CS 6363: Computer Algorithms – Fall 2019
Homework #5 Solutions

Problem #1.

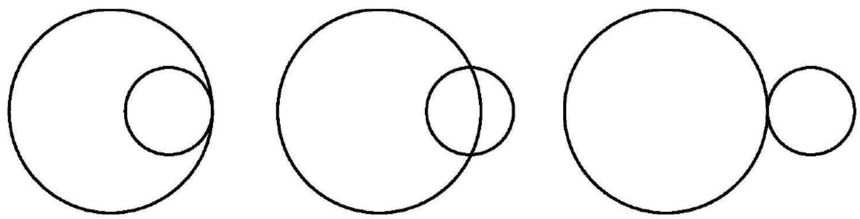
<u>Input</u>: A set $Q = \{p_1, p_2, ..., p_n\}$ of n points
<u>Output</u>: A pair $P(p_i, p_j)$ that have maximal slope

The slope of $p_k = (x_k, y_k)$ and $p_l = (x_l, y_l)$ is defined as $s_{kl} = \frac{|y_k - y_l|}{|x_k - x_l|}$.

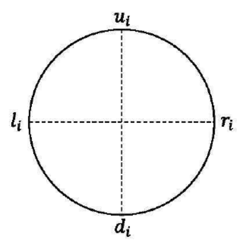
Assume that the input points are sorted according to x-coordinates. Then, the maximum slope s_{max} must be defined by a pair of adjacent points. Why? Suppose s_{max} is defined by p_i and p_j, and other point p_{i+1} has an x-coordinate somewhere between x_i and x_j. Now, suppose that the point p_{i+1} lies on the line $\overline{p_i p_j}$, then p_i and p_{i+1} define s_{max} as required. Conversely, suppose that the point p_{i+1} does not lie on the line $\overline{p_i p_j}$, then either $\overline{p_i p_{i+1}}$ or $\overline{p_{i+1} p_j}$ must have greater slope than $\overline{p_i p_j}$. Contradiction occurs to the assumption that p_i and p_j define s_{max}.
Based on this idea, given an input of points, we sort the points according to x-coordinates using $O(n \lg n)$ sorting algorithm. Let us say that the set of points sorted is $\{p_1, p_2, ..., p_n\}$. Then, run linear time search to find an adjacent pair of points, in other words, we exam each pair of points, $(p_1, p_2), (p_2, p_3), ..., (p_{n-1}, p_n)$, and return the pair that defines the maximum slope s_{max}. Total running time is $O(n \lg n) + O(n) = O(n \lg n)$.

Problem #2. (#33.2 – 6)

All possible cases that two circles intersect: (Assume that there are no ***identical*** circles.)



As shown above, if any two circles intersect, then the sum of two radii of the two circles must be larger than or equal to the distance between two center points of the two circles. Otherwise, they do not intersect.



For this problem, we first sort circles according to x-coordinates of the most left points (l_i) of the circles. Let us denote $C = \{c_1, c_2, ..., c_n\}$ to be the set of sorted circles. Each circle c_i has its horizontal segment $\overline{l_i r_i}$ and vertical segment $\overline{u_i d_i}$, and both of them pass the center point of the circle. Now, we move a vertical sweep line and check (1) while a circle c_i is open, whether other circle c_j is also open or not. (Here, we say that a circle c_i is "open" from the moment the sweep line starts passing the point l_i to the moment right after the sweep line passes the point r_i.) If so, (2) we also check whether the vertical segments of the two circles are overlapped or not. When both of those two conditions are satisfied, we can simply check (3) if those two circles ***indeed*** intersect by checking the distance property for circle intersection. If we find any pair of circles, then we return FALSE.

The algorithm takes $O(n \lg n)$ like the SEGMENT-INTERSECTION algorithm. (We also have $2n$ event points, and both of overlapping test of horizontal segements and circle intersection test take $O(1)$ time.)

Problem #3.

