Spatial Data Structures Class

Professor: Hanan Samet

Presentation by: Guilherme Fonseca

Paper: Deformable Spanners With Applications

Authors:

J. Gao

L. Guibas

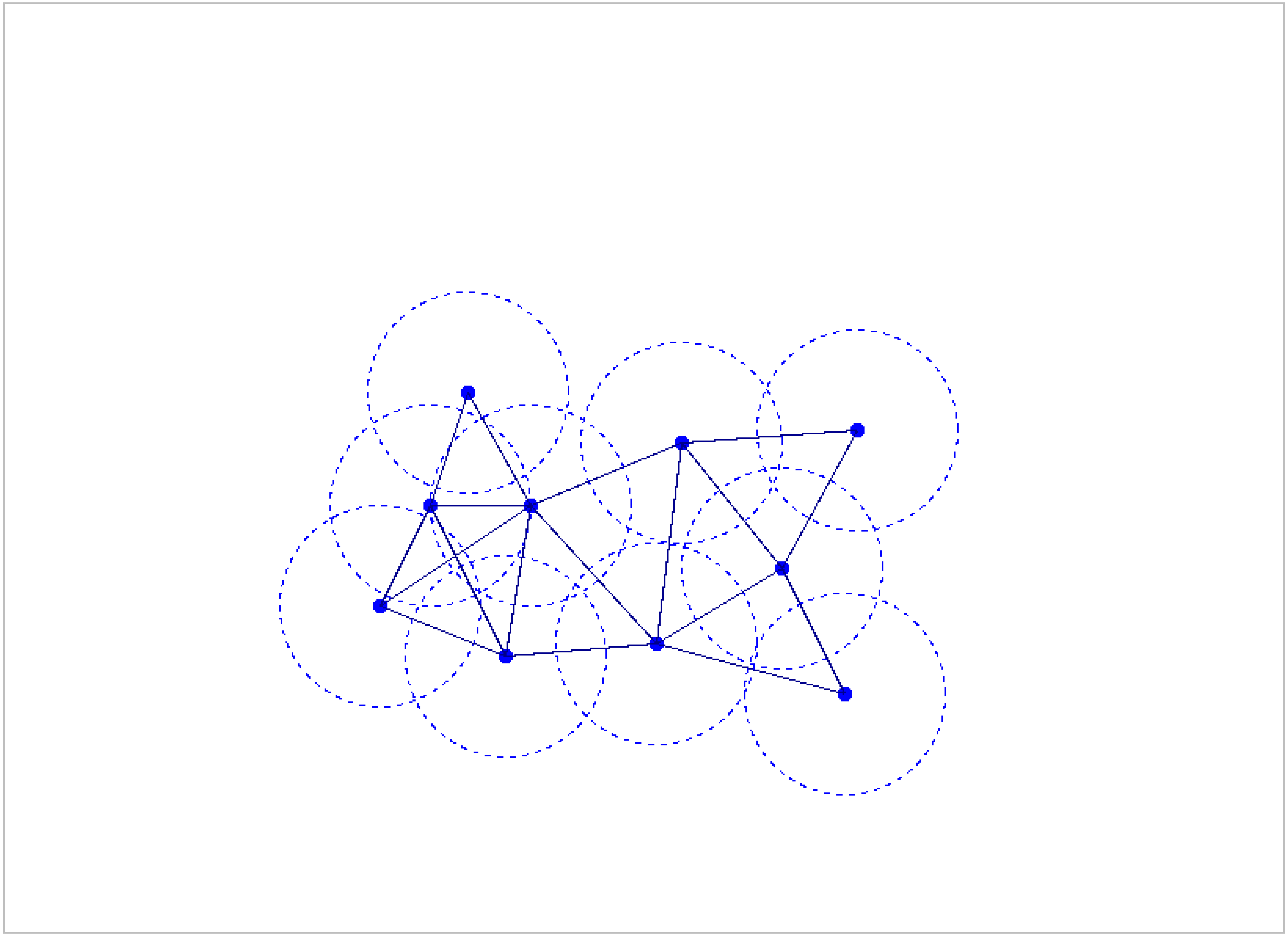
A. Nguyen

Properties of DefSpanner

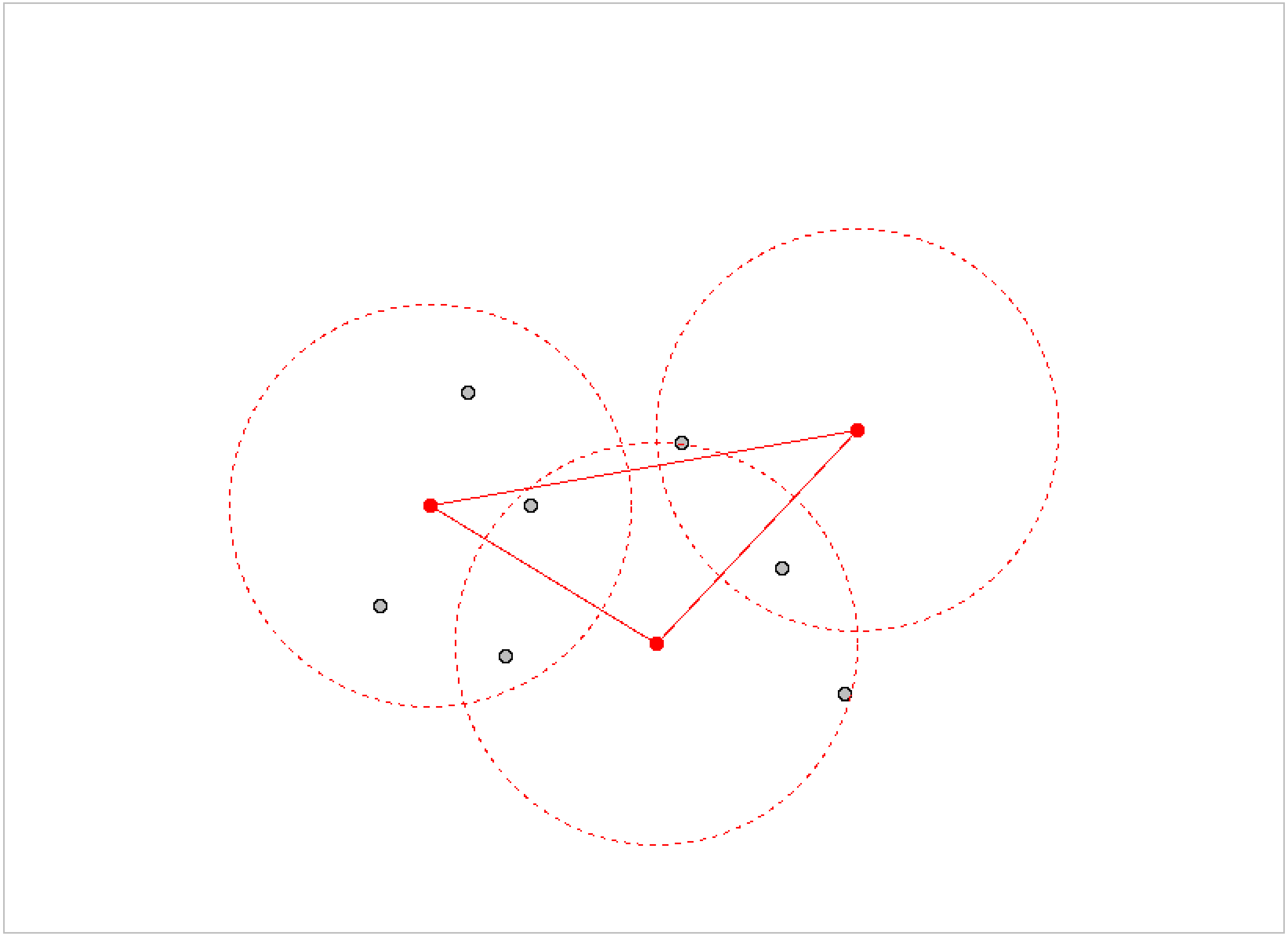
- ANN query time is O(log f / ε^d), where f is the ratio between the distance of the furthest points and the distance of the closest points.
- The parameter ε is part of the query, not of the structure!
- Structure size is O(n).
- It can be constructed in O(n log f) time.
- Supports insertions and deletions in O(log f) time, assuming f is O(poly(n)).
- Allows continuous movement of the points.

