COMP 761: Lecture 5 – Invariants and Monovariants

David Rolnick

September 14, 2020

Problem

Given n red points and n blue points in the plane, show that we can draw n non-intersecting line segments, each having one red endpoint and one blue endpoint.

(Please don't post your ideas in the chat just yet, we'll discuss the problem soon in class.)

Course Announcements

Course Announcements

- Reminder that the problem set is due on Friday

Course Announcements

- Reminder that the problem set is due on Friday
- Post in the Slack if looking for collaborators

Course Announcements

- Reminder that the problem set is due on Friday
- Post in the Slack if looking for collaborators
- Office hours - right after class! (stay in the Zoom room)

Problem 1

The numbers $1, 2, \ldots, 100$ are written on a blackboard. You may choose any two numbers a and b and erase them, replacing them with the single number $a + b$. After 99 steps, only a single number will be left. What are the possibilities for that number?

Problem 1

The numbers $1, 2, \ldots, 100$ are written on a blackboard. You may choose any two numbers a and b and erase them, replacing them with the single number $a + b$. After 99 steps, only a single number will be left. What are the possibilities for that number?

- We are replacing two numbers by their sum. What happens if we do this operation twice?

Problem 1

The numbers $1, 2, \ldots, 100$ are written on a blackboard. You may choose any two numbers a and b and erase them, replacing them with the single number $a + b$. After 99 steps, only a single number will be left. What are the possibilities for that number?

- We are replacing two numbers by their sum. What happens if we do this operation twice?
- We get the sum of three numbers.

Problem 1

The numbers $1, 2, \ldots, 100$ are written on a blackboard. You may choose any two numbers a and b and erase them, replacing them with the single number $a + b$. After 99 steps, only a single number will be left. What are the possibilities for that number?

- We are replacing two numbers by their sum. What happens if we do this operation twice?
- We get the sum of three numbers.
- In general?

Problem 1

The numbers $1, 2, \ldots, 100$ are written on a blackboard. You may choose any two numbers a and b and erase them, replacing them with the single number $a + b$. After 99 steps, only a single number will be left. What are the possibilities for that number?

- We are replacing two numbers by their sum. What happens if we do this operation twice?
- We get the sum of three numbers.
- In general?
- No matter the order in which we combine the numbers, we will eventually be left with the sum of all of them.

Invariants

Invariants

- An *invariant* is a quantity that remains unchanged.

Invariants

- An *invariant* is a quantity that remains unchanged.
- General problem-solving technique: Identify an invariant, calculate its initial value, conclude that its final value has to be the same.

Invariants

- An *invariant* is a quantity that remains unchanged.
- General problem-solving technique: Identify an invariant, calculate its initial value, conclude that its final value has to be the same.

When is this useful?

Invariants

- An *invariant* is a quantity that remains unchanged.
- General problem-solving technique: Identify an invariant, calculate its initial value, conclude that its final value has to be the same.

When is this useful?

- Working out what the final state must be, based on the value of the invariant.

Invariants

- An *invariant* is a quantity that remains unchanged.
- General problem-solving technique: Identify an invariant, calculate its initial value, conclude that its final value has to be the same.

When is this useful?

- Working out what the final state must be, based on the value of the invariant.
- Showing some final states are *impossible* because the invariant has a different value.

Proof of Problem 1

Proof of Problem 1

Observe that the sum of all the numbers on the blackboard is invariant under the operation described, since we are replacing two numbers by a single number that is their sum.

Proof of Problem 1

Observe that the sum of all the numbers on the blackboard is invariant under the operation described, since we are replacing two numbers by a single number that is their sum. Therefore, the final sum must be the same as the initial sum – and hence if there is a single number left, that number must be:

$$1 + 2 + \cdots + 100 = \frac{100(101)}{2} = 5050.$$

■

Problem 1b

What about if a and b get replaced by $a + b - 1$ at each step?

Problem 1b

What about if a and b get replaced by $a + b - 1$ at each step?

- Our invariant before was the sum of all the numbers.

Problem 1b

What about if a and b get replaced by $a + b - 1$ at each step?

- Our invariant before was the sum of all the numbers.
- How can we change that to make something that is invariant under the new operation?

Problem 1b

What about if *a* and *b* get replaced by $a + b - 1$ at each step?

- Our invariant before was the sum of all the numbers.
- How can we change that to make something that is invariant under the new operation?
- Since 1 is being taken away each step, we can add 1 back in at each step.

Problem 1b

What about if a and b get replaced by $a + b - 1$ at each step?

- Our invariant before was the sum of all the numbers.
- How can we change that to make something that is invariant under the new operation?
- Since 1 is being taken away each step, we can add 1 back in at each step.
- Sum of the numbers + number of steps already taken

Proof of Problem 1b

Proof of Problem 1b

We claim that the sum S of the numbers on the board, plus the number N of operations performed, is invariant through the process.

Proof of Problem 1b

We claim that the sum S of the numbers on the board, plus the number N of operations performed, is invariant through the process. As the operation replaces two numbers by one less than their sum, the overall sum S decreases by 1 each iteration.

