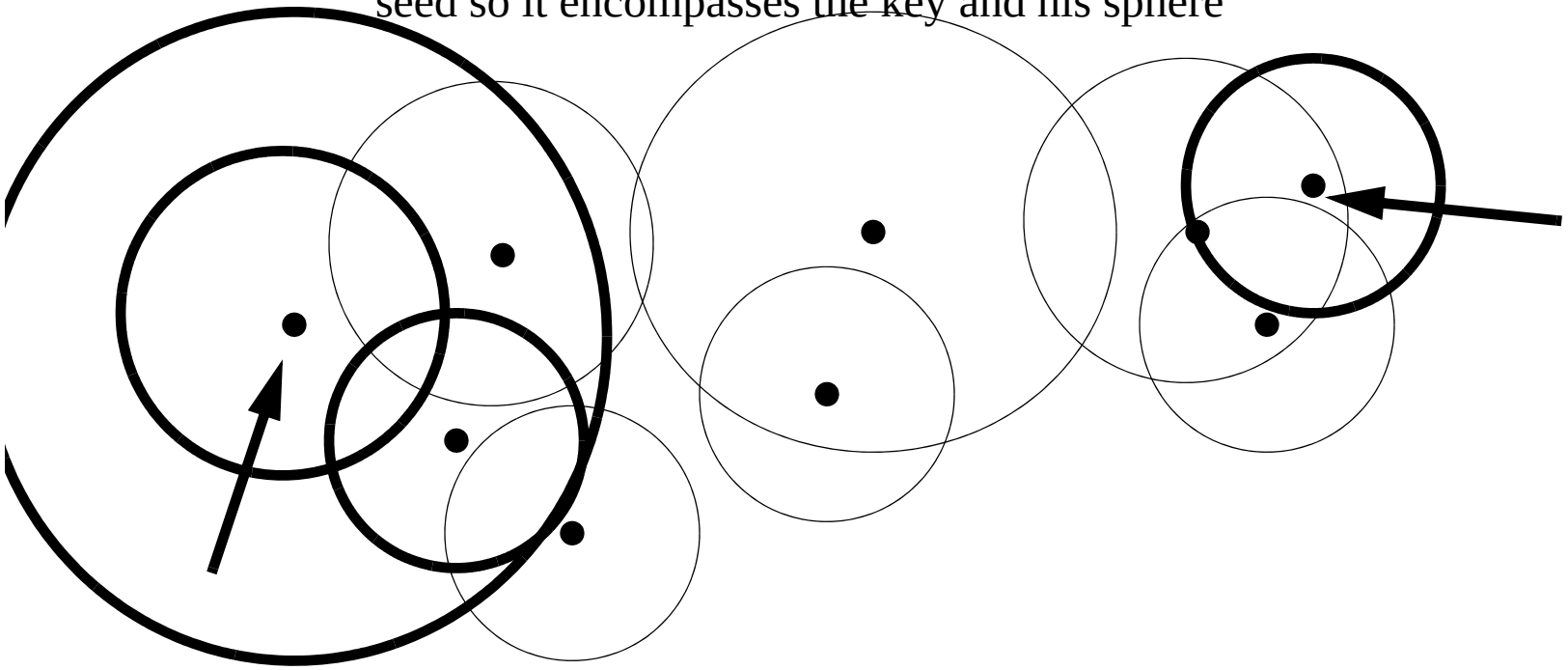
Step Two

Pick one seed and find the key closest to it



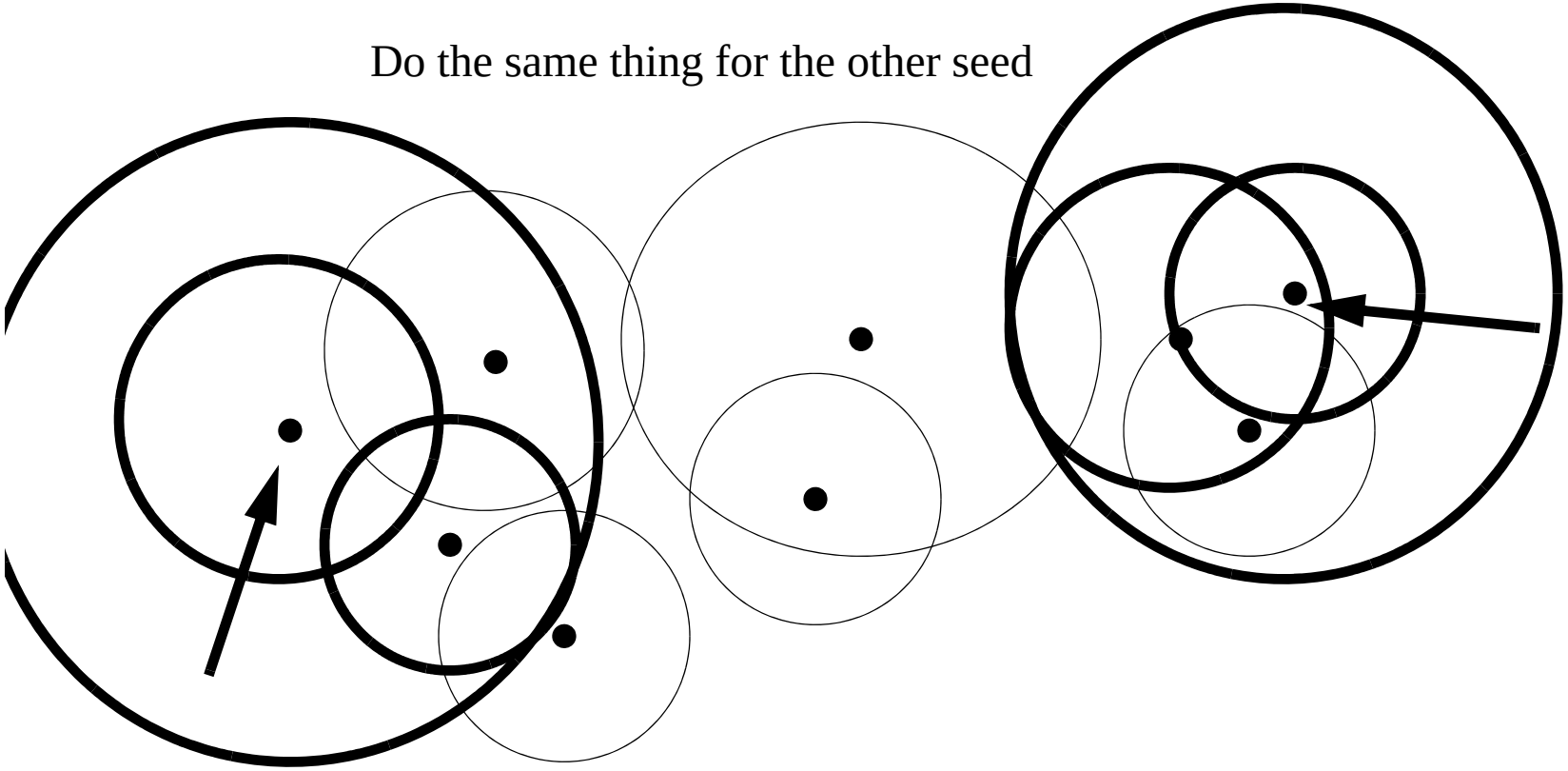
Step Three

“Attach” that key to the seed, and expand a sphere centered at the seed so it encompasses the key and his sphere