Building a DefSpanner

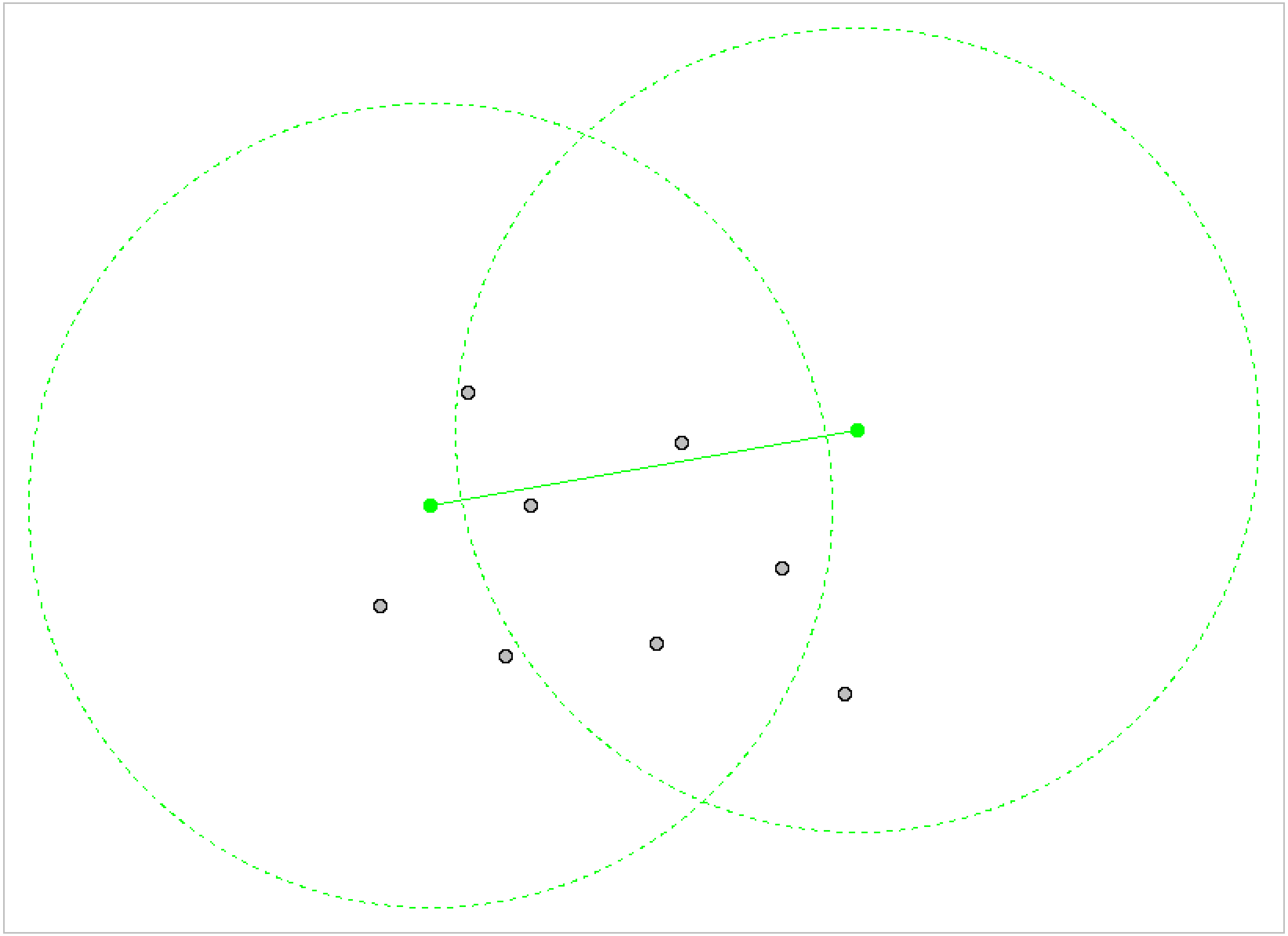
- Say the closest pair of points has distance 1 and the furthest has distance *f*.
- For *i* from 0 to lg(*f*)
  - Find a maximal set of circles of radius 2^i centered at the input points such that the center of a circle is not contained in another circle
  - Connect 2 circles if they intersect



