CS-UY 2413: Design&Analysis of Algorithms—Spring 2017 Prof. Boris Aronov
Department of Computer Science and Engineering Prof. Lisa Hellerstein

Homework 5 Solutions

Due March 27 at the start of lecture (Aronov's sections).
Due March 28 at the start of lecture (Hellerstein's section).
This homework should be handed in on paper. No late homeworks accepted. Contact your professor for special circumstances.

Policy on collaboration on this homework: The policy for collaboration on this homework is the same as in previous homeworks. By handing in this homework, you accept that policy. Remember: A maximum of 3 people per group.

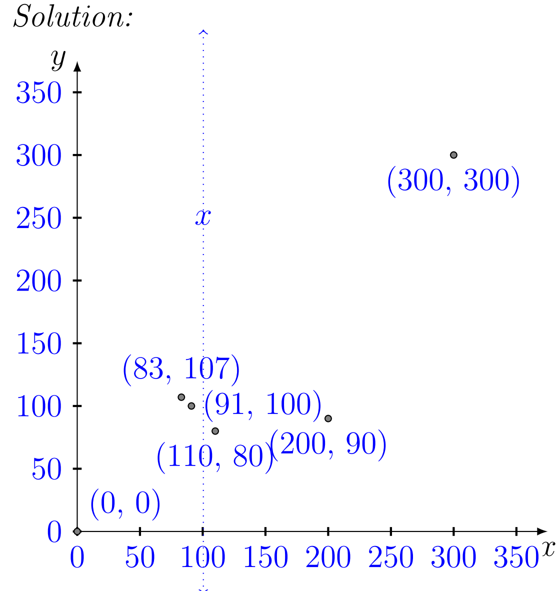
Notes: (i) Every answer has to be justified unless otherwise stated. Show your work! All performance estimates (running times, etc) should be in asymptotic notation unless otherwise noted. (ii) You may use any theorem/property/fact proven in class or in the textbook. You do not need to re-prove any of them.

1. The closest pair of points algorithm (in 2 dimensions) asks for the 2 points (x_1, y_1), (x_2, y_2) such that the Euclidean distance between the points is as small as possible. The Euclidean distance between the points (x_1, y_1) and (x_2, y_2) is $\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$.

   (a) Suppose the input is the set $\{(0, 0), (83, 107), (91, 100), (110, 80), (200, 90), (300, 300)\}$. Plot the points in this set.

   Which pair of points in this dataset is the closest, as measured by their Euclidean distance? List the two points AND the Euclidean distance between them. (Note: You do not need to compute square roots. You may express your answer as e.g. $\sqrt{130}$).

   Solution:

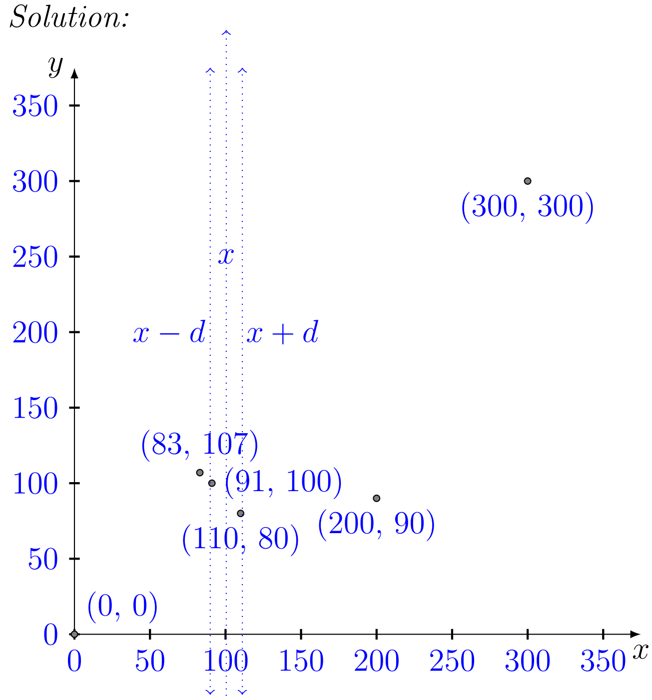


   Closest pair: (83, 107), (91, 100) with a Euclidean distance of $\sqrt{113}$

(b) The algorithm presented in class sorts the input set by x coordinate, and computes the equation of a vertical line dividing the left half of the points from the right half. It then recursively computes the closest pair of points on the left side, and the closest pair of points on the right side.