Proof of Problem 1b

We claim that the sum S of the numbers on the board, plus the number N of operations performed, is invariant through the process. As the operation replaces two numbers by one less than their sum, the overall sum S decreases by 1 each iteration. Each operation also clearly increases N by 1, so $S + N$ stays constant.

Proof of Problem 1b

We claim that the sum S of the numbers on the board, plus the number N of operations performed, is invariant through the process. As the operation replaces two numbers by one less than their sum, the overall sum S decreases by 1 each iteration. Each operation also clearly increases N by 1, so $S + N$ stays constant.

Therefore, the final value of $S + N$ must be the same as the initial value. Initially $N = 0$ and

$$S = 1 + 2 + \cdots + 100 = \frac{100(101)}{2} = 5050.$$

Proof of Problem 1b

We claim that the sum S of the numbers on the board, plus the number N of operations performed, is invariant through the process. As the operation replaces two numbers by one less than their sum, the overall sum S decreases by 1 each iteration. Each operation also clearly increases N by 1, so $S + N$ stays constant.

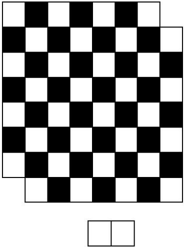
Therefore, the final value of $S + N$ must be the same as the initial value. Initially $N = 0$ and

$$S = 1 + 2 + \cdots + 100 = \frac{100(101)}{2} = 5050.$$

Hence, after 99 operations, we must have $5050 = S + N = S + 99$, and therefore $S = 4951$. As only one number is left, this number must equal 4951. ■

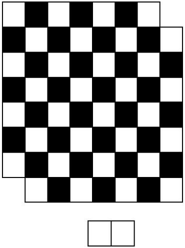
Problem 2

Two opposite corners are removed from an 8×8 chessboard. Is it possible to cover it with 1×2 dominoes?



Problem 2

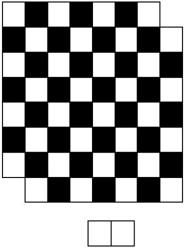
Two opposite corners are removed from an 8×8 chessboard. Is it possible to cover it with 1×2 dominoes?



- Each domino covers 2 squares. But the total number of squares to cover is 62, which is also even. Anything else we can use?

Problem 2

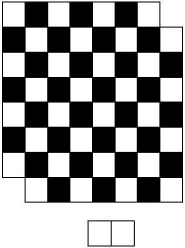
Two opposite corners are removed from an 8×8 chessboard. Is it possible to cover it with 1×2 dominoes?



- Each domino covers 2 squares. But the total number of squares to cover is 62, which is also even. Anything else we can use?
- Aha! Each domino covers 2 *adjacent* squares, so 1 white, 1 black.

Problem 2

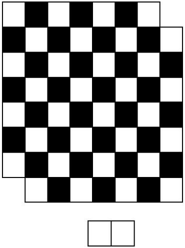
Two opposite corners are removed from an 8×8 chessboard. Is it possible to cover it with 1×2 dominoes?



- Each domino covers 2 squares. But the total number of squares to cover is 62, which is also even. Anything else we can use?
- Aha! Each domino covers 2 *adjacent* squares, so 1 white, 1 black.
- Therefore, each covering by dominoes would have even numbers of white and black squares.

Problem 2

Two opposite corners are removed from an 8×8 chessboard. Is it possible to cover it with 1×2 dominoes?



- Each domino covers 2 squares. But the total number of squares to cover is 62, which is also even. Anything else we can use?
- Aha! Each domino covers 2 *adjacent* squares, so 1 white, 1 black.
- Therefore, each covering by dominoes would have even numbers of white and black squares.
- But there are 30 black and 32 white squares, so this is impossible.

Problem 3

You are given a 100×100 array of lights. You can flip all the lights in any row or column, so that any lights that were off get turned on, and vice versa. Initially, one light in the grid is on. Can you flip the rows and columns in such a way that all the lights are turned on?

Problem 3

You are given a 100×100 array of lights. You can flip all the lights in any row or column, so that any lights that were off get turned on, and vice versa. Initially, one light in the grid is on. Can you flip the rows and columns in such a way that all the lights are turned on?

- What's a good first step in a problem with large numbers like this?

Problem 3

You are given a 100×100 array of lights. You can flip all the lights in any row or column, so that any lights that were off get turned on, and vice versa. Initially, one light in the grid is on. Can you flip the rows and columns in such a way that all the lights are turned on?

- What's a good first step in a problem with large numbers like this?
- Let's try out small examples. Hopefully it works for a 4×4 grid:

Problem 3

You are given a 100×100 array of lights. You can flip all the lights in any row or column, so that any lights that were off get turned on, and vice versa. Initially, one light in the grid is on. Can you flip the rows and columns in such a way that all the lights are turned on?

- What's a good first step in a problem with large numbers like this?
- Let's try out small examples. Hopefully it works for a 4×4 grid:

0	1	0	0
0	0	0	0
0	0	0	0
0	0	0	0

Problem 3

You are given a 100×100 array of lights. You can flip all the lights in any row or column, so that any lights that were off get turned on, and vice versa. Initially, one light in the grid is on. Can you flip the rows and columns in such a way that all the lights are turned on?