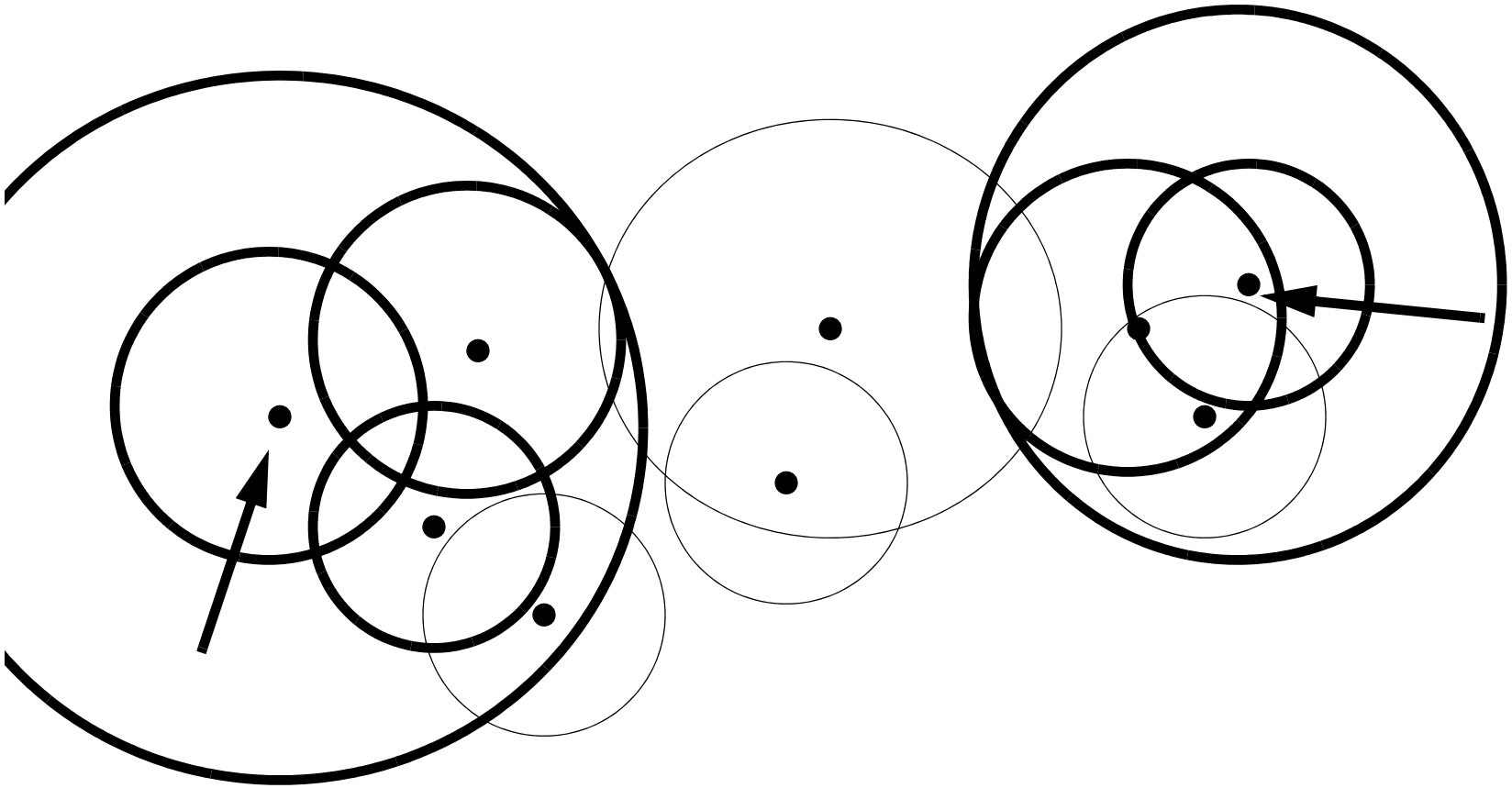


Step Four