i. For the dataset in part (a), the dividing line is $x = 100.5$.
Let d_L be the smallest distance between two points on the left side of this line, and d_R be the smallest distance between two points on the right side of this line. What are the values of d_L and d_R? (You don't have to run the algorithm to answer this question, as long as you answer the question correctly.)
Solution:
$d_L = \sqrt{113}$ between points $(83, 107)$ and $(91, 100)$
$d_R = \sqrt{8200}$ between points $(110, 80)$ and $(200, 90)$

ii. On your plot, draw the two vertical lines at distance $d = \min\{d_L, d_R\}$ from the dividing line. Which points are in the strip between two lines? List them in increasing order of their y coordinates. Call this list L.
Solution:



$d = \sqrt{113}$
$L = \{(110, 80), (91, 100)\}$

iii. The closest pair algorithm from class computes the distance from each point (x, y) in the list to the 7 points coming after it in the list (of course, if there are fewer than 7 points that come after (x, y) in the list, you just compute the distance to all of the points coming after it). If you did these computations for

the list L you just generated, what would be the smallest distance computed? Based on this answer, and your answers to the previous questions, what would the algorithm return as the smallest distance between two pairs of points in the original dataset?
Solution: $d_M = \sqrt{761}$ between the two points in $L = \{(110, 80), (91, 100)\}$
The algorithm would return $min(d_L, d_M, d_R) = \sqrt{113}$

(c) Suppose you ran the closest pair of points algorithm on a set S of n points and it returned the closest pair of points, and the distance d between them. Suppose someone then gave you an additional point, (x', y') to insert into set S, and asked you for the closest pair of points in the modified set. One possible way for you to answer this question would be to re-run the closest pair algorithm from scratch on the modified dataset. This would take time $O(n \log n)$. What is a simple way to answer the question in time $O(n)$, without re-running the whole algorithm?

(An interesting and difficult question, which you do NOT have to answer, would be to try and figure out whether you could answer this question *even* faster, by saving some of the information that you computed when you ran the algorithm on the original dataset, in appropriate data structures. If you want to know more, ask Prof. Aronov, who can tell you a LOT about it.)

Solution: Compute the distance between (x', y') with all other points in S, and store the minimum distance from (x', y') to any other point. Return the minimum of d and the stored minimum for (x', y').

2. Define the M-distance between two points in 2 dimensions, (x_1, y_1) and (x_2, y_2), to be $\min\{|x_1 - x_2|, |y_1 - y_2|\}$. Describe a *simple* $O(n \log n)$ algorithm that takes as input a set of points $\{(x_1, y_1), \ldots, (x_n, y_n)\}$, and computes the "closest" pair of points, as measured by their M-distance.

Solution:

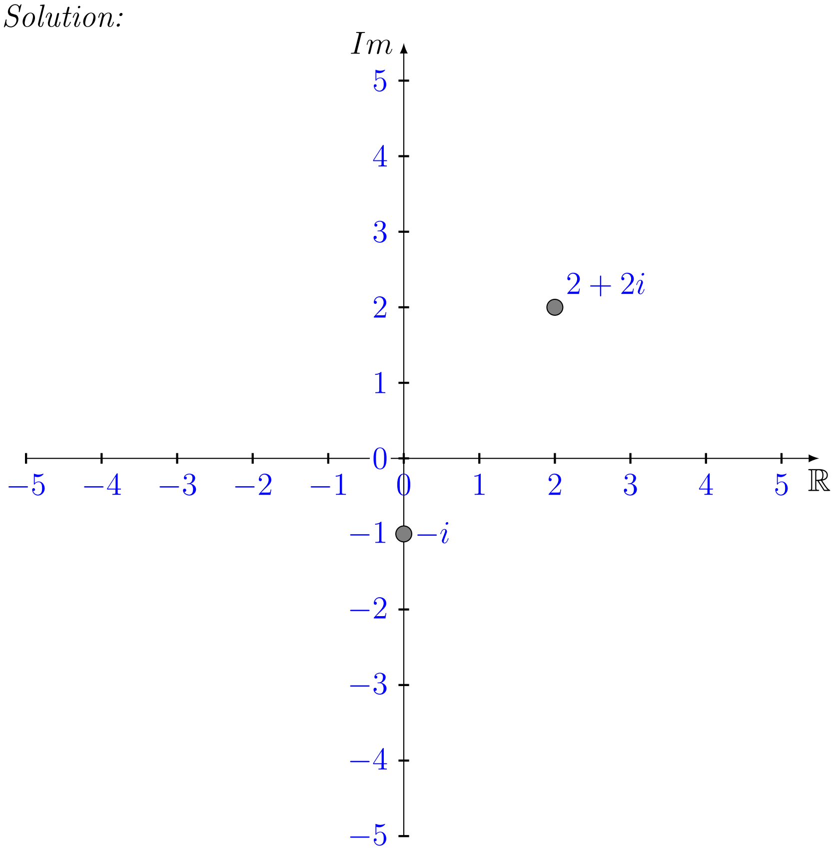
1. Sort the points by increasing x-coordinates
2. Compute the minimum Δx of consecutive points
3. Sort the points by increasing y-coordinates
4. Compute the minimum Δy of consecutive points
5. Return the minimum of the two