- What's a good first step in a problem with large numbers like this?
- Let's try out small examples. Hopefully it works for a 4×4 grid:

0	1	0	0
0	0	0	0
0	0	0	0
0	0	0	0

$\Rightarrow$

0	0	0	0
0	1	0	0
0	1	0	0
0	1	0	0

Problem 3

You are given a 100×100 array of lights. You can flip all the lights in any row or column, so that any lights that were off get turned on, and vice versa. Initially, one light in the grid is on. Can you flip the rows and columns in such a way that all the lights are turned on?

- What's a good first step in a problem with large numbers like this?
- Let's try out small examples. Hopefully it works for a 4×4 grid:

0	1	0	0
0	0	0	0
0	0	0	0
0	0	0	0

$\Rightarrow$

0	0	0	0
0	1	0	0
0	1	0	0
0	1	0	0

$\Rightarrow$

0	0	1	0
0	1	1	0
0	1	1	0
0	1	1	0

$\Rightarrow$

0	0	1	0
1	0	0	1
0	1	1	0
0	1	1	0

Problem 3

You are given a 100×100 array of lights. You can flip all the lights in any row or column, so that any lights that were off get turned on, and vice versa. Initially, one light in the grid is on. Can you flip the rows and columns in such a way that all the lights are turned on?

- What's a good first step in a problem with large numbers like this?
- Let's try out small examples. Hopefully it works for a 4×4 grid:

0	1	0	0
0	0	0	0
0	0	0	0
0	0	0	0

$\Rightarrow$

0	0	0	0
0	1	0	0
0	1	0	0
0	1	0	0

$\Rightarrow$

0	0	1	0
0	1	1	0
0	1	1	0
0	1	1	0

$\Rightarrow$

0	0	1	0
1	0	0	1
0	1	1	0
0	1	1	0

- Notice anything?

Problem 3

You are given a 100×100 array of lights. You can flip all the lights in any row or column, so that any lights that were off get turned on, and vice versa. Initially, one light in the grid is on. Can you flip the rows and columns in such a way that all the lights are turned on?

- What's a good first step in a problem with large numbers like this?
- Let's try out small examples. Hopefully it works for a 4×4 grid:

0	1	0	0
0	0	0	0
0	0	0	0
0	0	0	0

$\Rightarrow$

0	0	0	0
0	1	0	0
0	1	0	0
0	1	0	0

$\Rightarrow$

0	0	1	0
0	1	1	0
0	1	1	0
0	1	1	0

$\Rightarrow$

0	0	1	0
1	0	0	1
0	1	1	0
0	1	1	0

- Notice anything?
- We seem to always have an odd number of 1s.

Problem 3

You are given a 100×100 array of lights. You can flip all the lights in any row or column, so that any lights that were off get turned on, and vice versa. Initially, one light in the grid is on. Can you flip the rows and columns in such a way that all the lights are turned on?

- What's a good first step in a problem with large numbers like this?
- Let's try out small examples. Hopefully it works for a 4×4 grid:

0	1	0	0
0	0	0	0
0	0	0	0
0	0	0	0

$\Rightarrow$

0	0	0	0
0	1	0	0
0	1	0	0
0	1	0	0

$\Rightarrow$

0	0	1	0
0	1	1	0
0	1	1	0
0	1	1	0

$\Rightarrow$

0	0	1	0
1	0	0	1
0	1	1	0
0	1	1	0

- Notice anything?
- We seem to always have an odd number of 1s.
- Is this always true? If so, why?

Problem 3

Problem 3

- Yes, when we flip a row or column, we have:

# of lights before a flip	0	1	2	3	4
# of lights after	4	3	2	1	0

Problem 3

- Yes, when we flip a row or column, we have:

# of lights before a flip	0	1	2	3	4
# of lights after	4	3	2	1	0

- How about for a 100×100 grid?

Problem 3

- Yes, when we flip a row or column, we have:

# of lights before a flip	0	1	2	3	4
# of lights after	4	3	2	1	0

- How about for a 100×100 grid?
- We are changing n lights in a row or column to $100 - n$ lights.

Problem 3

- Yes, when we flip a row or column, we have:

# of lights before a flip	0	1	2	3	4
# of lights after	4	3	2	1	0

- How about for a 100×100 grid?
- We are changing n lights in a row or column to $100 - n$ lights.
- Since n and $100 - n$ are either both even or both odd, we don't change the parity of the total number of lights.

Problem 3

- Yes, when we flip a row or column, we have:

# of lights before a flip	0	1	2	3	4
# of lights after	4	3	2	1	0

- How about for a 100×100 grid?
- We are changing n lights in a row or column to $100 - n$ lights.
- Since n and $100 - n$ are either both even or both odd, we don't change the parity of the total number of lights.
- How do we finish?