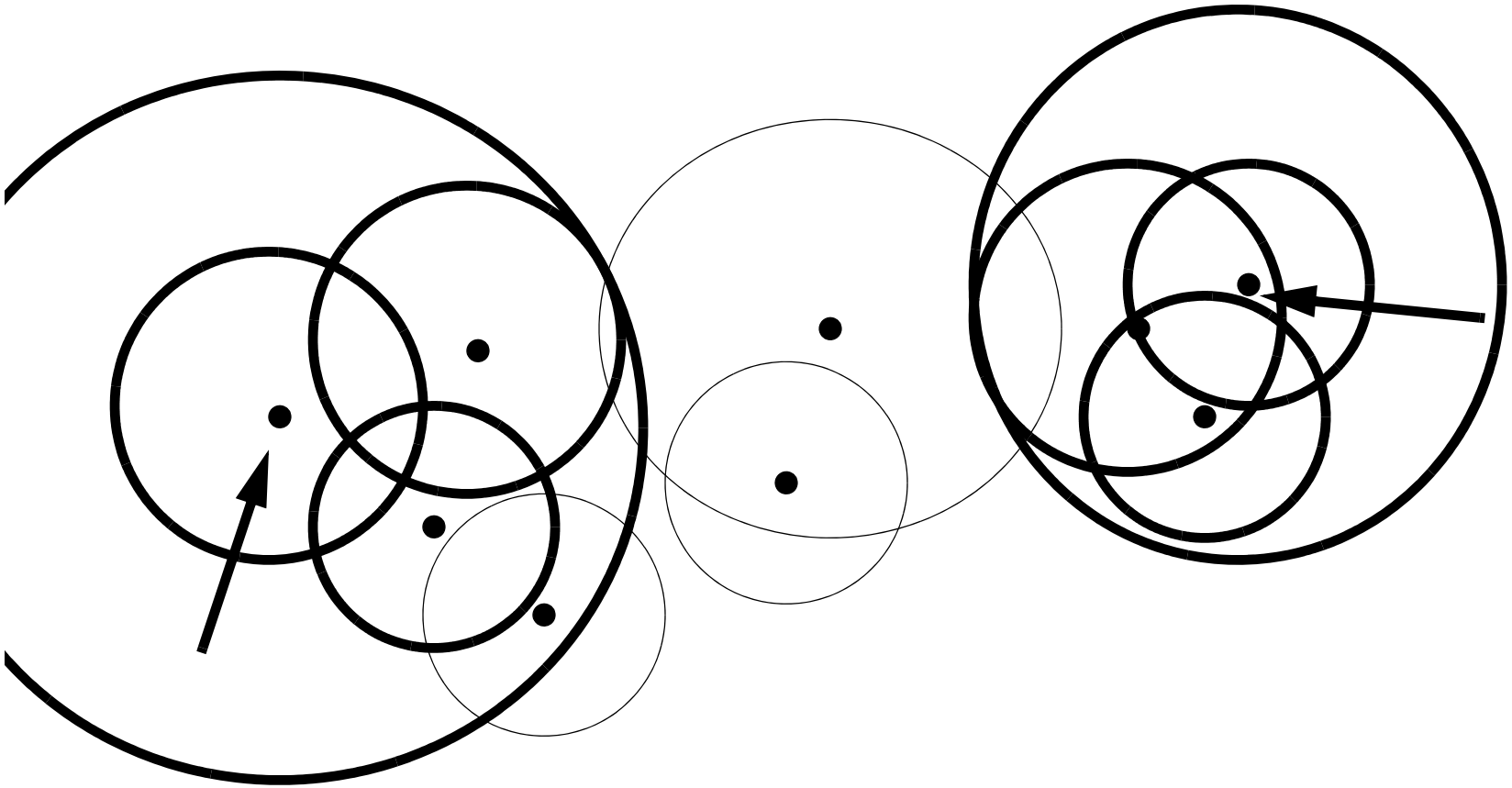
Do the same thing for the other seed



