

6.046 Problem 1-2Collaborators: *none*

(a) The farthest distance between two points within a $\frac{1}{2} \times \frac{1}{2}$ square is $\frac{\sqrt{2}}{2} < 1$. Then FCC must reject the set of requests if two requests are in or on the boundary of the same square.

(b) The problem is that given two lists of points (x, y) : L_x and L_y , sorted by x coordinate and y coordinate respectively, and contain the same set of points, we wish to check whether there are two points within Euclidean distance 1.

To solve this problem, we first find $L_x[\lceil \frac{n}{2} \rceil]$, and divide L_x into L_{xl} and L_{xr} , based on whether a point is before or after $L_x[\lceil \frac{n}{2} \rceil]$ in L_x . Then we get the corresponding L_{yl} and L_{yr} , by comparing every point in L_y with $L_x[\lceil \frac{n}{2} \rceil]$. Then check the left half and the right half separately. If both sides return null, we need to check whether there is a point from the left and a point from the right that the Euclidean distance between them is less than 1. We can do this by finding all points whose x coordinate is within 1 of $L_x[\lceil \frac{n}{2} \rceil]$'s x coordinate and putting them in a list L' with increasing y coordinate (we can do so by going through L_y). For any point $L'[k]$ in L' , let's check the distance between $L'[k]$ and $L'[k+1]$, $L'[k+2]$, ..., $L'[k+8]$ (if there exists). If all these 8 distances for all k are greater than 1, we can return null. Otherwise, return the first pair we find whose distance is less than 1.

Let's prove that only checking these pairs of points works. Otherwise null is returned, but there is a $L'[k]$ and $L'[k+d]$, where $d > 8$, and the distance between $L'[k]$ and $L'[k+d]$ is less than 1. Then $L'[k+d]_y - L'[k]_y < 1$, and $L'[k+8]_y - L'[k]_y < 1$. Then we know that at least one of $L'[k+8]_y - L'[k+4]_y < \frac{1}{2}$ and $L'[k+4]_y - L'[k]_y < \frac{1}{2}$ holds. We can assume that $L'[k+4]_y - L'[k]_y < \frac{1}{2}$, (as the other condition is basically the same) then all five points in $L'[k]$, $L'[k+1]$, $L'[k+2]$, $L'[k+3]$, $L'[k+4]$ are within lines $x = L_x[\lceil \frac{n}{2} \rceil]_x - 1$, $x = L_x[\lceil \frac{n}{2} \rceil]_x + 1$, $y = L'[k]_y$ and $y = L'[k+3]_y$. Draw lines $x = L_x[\lceil \frac{n}{2} \rceil]_x$, $x = L_x[\lceil \frac{n}{2} \rceil]_x + \frac{1}{2}$, $x = L_x[\lceil \frac{n}{2} \rceil]_x - \frac{1}{2}$, we divide that area into 4 identical subareas, each of which can be contained in a $\frac{1}{2} \times \frac{1}{2}$ square. Then there must be two points in the same area. According to (a), the distance between them is less than 1. This contradicts with our hypothesis.

As all we do every time is going through L_y twice, and checking at most $8n$ pairs of points, we have

$$T(n) = 2T(\frac{n}{2}) + O(n)$$

Then the overall running time is $O(n \lg n)$.

(c) This problem is that given the same lists in (b), we wish to check whether there are three points within Euclidean distance 1 of each other.

The divide part is the same as (b). If both sides return true, we still need to get L' . Then let's do the same checking: for $L'[k]$, check the distance between $L'[k]$ and $L'[k+1], \dots, L'[k+16]$. If two of them are less than 1, say, $L'[k]L'[k+i] < 1$ and $L'[k]L'[k+j] < 1$, check the distance between $L'[k+i]$ and $L'[k+j]$. For any k there are at most $16 + C_{16}^2 = 126$ checks. If we cannot find three points within distance 1 of each other, return null.

The proof is very similiar. Assume that the points passed all the checks, but there are $L'[k]$, $L'[k+d]$ and $L'[k+d+t]$ where $d+t > 16$, and the distance between $L'[k]$ and $L'[k+d+t]$ is less than 1. Then $L'[k+d+t]_y - L'[k]_y < 1$, and $L'[k+16]_y - L'[k]_y < 1$. Then we know that at least one of $L'[k+16]_y - L'[k+8]_y < \frac{1}{2}$ and $L'[k+8]_y - L'[k]_y < \frac{1}{2}$ holds. We can assume that $L'[k+8]_y - L'[k]_y < \frac{1}{2}$, (as the other condition is basicly the same) then all nine points in $L'[k], \dots, L'[k+8]$ are within lines $x = L_x[\lfloor \frac{n}{2} \rfloor]_x - 1$, $x = L_x[\lfloor \frac{n}{2} \rfloor]_x + 1$, $y = L'[k]_y$ and $y = L'[k+8]_y$. Draw lines $x = L_x[\lfloor \frac{n}{2} \rfloor]_x$, $x = L_x[\lfloor \frac{n}{2} \rfloor]_x + \frac{1}{2}$, $x = L_x[\lfloor \frac{n}{2} \rfloor]_x - \frac{1}{2}$, we divide that area into 4 identical subareas, each of which can be contained in a $\frac{1}{2} \times \frac{1}{2}$ square. Then there must be three points in the same area. According to (a), the distances between them are all less than 1. This contradicts with our hypothesis, as we should have found these three points when checking on the lowest point of them.

Time used is basicly the same as (b), while this time we checked at most $126n$ pairs of points. Then we still have

$$T(n) = 2T\left(\frac{n}{2}\right) + O(n)$$

Then the overall running time is still $O(n \lg n)$.