Problem 3

- Yes, when we flip a row or column, we have:

# of lights before a flip	0	1	2	3	4
# of lights after	4	3	2	1	0

- How about for a 100×100 grid?
- We are changing n lights in a row or column to $100 - n$ lights.
- Since n and $100 - n$ are either both even or both odd, we don't change the parity of the total number of lights.
- How do we finish?
- We started out with 1 light on. Therefore we must always have an odd number of lights on.

Problem 3

- Yes, when we flip a row or column, we have:

# of lights before a flip	0	1	2	3	4
# of lights after	4	3	2	1	0

- How about for a 100×100 grid?
- We are changing n lights in a row or column to $100 - n$ lights.
- Since n and $100 - n$ are either both even or both odd, we don't change the parity of the total number of lights.
- How do we finish?
- We started out with 1 light on. Therefore we must always have an odd number of lights on.
- But 100^2 is even, so we can't get all the lights on.

Problem 4

In the parliament of Freedonia, certain pairs of members are enemies with each other. Each member has a maximum of 3 enemies. Prove that the house can be separated into 2 houses, so that each member has at most 1 enemy in their own house.

Problem 4

In the parliament of Freedonia, certain pairs of members are enemies with each other. Each member has a maximum of 3 enemies. Prove that the house can be separated into 2 houses, so that each member has at most 1 enemy in their own house.

- Let's try separating the two houses and see what can go wrong.

Problem 4

In the parliament of Freedonia, certain pairs of members are enemies with each other. Each member has a maximum of 3 enemies. Prove that the house can be separated into 2 houses, so that each member has at most 1 enemy in their own house.

- Let's try separating the two houses and see what can go wrong.
- Suppose there is someone with at least 2 enemies in their own house. What do we do?

Problem 4

In the parliament of Freedonia, certain pairs of members are enemies with each other. Each member has a maximum of 3 enemies. Prove that the house can be separated into 2 houses, so that each member has at most 1 enemy in their own house.

- Let's try separating the two houses and see what can go wrong.
- Suppose there is someone with at least 2 enemies in their own house. What do we do?
- We can move them to the other house. Does this help?

Problem 4

In the parliament of Freedonia, certain pairs of members are enemies with each other. Each member has a maximum of 3 enemies. Prove that the house can be separated into 2 houses, so that each member has at most 1 enemy in their own house.

- Let's try separating the two houses and see what can go wrong.
- Suppose there is someone with at least 2 enemies in their own house. What do we do?
- We can move them to the other house. Does this help?
- Yes! Since they have at most 3 total enemies, they now have at most 1 enemy in their house.

Problem 4

In the parliament of Freedonia, certain pairs of members are enemies with each other. Each member has a maximum of 3 enemies. Prove that the house can be separated into 2 houses, so that each member has at most 1 enemy in their own house.

- Let's try separating the two houses and see what can go wrong.
- Suppose there is someone with at least 2 enemies in their own house. What do we do?
- We can move them to the other house. Does this help?
- Yes! Since they have at most 3 total enemies, they now have at most 1 enemy in their house.
- Let's keep on doing this. Do we ever finish, or can it cycle?

Problem 4

In the parliament of Freedonia, certain pairs of members are enemies with each other. Each member has a maximum of 3 enemies. Prove that the house can be separated into 2 houses, so that each member has at most 1 enemy in their own house.

- Let's try separating the two houses and see what can go wrong.
- Suppose there is someone with at least 2 enemies in their own house. What do we do?
- We can move them to the other house. Does this help?
- Yes! Since they have at most 3 total enemies, they now have at most 1 enemy in their house.
- Let's keep on doing this. Do we ever finish, or can it cycle?
- It has to finish, since the total number of enemies within houses is going down each time.

Monovariants

Monovariants

- A *monovariant* is a quantity that changes only in one direction – either always up or always down.

Monovariants

- A *monovariant* is a quantity that changes only in one direction – either always up or always down.
- If the monovariant is bounded (for example, if it is decreasing, but has to be above 0), then the process has to stop.

Monovariants

- A *monovariant* is a quantity that changes only in one direction – either always up or always down.
- If the monovariant is bounded (for example, if it is decreasing, but has to be above 0), then the process has to stop.

When is this useful?

Monovariants

- A *monovariant* is a quantity that changes only in one direction – either always up or always down.
- If the monovariant is bounded (for example, if it is decreasing, but has to be above 0), then the process has to stop.

When is this useful?

- Proving that a process terminates.

Monovariants

- A *monovariant* is a quantity that changes only in one direction – either always up or always down.
- If the monovariant is bounded (for example, if it is decreasing, but has to be above 0), then the process has to stop.

When is this useful?

- Proving that a process terminates.
- Finding the particular state in which a process terminates (where the monovariant is maximized/minimized).

Monovariants

- A *monovariant* is a quantity that changes only in one direction – either always up or always down.
- If the monovariant is bounded (for example, if it is decreasing, but has to be above 0), then the process has to stop.

When is this useful?

- Proving that a process terminates.
- Finding the particular state in which a process terminates (where the monovariant is maximized/minimized).

Note: Be careful that your monovariant goes strictly up/down rather than staying the same sometimes. Or if it does, prove that it will eventually go up/down after enough moves.