Since we are searching for the minimum Δx or Δy, we can linearize the points by sorting them and then perform a linear scan to find the minimum change. By sorting the points, we are assured that for any z_k, $z_{k+1} - z_k \leq z_{k+2} - z_k$, so we only need to look at consecutive points. The sorting takes $O(n \log n)$ and the linear scan takes $O(n)$ producing a total runtime of $O(n \log n)$.

3. In order to understand the FFT algorithm, you need to know a something about complex numbers. The basic information you need is given to you in Figure 2.6 of the text. The following exercises are designed to help you understand that information.

   (a) The first part of the figure shows how you can plot a complex number $a + bi$ as the point (a, b) on the complex plane. Plot the numbers $2 + 2i$ and $-i$ as points on the complex plane.

   Solution:



   (b) We can rewrite the complex number $a + bi$ as $r(\cos\theta + i\sin\theta)$ where $r = \sqrt{a^2 + b^2}$. The value θ is the measure of an angle, expressed in radians, such that $0 \le \theta \le 2\pi$, and θ satisfies $\cos\theta = a/r$. Also, $\sin\theta = b/r$.

   Fact: $r(\cos\theta + i\sin\theta) = e^{i\theta}$.

   Using these facts, we can write $a + bi$ as $re^{i\theta}$, for the appropriate values of r and θ.

   For example, to write the number $2 + 2i$ in the form $r(\cos\theta + i\sin\theta)$, we calculate $r = \sqrt{2^2 + 2^2} = \sqrt{8} = 2\sqrt{2}$ and $\theta = \arccos(2/2\sqrt{2}) = \arccos(1/\sqrt{2}) = \pi/4$. Therefore, $2 + 2i$ equals $2\sqrt{2}e^{i\pi/4}$.

   Write $3 - 3i$ in the form $re^{\theta i}$.

   Solution: $r = \sqrt{3^2 + (-3)^2} = \sqrt{18} = 3\sqrt{2}$. $\theta = \arcsin(-3/3\sqrt{2}) = \arcsin(-1/\sqrt{2}) = -\pi/4 = 7\pi/4$. Therefore $3 - 3i = 3\sqrt{2}e^{i7\pi/4}$

Note: The point $3 - 3i$ lies in the 4th quadrant, so if an arccos was used, the sign should be negated

(c) The product of two complex numbers, $r_1 e^{i\theta_1}$ and $r_2 e^{i\theta_2}$, is equal to $r_1 r_2 e^{i((\theta_1+\theta_2) \bmod 2\pi)}$.

That is, multipication works almost the same way as it would with real numbers. The one difference is that we want the θ in the exponent to be between 0 and 2π, so we reduce it mod 2π to be either 0 or a positive value that is less than 2π (which doesn't affect the value, since $\sin\theta$ and $\cos\theta$ are not affected if you reduce θ modulo 2π).

Working in polar coordinates, this implies that the product of two complex numbers expressed in polar coordinates, (r_1, θ_1) and (r_2, θ_2), is $(r_1 r_2, (\theta_1 + \theta_2) \bmod 2\pi)$, in polar coordinates. That is, you multiply the r's, and you add the θ's (and reduce the result mod 2π).