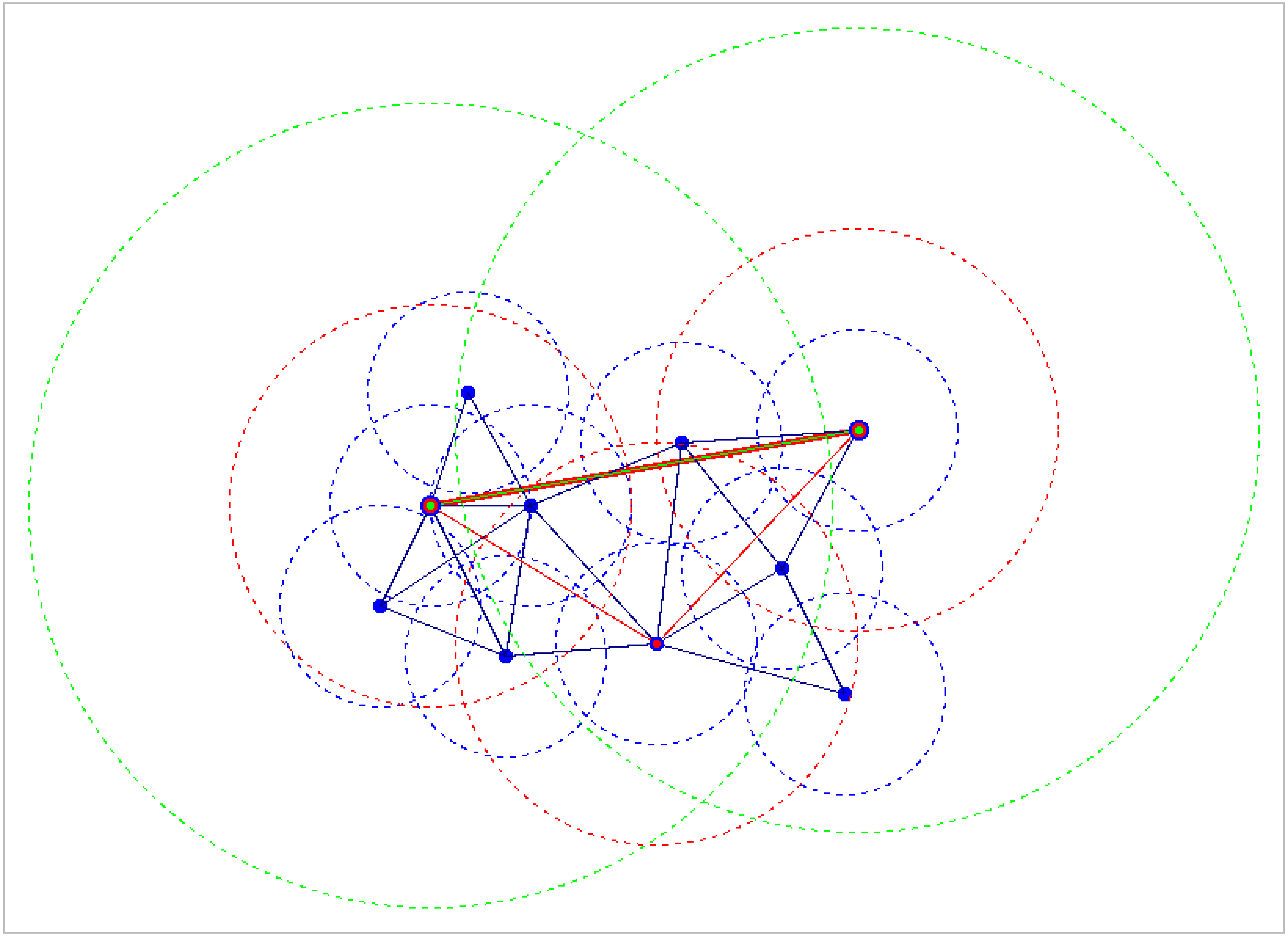
$i = 0, \qquad 2^i = 1$



$i = 1, \qquad 2^i = 2$