Proof of Problem 4

Proof of Problem 4

We will describe an algorithm for separating into two houses, so that each member has at most one enemy in their house. Start by initializing the houses randomly. Let E_1 be the number of pairs of enemies in house 1, and E_2 similarly for house 2.

Proof of Problem 4

We will describe an algorithm for separating into two houses, so that each member has at most one enemy in their house. Start by initializing the houses randomly. Let E_1 be the number of pairs of enemies in house 1, and E_2 similarly for house 2.

We perform the following operation iteratively: If there is any member who has more than 1 enemy in their house, move them to the other house. We claim that each such move strictly decreases the sum $E_1 + E_2$.

Proof of Problem 4

We will describe an algorithm for separating into two houses, so that each member has at most one enemy in their house. Start by initializing the houses randomly. Let E_1 be the number of pairs of enemies in house 1, and E_2 similarly for house 2.

We perform the following operation iteratively: If there is any member who has more than 1 enemy in their house, move them to the other house. We claim that each such move strictly decreases the sum $E_1 + E_2$. Consider performing a move, without loss of generality moving a member M from house 1 to house 2.

Proof of Problem 4

We will describe an algorithm for separating into two houses, so that each member has at most one enemy in their house. Start by initializing the houses randomly. Let E_1 be the number of pairs of enemies in house 1, and E_2 similarly for house 2.

We perform the following operation iteratively: If there is any member who has more than 1 enemy in their house, move them to the other house. We claim that each such move strictly decreases the sum $E_1 + E_2$. Consider performing a move, without loss of generality moving a member M from house 1 to house 2. Since M had at least 2 enemies in house 1, E_1 must decrease by at least 2.

Proof of Problem 4

We will describe an algorithm for separating into two houses, so that each member has at most one enemy in their house. Start by initializing the houses randomly. Let E_1 be the number of pairs of enemies in house 1, and E_2 similarly for house 2.

We perform the following operation iteratively: If there is any member who has more than 1 enemy in their house, move them to the other house. We claim that each such move strictly decreases the sum $E_1 + E_2$. Consider performing a move, without loss of generality moving a member M from house 1 to house 2. Since M had at least 2 enemies in house 1, E_1 must decrease by at least 2. Because M has at most 3 total enemies, they can have at most 1 enemy in house 2. Therefore, E_2 can increase by at most 1.

Proof of Problem 4

We will describe an algorithm for separating into two houses, so that each member has at most one enemy in their house. Start by initializing the houses randomly. Let E_1 be the number of pairs of enemies in house 1, and E_2 similarly for house 2.

We perform the following operation iteratively: If there is any member who has more than 1 enemy in their house, move them to the other house. We claim that each such move strictly decreases the sum $E_1 + E_2$. Consider performing a move, without loss of generality moving a member M from house 1 to house 2. Since M had at least 2 enemies in house 1, E_1 must decrease by at least 2. Because M has at most 3 total enemies, they can have at most 1 enemy in house 2. Therefore, E_2 can increase by at most 1. Hence, $E_1 + E_2$ decreases by at least 1, proving the claim.

Proof of Problem 4

We will describe an algorithm for separating into two houses, so that each member has at most one enemy in their house. Start by initializing the houses randomly. Let E_1 be the number of pairs of enemies in house 1, and E_2 similarly for house 2.

We perform the following operation iteratively: If there is any member who has more than 1 enemy in their house, move them to the other house. We claim that each such move strictly decreases the sum $E_1 + E_2$. Consider performing a move, without loss of generality moving a member M from house 1 to house 2. Since M had at least 2 enemies in house 1, E_1 must decrease by at least 2. Because M has at most 3 total enemies, they can have at most 1 enemy in house 2. Therefore, E_2 can increase by at most 1. Hence, $E_1 + E_2$ decreases by at least 1, proving the claim.

Since $E_1 + E_2$ cannot decrease beyond 0, we conclude that the process must stop. Since no more moves can be made, every member must have at most 1 enemy in their house. ■

Problem 5

A traveler leaves city C_1 and goes to city C_2, which is the farthest away from C_1. From C_2 she travels to C_3, which is the farthest away from C_2, and so on. (Assume that there are never any ties for the farthest distance.) Prove that if C_1 and C_3 are different, then the traveler will never return to C_1.

Problem 5

A traveler leaves city C_1 and goes to city C_2, which is the farthest away from C_1. From C_2 she travels to C_3, which is the farthest away from C_2, and so on. (Assume that there are never any ties for the farthest distance.) Prove that if C_1 and C_3 are different, then the traveler will never return to C_1.

- Let's try an example.

Problem 5

A traveler leaves city C_1 and goes to city C_2, which is the farthest away from C_1. From C_2 she travels to C_3, which is the farthest away from C_2, and so on. (Assume that there are never any ties for the farthest distance.) Prove that if C_1 and C_3 are different, then the traveler will never return to C_1.

- Let's try an example.
- The traveler starts at C_1 and goes 100 miles to C_2.