Calculate the product of the following 2 complex numbers expressed in polar coordinates: $(-1, \pi/4)$ and $(3, 3\pi/4)$. Express your answer in polar coordinates.

Solution: $(-1, \pi/4) \cdot (3, 3\pi/4) = ((-1) \cdot (3), (\pi/4) + (3\pi/4) \bmod 2\pi) = (-3, \pi)$
Reducing the resulting θ mod 2π doesn't change the resulting answer because 2π is a full rotation on the unit circle, so the end result is the same. For this reason, $\theta = \theta + 2\pi k$ with $k \in \mathbb{Z}$

(d) By the previous multiplication facts, if we take a complex number expressed in polar coordinates, (r, θ), and raise it to the kth power, the result is $(r^k, k\theta \bmod 2\pi)$.

Working in polar coordinates, what is $(1, \pi/2)$ raised to the 4th power?

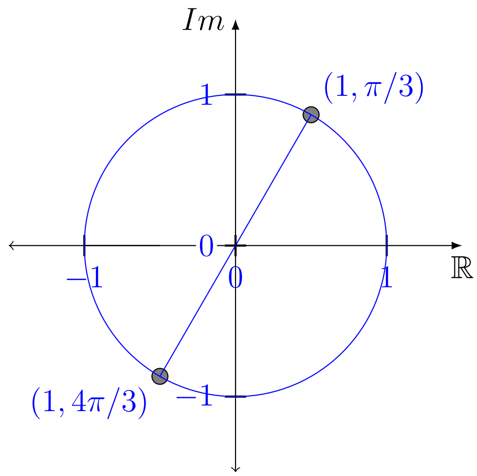
Solution: $(1, \pi/2)^4 = ((1)^4, (4) \cdot (\pi/2) \bmod 2\pi) = (1, 0)$

(e) The number -1 expressed in polar coordinates is $(1, \pi)$. Therefore, in polar coordintes, $-(r, \theta)$ is equal to the product of (r, θ) and $(1, \pi)$, which is $(r, (\theta + \pi) \bmod 2\pi)$.

Rewrite $-(1, \pi/3)$ (expressed in polar coordinates), so that it is in the form (r, θ) for some positive r. Draw the unit circle in the complex plane and plot the resulting point, together with the point $(1, \pi/3)$.

Draw a line between the two points. (It should go through the center of the circle, since the two points are opposite each other on the unit circle.)

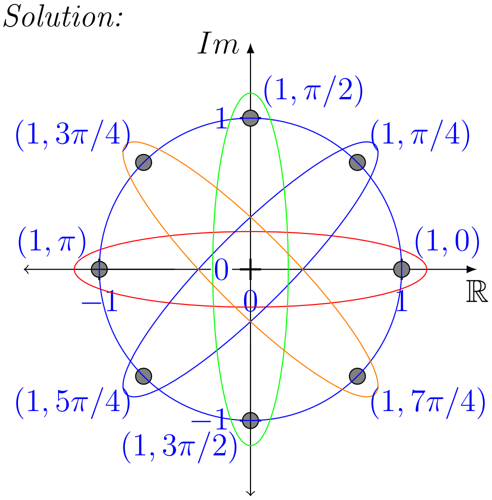
Solution: $-(1, \pi/3) = (1, \pi/3 + \pi \bmod 2\pi) = (1, 4\pi/3)$



(f) Draw the unit circle on the complex plane, and plot 8 evenly spaced points, the first one at the point $(1, 0)$ (in polar coordinates). These 8 points correspond to the 8 complex numbers satisfying the equation $x^8 = 1$. That is, they are the "8th roots of unity".

The 8th roots of unity can be paired up into 4 pairs, where each pair is $\pm x$ for some x, corresponding to opposite points on the unit circle. List the 8th roots of unity in polar coordinates, circling the pairs that are equal to $\pm x$ for some x.