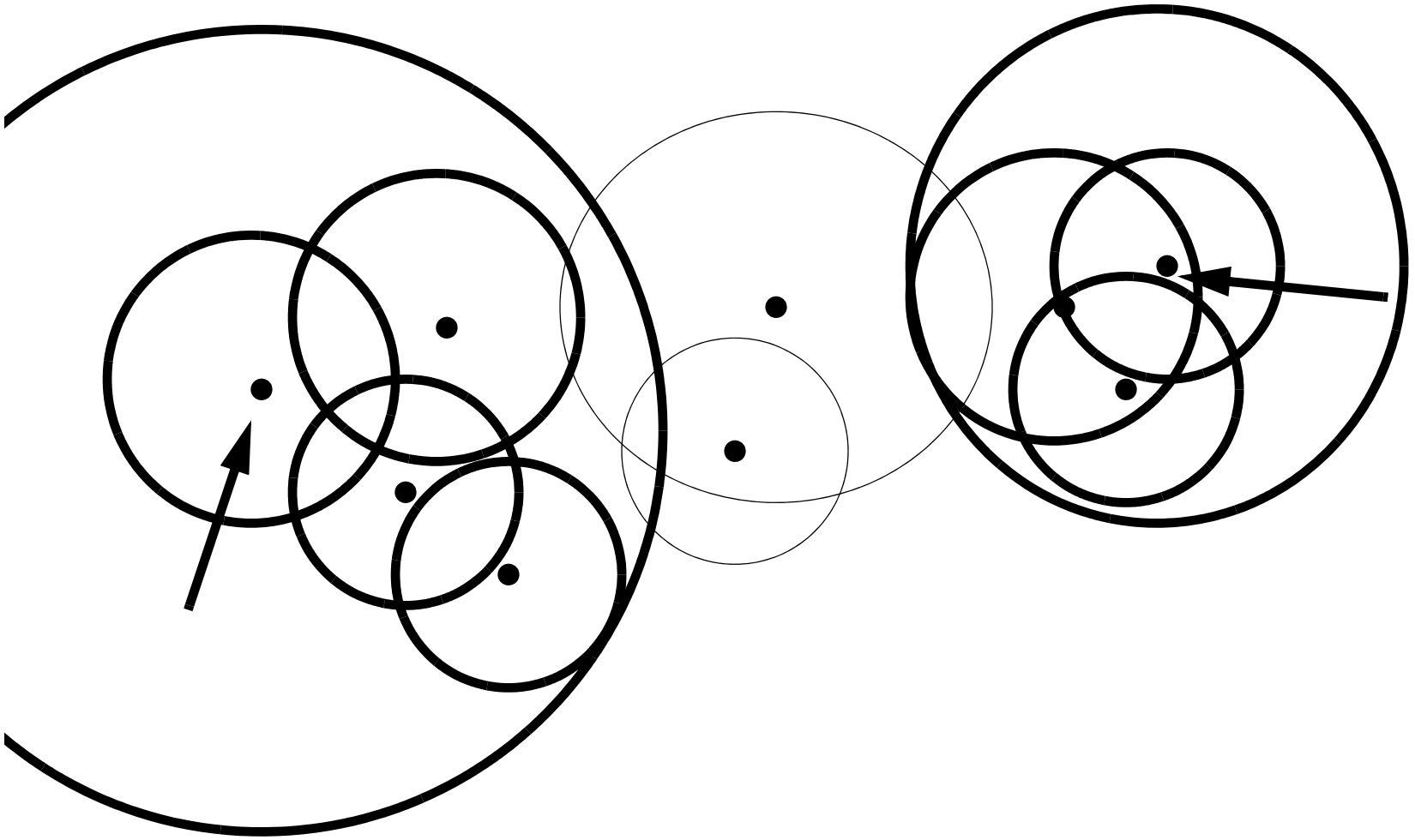
And Repeat!