Problem 5

A traveler leaves city C_1 and goes to city C_2, which is the farthest away from C_1. From C_2 she travels to C_3, which is the farthest away from C_2, and so on. (Assume that there are never any ties for the farthest distance.) Prove that if C_1 and C_3 are different, then the traveler will never return to C_1.

- Let's try an example.
- The traveler starts at C_1 and goes 100 miles to C_2.
- She now travels to C_3. How many miles shall we say it is?

Problem 5

A traveler leaves city C_1 and goes to city C_2, which is the farthest away from C_1. From C_2 she travels to C_3, which is the farthest away from C_2, and so on. (Assume that there are never any ties for the farthest distance.) Prove that if C_1 and C_3 are different, then the traveler will never return to C_1.

- Let's try an example.
- The traveler starts at C_1 and goes 100 miles to C_2.
- She now travels to C_3. How many miles shall we say it is?
- It has to be more than C_2C_1, or else C_1 would be further from C_2 than C_3 is. Let's say 150 miles.

Problem 5

A traveler leaves city C_1 and goes to city C_2, which is the farthest away from C_1. From C_2 she travels to C_3, which is the farthest away from C_2, and so on. (Assume that there are never any ties for the farthest distance.) Prove that if C_1 and C_3 are different, then the traveler will never return to C_1.

- Let's try an example.
- The traveler starts at C_1 and goes 100 miles to C_2.
- She now travels to C_3. How many miles shall we say it is?
- It has to be more than C_2C_1, or else C_1 would be further from C_2 than C_3 is. Let's say 150 miles.
- Now she goes to C_4, traveling 200 miles.

Problem 5

A traveler leaves city C_1 and goes to city C_2, which is the farthest away from C_1. From C_2 she travels to C_3, which is the farthest away from C_2, and so on. (Assume that there are never any ties for the farthest distance.) Prove that if C_1 and C_3 are different, then the traveler will never return to C_1.

- Let's try an example.
- The traveler starts at C_1 and goes 100 miles to C_2.
- She now travels to C_3. How many miles shall we say it is?
- It has to be more than C_2C_1, or else C_1 would be further from C_2 than C_3 is. Let's say 150 miles.
- Now she goes to C_4, traveling 200 miles.
- Can she come back to C_1 now?

Problem 5

A traveler leaves city C_1 and goes to city C_2, which is the farthest away from C_1. From C_2 she travels to C_3, which is the farthest away from C_2, and so on. (Assume that there are never any ties for the farthest distance.) Prove that if C_1 and C_3 are different, then the traveler will never return to C_1.

- Let's try an example.
- The traveler starts at C_1 and goes 100 miles to C_2.
- She now travels to C_3. How many miles shall we say it is?
- It has to be more than C_2C_1, or else C_1 would be further from C_2 than C_3 is. Let's say 150 miles.
- Now she goes to C_4, traveling 200 miles.
- Can she come back to C_1 now?
- No! The distance she traveled last is increasing each time.

Problem 5

A traveler leaves city C_1 and goes to city C_2, which is the farthest away from C_1. From C_2 she travels to C_3, which is the farthest away from C_2, and so on. (Assume that there are never any ties for the farthest distance.) Prove that if C_1 and C_3 are different, then the traveler will never return to C_1.

- Let's try an example.
- The traveler starts at C_1 and goes 100 miles to C_2.
- She now travels to C_3. How many miles shall we say it is?
- It has to be more than C_2C_1, or else C_1 would be further from C_2 than C_3 is. Let's say 150 miles.
- Now she goes to C_4, traveling 200 miles.
- Can she come back to C_1 now?
- No! The distance she traveled last is increasing each time.
- If she goes to C_n, then $C_nC_{n-1} > C_{n-1}C_{n-2} > \cdots > C_2C_1$.

Problem 5

A traveler leaves city C_1 and goes to city C_2, which is the farthest away from C_1. From C_2 she travels to C_3, which is the farthest away from C_2, and so on. (Assume that there are never any ties for the farthest distance.) Prove that if C_1 and C_3 are different, then the traveler will never return to C_1.

- Let's try an example.
- The traveler starts at C_1 and goes 100 miles to C_2.
- She now travels to C_3. How many miles shall we say it is?
- It has to be more than C_2C_1, or else C_1 would be further from C_2 than C_3 is. Let's say 150 miles.
- Now she goes to C_4, traveling 200 miles.
- Can she come back to C_1 now?
- No! The distance she traveled last is increasing each time.
- If she goes to C_n, then $C_nC_{n-1} > C_{n-1}C_{n-2} > \cdots > C_2C_1$.
- If $C_n = C_1$, then $C_1C_{n-1} > C_2C_1$, which means she wouldn't have gone to C_2 on the first move!