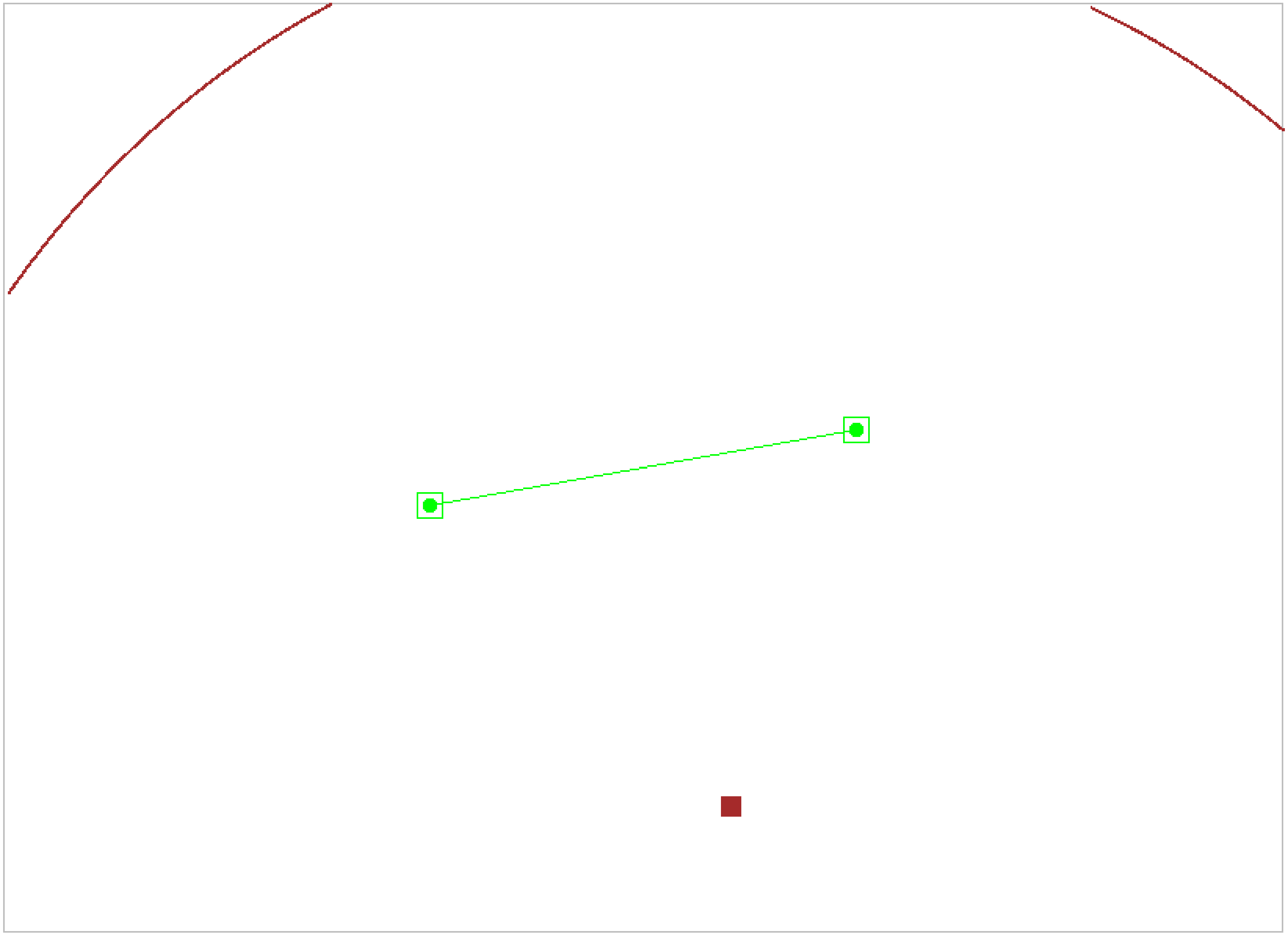
$i = 2, \qquad 2^i = 4$



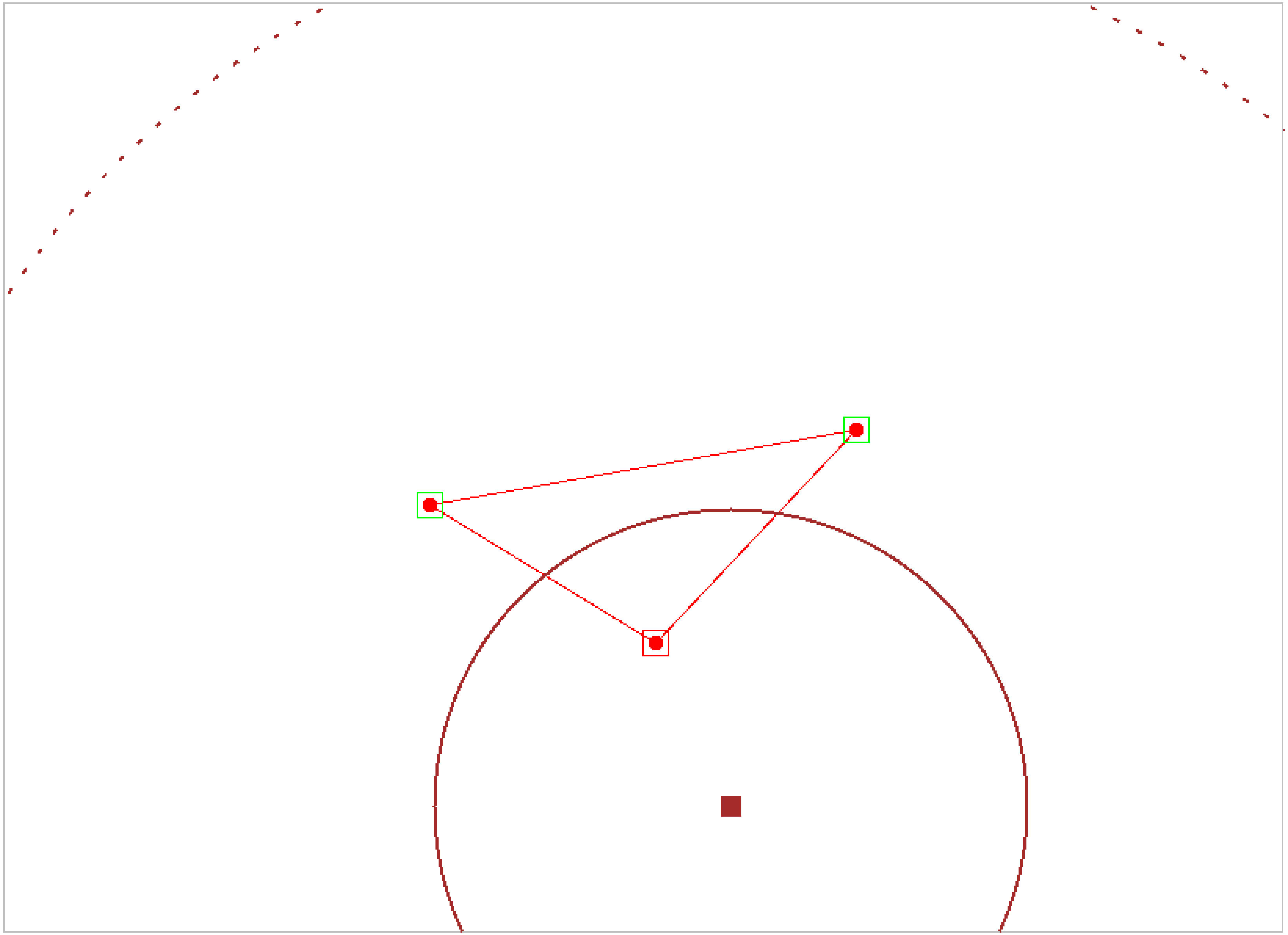
ANN Queries

To find a point within distance $(1+\varepsilon)$ of the NN:

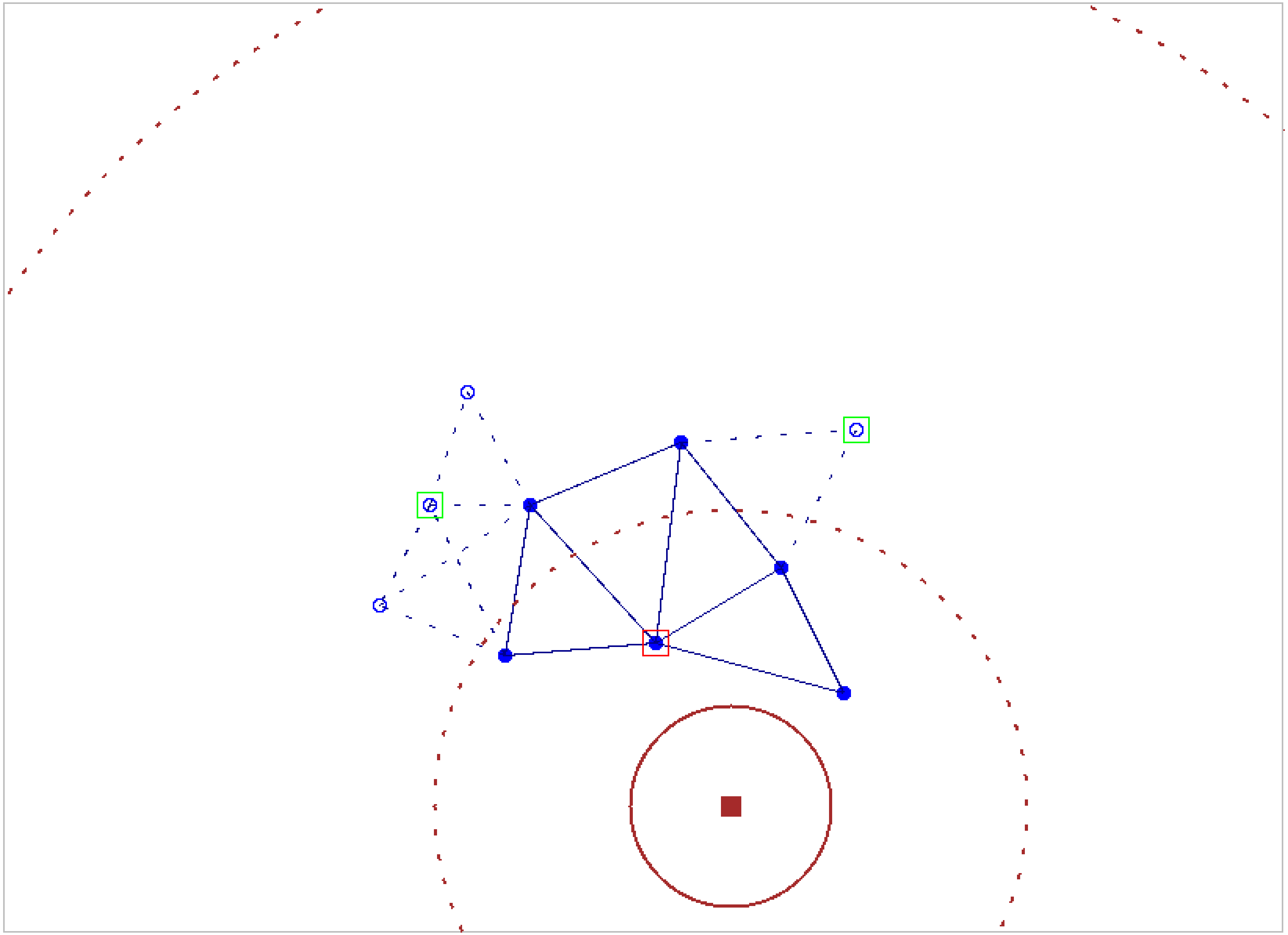
- Starting top level (maximum *i*), we keep track of all points within distance at most $(2+4/\varepsilon)^i$.
- Fortunately, we only need to check the points adjacent to the points chosen on the previous level.
- We return the closest point among these (compare all of them).
- (In the example, $\varepsilon=4$, so the radius is not too big.)



$\varepsilon = 4, \qquad i = 2, \qquad (2+4/\varepsilon)^i = 9$



$\varepsilon = 4,\qquad i = 1,\qquad (2+4/\varepsilon)^i = 3$



$\varepsilon = 4, \qquad i = 0, \qquad (2+4/\varepsilon)^i = 1$

Why is it efficient?

- Query time: O(log f / ε^d).
- Structure size: O(n).
- The number o balls of radius ε that can be contained in a ball of radius 1 is O(1/ε^d).
- The same result holds if you don't require the balls to be disjoint, but require that the center of a ball is not contained in another ball.
- Scaling this result, we can prove that at most O(1/ε^d) points will be searched for each level in a query.
- A similar argument can be used to bound the size.