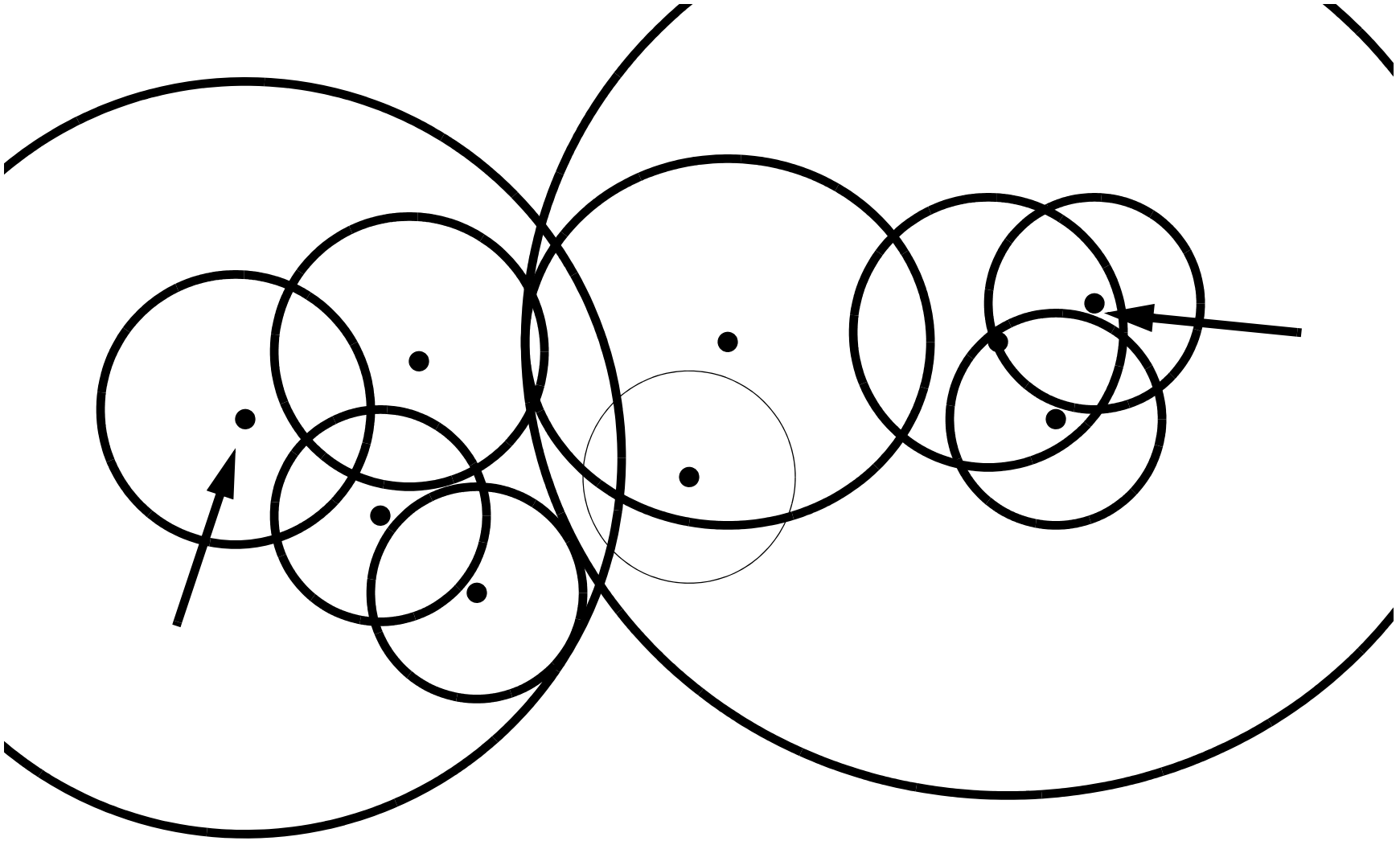
And Repeat!