The extremal principle

The extremal principle

- An elegant way to write some monovariant proofs

The extremal principle

- An elegant way to write some monovariant proofs
- If you need to show that a process ends in a particular way

The extremal principle

- An elegant way to write some monovariant proofs
- If you need to show that a process ends in a particular way
- Find a monovariant, and suppose it's as big/small as possible

The extremal principle

- An elegant way to write some monovariant proofs
- If you need to show that a process ends in a particular way
- Find a monovariant, and suppose it's as big/small as possible
- Show that if what you want isn't true, then you can make the monovariant even bigger or smaller

The extremal principle

- An elegant way to write some monovariant proofs
- If you need to show that a process ends in a particular way
- Find a monovariant, and suppose it's as big/small as possible
- Show that if what you want isn't true, then you can make the monovariant even bigger or smaller

The extremal principle

- An elegant way to write some monovariant proofs
- If you need to show that a process ends in a particular way
- Find a monovariant, and suppose it's as big/small as possible
- Show that if what you want isn't true, then you can make the monovariant even bigger or smaller
- (This is clearer with an example)

Reminder of Problem 4

In the parliament of Freedonia, certain pairs of members are enemies with each other. Each member has a maximum of 3 enemies. Prove that the house can be separated into 2 houses, so that each member has at most 1 enemy in their own house.

Alternate proof of Problem 4

Alternate proof of Problem 4

We will describe an algorithm for separating into two houses, so that each member has at most one enemy in their house. Start by initializing the houses randomly. Let E_1 be the number of pairs of enemies in house 1, and E_2 similarly for house 2.

Alternate proof of Problem 4

We will describe an algorithm for separating into two houses, so that each member has at most one enemy in their house. Start by initializing the houses randomly. Let E_1 be the number of pairs of enemies in house 1, and E_2 similarly for house 2. Consider the partition into house 1 and house 2 such that $E_1 + E_2$ is minimized.

Alternate proof of Problem 4

We will describe an algorithm for separating into two houses, so that each member has at most one enemy in their house. Start by initializing the houses randomly. Let E_1 be the number of pairs of enemies in house 1, and E_2 similarly for house 2. Consider the partition into house 1 and house 2 such that $E_1 + E_2$ is minimized. We claim that every member has at most 1 enemy in their house.

Alternate proof of Problem 4

We will describe an algorithm for separating into two houses, so that each member has at most one enemy in their house. Start by initializing the houses randomly. Let E_1 be the number of pairs of enemies in house 1, and E_2 similarly for house 2. Consider the partition into house 1 and house 2 such that $E_1 + E_2$ is minimized. We claim that every member has at most 1 enemy in their house.

Suppose towards contradiction that this is not true. Then, consider some member M (without loss of generality in house 1) who has more than 1 enemy in their house.

Alternate proof of Problem 4

We will describe an algorithm for separating into two houses, so that each member has at most one enemy in their house. Start by initializing the houses randomly. Let E_1 be the number of pairs of enemies in house 1, and E_2 similarly for house 2. Consider the partition into house 1 and house 2 such that $E_1 + E_2$ is minimized. We claim that every member has at most 1 enemy in their house.

Suppose towards contradiction that this is not true. Then, consider some member M (without loss of generality in house 1) who has more than 1 enemy in their house. Let us move M to house 2.

Alternate proof of Problem 4

We will describe an algorithm for separating into two houses, so that each member has at most one enemy in their house. Start by initializing the houses randomly. Let E_1 be the number of pairs of enemies in house 1, and E_2 similarly for house 2. Consider the partition into house 1 and house 2 such that $E_1 + E_2$ is minimized. We claim that every member has at most 1 enemy in their house.

Suppose towards contradiction that this is not true. Then, consider some member M (without loss of generality in house 1) who has more than 1 enemy in their house. Let us move M to house 2. Since M had at least 2 enemies in house 1, E_1 must decrease by at least 2.

Alternate proof of Problem 4

We will describe an algorithm for separating into two houses, so that each member has at most one enemy in their house. Start by initializing the houses randomly. Let E_1 be the number of pairs of enemies in house 1, and E_2 similarly for house 2. Consider the partition into house 1 and house 2 such that $E_1 + E_2$ is minimized. We claim that every member has at most 1 enemy in their house.

Suppose towards contradiction that this is not true. Then, consider some member M (without loss of generality in house 1) who has more than 1 enemy in their house. Let us move M to house 2. Since M had at least 2 enemies in house 1, E_1 must decrease by at least 2. Because M has at most 3 total enemies, they can have at most 1 enemy in house 2. Therefore, E_2 can increase by at most 1.

Alternate proof of Problem 4

We will describe an algorithm for separating into two houses, so that each member has at most one enemy in their house. Start by initializing the houses randomly. Let E_1 be the number of pairs of enemies in house 1, and E_2 similarly for house 2. Consider the partition into house 1 and house 2 such that $E_1 + E_2$ is minimized. We claim that every member has at most 1 enemy in their house.

Suppose towards contradiction that this is not true. Then, consider some member M (without loss of generality in house 1) who has more than 1 enemy in their house. Let us move M to house 2. Since M had at least 2 enemies in house 1, E_1 must decrease by at least 2. Because M has at most 3 total enemies, they can have at most 1 enemy in house 2. Therefore, E_2 can increase by at most 1.

Hence, $E_1 + E_2$ decreases by at least 1, which is a contradiction since we supposed it was minimal. We conclude that every member has at most 1 enemy in their house. ■

Problem 6

Given n red points and n blue points in the plane, show that we can draw n non-intersecting line segments, each having one red endpoint and one blue endpoint.