Solution:

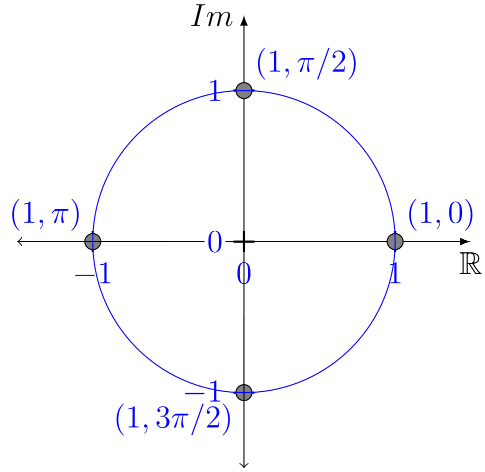


(g) Take the 8th roots of unity and square each of them. What are the resulting 4 complex numbers? Give the answer in polar coordinates and plot the points on the unit circle.

Solution:

We need to compute the set of squares of the following numbers:
$\{(1, 0), (1, \pi/4), (1, \pi/2), (1, 3\pi/4), (1, \pi), (1, 5\pi/4), (1, 3\pi/2), (1, 7\pi/4)\}$
Squaring each item produces:
$\{(1, 0), (1, \pi/2), (1, \pi), (1, 6\pi/4), (1, 0), (1, \pi/2), (1, \pi), (1, 6\pi/4)\}$

And when we remove duplicates:
$\{(1,0),(1,\pi/2),(1,\pi),(1,6\pi/4)\}$



(h) More generally, for any positive integer n, the nth roots of unity correspond to the n evenly spaced points on the unit circle, in the complex plane (where the first point is $(1,0)$, in polar coordinates). In other words, they are the complex numbers that in polar coordinates are $(1,\frac{2\pi k}{n})$, for $k=0,1,2,\ldots,n-1$.

A *primitive* nth root of unity is a complex number ω such that $\omega^n = 1$ (it is an nth root of unity) and for $j \in \{1,2,\ldots,n-1\}$, $\omega^j \neq 1$. The complex number $e^{i2\pi/n}$ is a primitive nth root of unity.

Which of the following statements is true about the complex number, written in polar coordinates, $(1,\frac{10\pi}{7})$?

i. It is a 7th root of unity.
Solution: True, $(1,\frac{10\pi}{7})^7 = (1, 7\cdot\frac{10\pi}{7} \bmod 2\pi) = (1,0) = 1$

ii. It is a 14th root of unity.
Solution: True, $(1,\frac{10\pi}{7})^{14} = (1, 14\cdot\frac{10\pi}{7} \bmod 2\pi) = (1,0) = 1$

iii. It is a primitive 14th root of unity.
Solution: False, as shown in 3hi., there is a $k < 14$ such that $(1,\frac{10\pi}{7})^k = 1$

(i) True or false: The 4th roots of unity are the numbers you get by squaring the 8th roots of unity.

Solution: True, we can look at the results of squaring the 8th roots of unity from 3(g). The resulting values are exactly the 4th roots of unity. More generally, when n is a multiple of 2, squaring the nth roots of unity produces the $n/2$ roots of unity.

4. The body of the for loop in Figure 2.7 uses the values ω^j. However, if we represent these values in the form $e^{i\theta}$, we will have a hard time executing the body of the loop, which requires us to add values.

Instead, we can represent the values ω^j in the form $a + bi$. We can calculate these representations by first expressing ω in the form $a + bi$, and then calculating the value of ω^j for $j = 2, \ldots, n$, using the fact that $\omega^j = \omega * \omega^{j-1}$.

For $\omega = e^{i2\pi/8}$, which is equal to $\cos\frac{\pi}{4} + i\sin\frac{\pi}{4} = \frac{1}{\sqrt{2}} + \frac{1}{\sqrt{2}}i$, we get the following:

- $\omega^0 = 1$
- $\omega^1 = \frac{1}{\sqrt{2}} + \frac{1}{\sqrt{2}}i$
- $\omega^2 = i$
- $\omega^3 = -\frac{1}{\sqrt{2}} + \frac{1}{\sqrt{2}}i$
- $\omega^4 = -1$
- $\omega^5 = -\frac{1}{\sqrt{2}} - \frac{1}{\sqrt{2}}i$
- $\omega^6 = -i$

(a) Continuing with the above, what is ω^7?

Solution: $w^7 = w^6 \cdot w^1 = -i \cdot (\frac{1}{\sqrt{2}} + \frac{1}{\sqrt{2}}i) = \frac{1}{\sqrt{2}} - \frac{1}{\sqrt{2}}i$

(b) For $k \geq 8$, ω^k is equal to ω^t, for some $t \in \{0, 1, \ldots, 7\}$. Express t as a function of k.