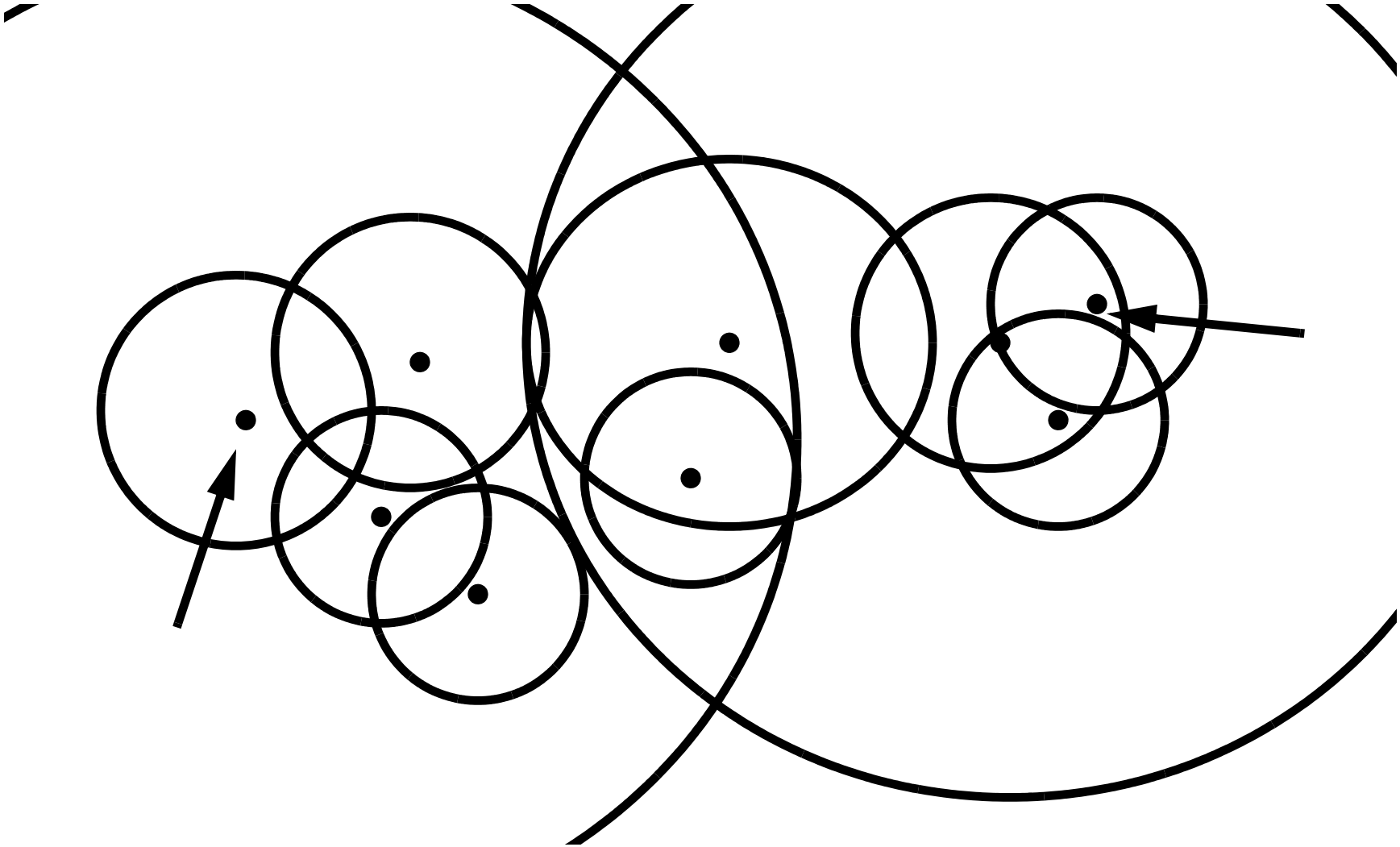
And Repeat!



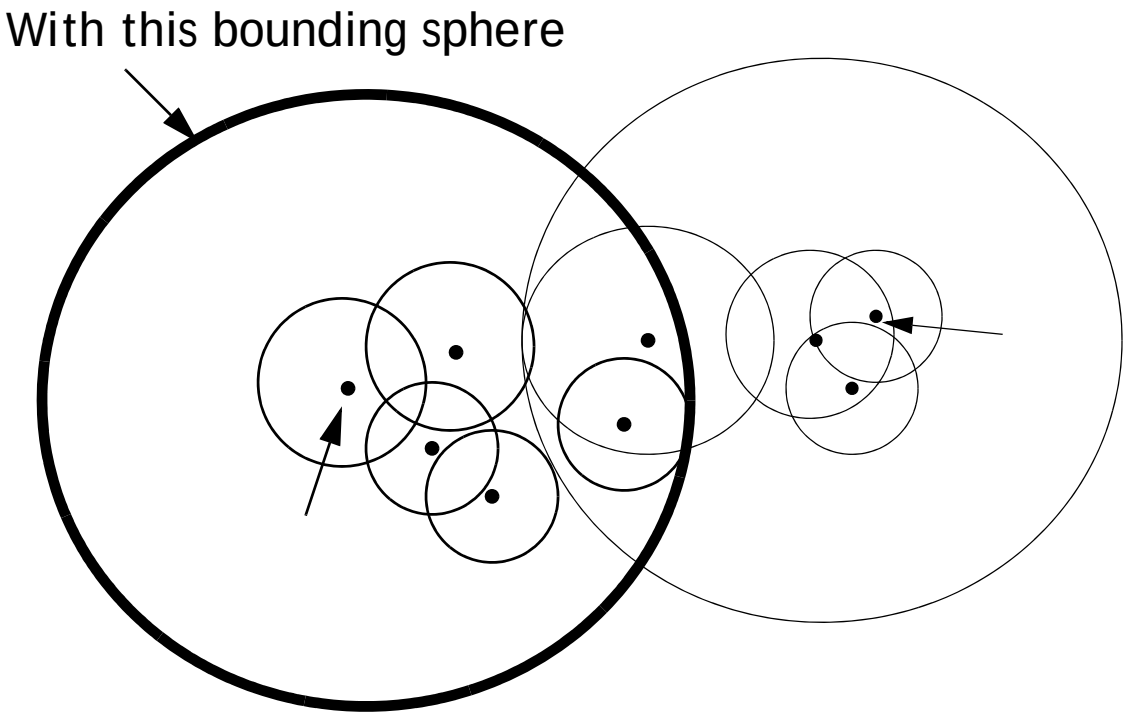
And Repeat!