$(a + bi)(c + di) = (ac - bd) + (ad + bc)i$

<u>Idea</u>: $(a - b)(c + d) = ac + ad - bc - bd = A$, $(a + b)(c - d) = ac - ad + bc - bd = B$
$(A + B) = 2(ac - bd) \Longrightarrow ac - bd = (A + B)/2$
$(A - B) = 2(ad - bc) \Longrightarrow ad - bc = (A - B)/2$
$(A - B) + 4bc = 2(ad - bc) + 4bc = 2(ad + bc) \Longrightarrow ad + bc = \frac{(A-B)}{2} + 2bc$
From the above results, $(ac-bd)+(ad+bc)i$ can be computed by using only three multiplications, $(a-b)(c+d)$, $(a + d)(c - d)$, and bc.

Input: $(a + bi)(c + di)$
Output: $real + imaginary$ using three multiplications

1.	$\alpha \leftarrow a - b, \beta \leftarrow c + d, \gamma \leftarrow a + b, \delta \leftarrow c - d$
2.	$p_1 \leftarrow \alpha \cdot \beta$
3.	$p_2 \leftarrow \gamma \cdot \delta$
4.	$p_3 \leftarrow b \cdot c$
5.	$real \leftarrow \frac{p_1 + p_2}{2}$
6.	$imaginary \leftarrow \frac{p_1 - p_2}{2} + 2p_3$

Problem #4. (#30.1 – 7)

Represent A and B as polynomial of following forms:

$p(x) = \sum_{i=0}^{10n} a_i x^i$, where $a_i = \begin{cases} 1 & \text{if } i \in A \\ 0 & \text{otherwise} \end{cases}$

$q(x) = \sum_{i=0}^{10n} b_i x^i$, where $b_i = \begin{cases} 1 & \text{if } i \in B \\ 0 & \text{otherwise} \end{cases}$

Compute the product of $p(x) \cdot q(x)$ using FFT, we have $r(x) = p(x) \cdot q(x) = \sum_{i=0}^{10n} c_i x^i$.
From $r(x)$, $C = \{i | 0 \leq i \leq 20n \text{ and } c_i \neq 0\}$.
The conversion of set to polynomial or vice versa is bounded by time complexity of $O(n)$.
The time complexity for FFT is $O(n \log n)$.
Therefore, total time complexity is $O(n \log n)$.
The coefficient c_i is the number of times the number i in C is realized as a sum of elements in A and B. (i is realized as sum $i = j + k$ where $j \in A$ and $k \in B$.)

Example:
$n = 2$, range from 0 to 20
$A = \{1, 10\}$, $B = \{11, 20\}$, $C = \{12, 21, 30\}$
$p(x) = x^1 + x^{10}$, $q(x) = x^{11} + x^{20}$
$r(x) = x^1 x^{11} + x^1 x^{20} + x^{10} x^{11} + x^{10} x^{20} = x^{12} + 2x^{21} + x^{22}$
In the set C of Cartesian sum of the two sets, A and B, the number of times the number 12 was realized as a sum is 1, the number of times the number 21 was realized as a sum is 2, and the number of times the number 30 was realized as a sum is 1.

Problem #5

1. We have to find the DFT of the vectors a = (2,4,6,8)

By multiplying (2,4,6,8) by the principal nth root of unity , we have

p(1) = 2+4+6+8=20

P(w)= 2+4i-6-8i= -4-4i

$P(w^2)$ = 2-4+6-8= -4

$P(w^3)$ = 2-4i-6+8i= -4+4i

Therefore, the DFT of the vector is (20,-4-4i,-4,-4+4i)

2. (1,3,2,4,3,1,2,0)

To get the DFT, we also multiply by W(the principal nth root of unity)

Here, n=8. $Y_k = A(W_8^K)$ = (16, $-2+\sqrt{2}\ (3I-1), 0, -2+2\sqrt{2}\ (3I+1), 0, -2-\sqrt{2}\ (3I-1), 0, -2-2\sqrt{2}(3I+1)$

Problem #6

Suppose that p and q are the two furthest apart points. Also, to a contradiction, suppose, without loss of generality that p is on the interior of the convex hull. Then, construct the circle whose center is q and which has p on the circle. Then, if we have that there are any vertices of the convex hull that are outside this circle, we could pick that vertex and q, they would have a higher distance than between p and q. So, we know that all of the vertices of the convex hull lie inside the circle. This means that the sides of the convex hull consist of line segments that are contained within the circle. So, the only way that they could contain p, a point on the circle is if it was a vertex, but we supposed that p wasn't a vertex of the convex hull, giving us our contradiction.