Fixed-Radius Near Neighbors Problem

Definition of Fixed-Radius Near Neighbors Problem: Given a set $P$ and a distance $r > 0$, report all pairs of distinct points $p, q \in P$ such that $|pq| < r$.

Application: air traffic control, molecular dynamics

Start from the simplest case: assume $P$ in 1-dimension

Naïve solution: Enumerate all pairs of distinct points and compute the distance between each pair. The number of distinct pairs of $n$ points is:

$$\binom{n}{2} = \frac{n(n - 1)}{2}$$

So the computational complexity is $O(n^2)$.

Solution 2: Sorting based algorithm

Assume we have a set of sorted points $P$: $x_1 < x_2 < \cdots < x_n$, for $i$ running from 1 to $n$, we consider and report the successive points of $x_i$ with distance less than $r$: $x_{i+1}, x_{i+2}$ and so on, until we first find a point whose distance exceeds $r$.

The computational complexity of the algorithm is dominated by sorting, which is $O(n \log n)$.

Solution 3: Bucketing based algorithm

Rather than sorting the points, we can subdivide the 1-dimensional line into intervals of length $r$: $\ldots, [-2r, -r), [-r, 0), [0, r), [r, 2r), \ldots$

We call each interval a bucket. A bucket with indices $b$ is $[br, (b + 1)r)$.

In $O(n)$ time, we can associate the $n$ points of $P$ with a set of buckets. The size of the occupied buckets is at most $n$. We then store the indices of the occupied buckets in a hash table. Thus in $O(1)$ expected time, we can determine which bucket contains a given point and look this bucket up in the hash table. Considering that a point $x$ lies in bucket $b$, then any successors that lie within
distance $r$ must lie either in bucket $b$ or in $b + 1$. This suggests the following algorithm shown below.

<table>
<thead>
<tr>
<th>Fixed-Radius Near Neighbor on the Line by Bucketing</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1) Store the points of $P$ into buckets of size $r$, stored in a hash table.</td>
</tr>
<tr>
<td>(2) For each $x \in P$ do the following:</td>
</tr>
<tr>
<td>(a) Let $b$ be the bucket containing $x$.</td>
</tr>
<tr>
<td>(b) Search all the points of buckets $b$ and $b + 1$, report $x$ along with all those points $x'$ that lie within distance $r$ of $x$.</td>
</tr>
</tbody>
</table>

The time complexity of the algorithm is $O(n + k)$ where $k$ is the number of the reported neighbors. Note that the total number of the distance computation in step (2)(b) is no more than $2k$.

Generalization to n-dimension of the bucketing based algorithm: This bucketing based algorithm is easy to extend to higher dimensions. For example, in 2-dimension, we bucket points into a square grid in which each grid square is of side length $r$. 