And Repeat!

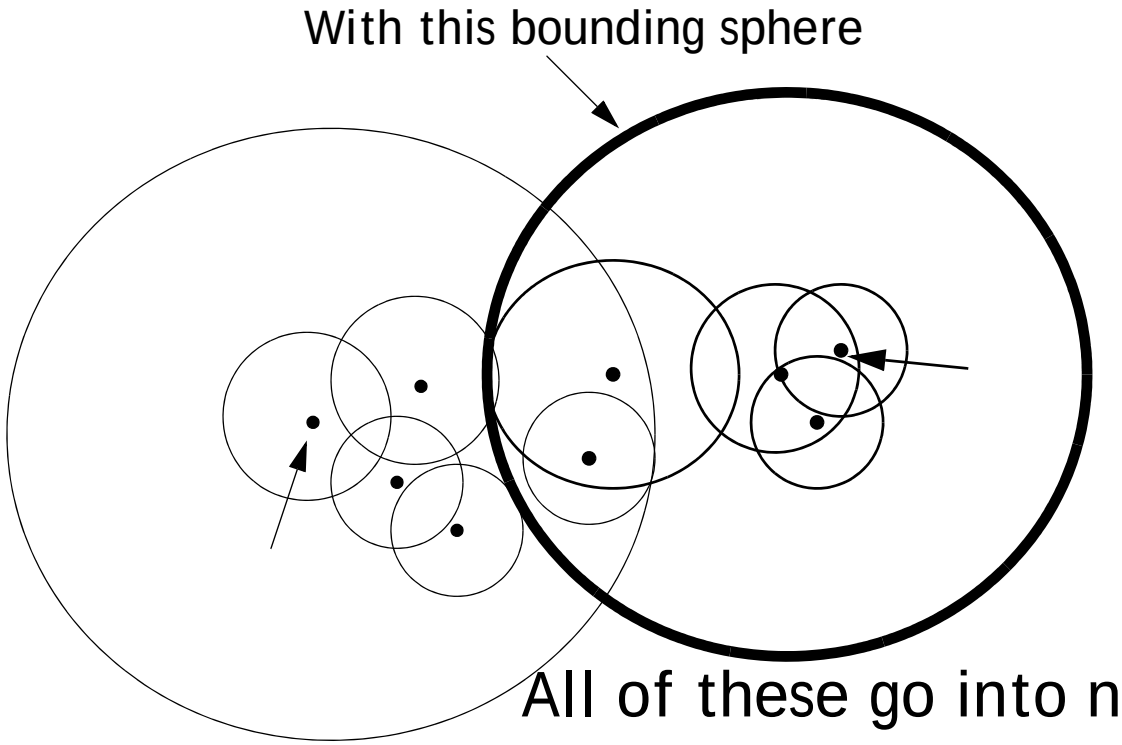


In The End…



All of these stay in the old node

In The End…



Questions?