Solution: The terms start repeating from $t = 0$ starting when $k = 8$ so $t = k \bmod 8$

5. Consider the polynomial formulation of the FFT algorithm in Figure 2.7 of the textbook. The input is a coefficient representation of a polynomial $A(x)$ of degree at most $n - 1$, where n is a power of 2, and an nth root of unity, ω. The output is a value representation of $A(\omega^0), A(\omega^1), \ldots, A(\omega^{n-1})$.

   Suppose we run the FFT algorithm in Figure 2.7 on the polynomial $A(x) = 1 + 3x + 5x^2 + 7x^3 + 8x^4 + 6x^5 + 3x^6 + 2x^7$, with $\omega = e^{i2\pi/8}$ (which is a primitive 8th root of unity).

   (a) At the top level, the algorithm will make two recursive calls. Specify the inputs to those two recursive calls. In each case, give both the polynomial (in its coefficient representation) and the relevant root of unity.

   Solution: The FFT algorithm splits its input coefficients into A_e, the even powered coefficients and A_o, the odd powered coefficients, and recursively calls FFT(A_e, w^2) and FFT(A_o, w^2)
   $A_e =< 1, 5, 8, 3 >, w^2 = i$
   $A_o =< 3, 7, 6, 2 >, w^2 = i$

   (b) The recursive call on A_e, ω^2 computes the value $A_e(x) = 1 + 5x + 8x^2 + 3x^3$ for $x = \omega^0, \omega^2, \omega^4, \omega^6$, that is, for $x = 1, i, -1, -i$. To determine these values by hand, instead of thinking about another recursive call, we'll just plug the values into

the coefficient representation of $A_e(x)$, using the fact that $i^2 = -1$, to get the following: $A_e(\omega^0) = A_e(1) = 1 + 5 * 1 + 8 * 1^2 + 3 * 1^3 = 17$, $A_e(\omega^2) = A_e(i) = 1 + 5i + 8(i^2) + 3(i^3) = -7 + 2i$ $A_e(\omega^4) = A_e(-1) = 1$ $A_e(\omega^6) = A_e(-i) = -7 - 2i$
Using this same approach, calculate the values of $A_o(\omega^0)$, $A_o(\omega^2)$, $A_o(\omega^4)$, $A_o(\omega^6)$.

Solution: $A_o(w^0) = A_o(1) = 3 + 7(1) + 6(1^2) + 2(1^3) = 18$
$A_o(w^2) = A_o(i) = 3 + 7(i) + 6(i^2) + 2(i^3) = -3 + 5i$
$A_o(w^4) = A_o(-1) = 3 + 7(-1) + 6((-1)^2) + 2((-1)^3) = 0$
$A_o(w^6) = A_o(-i) = 3 + 7(-i) + 6((-i)^2) + 2((-i)^3) = -3 - 5i$

(c) Using the values of ω^j and the values of A_e and A_o, computed or given above, execute the statement in the body of the for loop in Figure 2.7, to calculate the value of $A(\omega^3)$ and $A(\omega^7)$ for $A(x) = 1 + 3x + 5x^2 + 7x^3 + 8x^4 + 6x^5 + 3x^6 + 2x^7$, and $\omega = \frac{1}{\sqrt{2}} + \frac{1}{\sqrt{2}}i$. Express your answer in the form $a + bi$. Show your work.

Solution: $A(w^3) = A_e(w^6) + w^3 A_o(w^6) = (-7 - 2i) + (-\frac{1}{\sqrt{2}} + \frac{1}{\sqrt{2}}i)(-3 - 5i) = (-7 - 2i) + (\frac{8}{\sqrt{2}} + \frac{2}{\sqrt{2}}i) = (-7 - 2i) + (4\sqrt{2} + \sqrt{2}i) = (-7 + 4\sqrt{2}) + (-2 + \sqrt{2})i$

$A(w^7) = A_e(w^{14} = w^6) + w^7 A_o(w^{14} = w^6) = (-7 - 2i) + (\frac{1}{\sqrt{2}} - \frac{1}{\sqrt{2}}i)(-3 - 5i) = (-7 - 4\sqrt{2}) + (-2 - \sqrt{2})i$