Problem 6

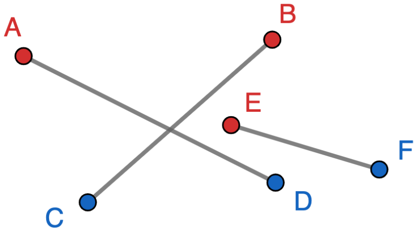
Given n red points and n blue points in the plane, show that we can draw n non-intersecting line segments, each having one red endpoint and one blue endpoint.

- Let try out a small example, say $n = 3$:

Problem 6

Given n red points and n blue points in the plane, show that we can draw n non-intersecting line segments, each having one red endpoint and one blue endpoint.

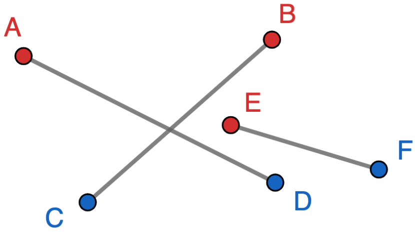
- Let try out a small example, say $n = 3$:



Problem 6

Given n red points and n blue points in the plane, show that we can draw n non-intersecting line segments, each having one red endpoint and one blue endpoint.

- Let try out a small example, say $n = 3$:

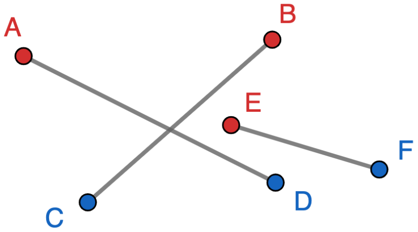


- Oops, there is a crossing! What can we do?

Problem 6

Given n red points and n blue points in the plane, show that we can draw n non-intersecting line segments, each having one red endpoint and one blue endpoint.

- Let try out a small example, say $n = 3$:

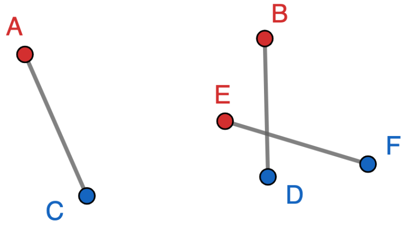


- Oops, there is a crossing! What can we do?
- Let's try uncrossing that one crossing.

Problem 6

Given n red points and n blue points in the plane, show that we can draw n non-intersecting line segments, each having one red endpoint and one blue endpoint.

- Let try out a small example, say $n = 3$:

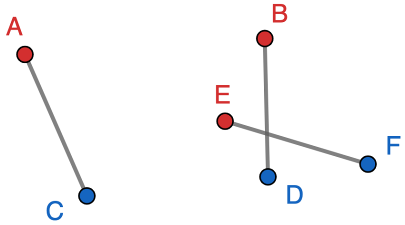


- Oops, there is a crossing! What can we do?
- Let's try uncrossing that one crossing.

Problem 6

Given n red points and n blue points in the plane, show that we can draw n non-intersecting line segments, each having one red endpoint and one blue endpoint.

- Let try out a small example, say $n = 3$:

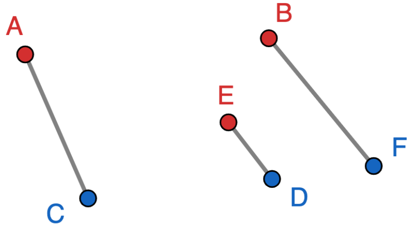


- Oops, there is a crossing! What can we do?
- Let's try uncrossing that one crossing.
- Hmm, another one. Maybe we can keep undoing them...

Problem 6

Given n red points and n blue points in the plane, show that we can draw n non-intersecting line segments, each having one red endpoint and one blue endpoint.

- Let try out a small example, say $n = 3$:

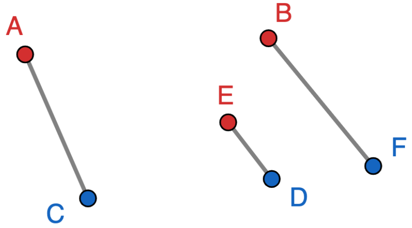


- Oops, there is a crossing! What can we do?
- Let's try uncrossing that one crossing.
- Hmm, another one. Maybe we can keep undoing them...

Problem 6

Given n red points and n blue points in the plane, show that we can draw n non-intersecting line segments, each having one red endpoint and one blue endpoint.

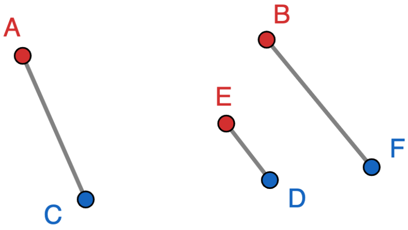
- Let try out a small example, say $n = 3$:



- Oops, there is a crossing! What can we do?
- Let's try uncrossing that one crossing.
- Hmm, another one. Maybe we can keep undoing them...
- Will uncrossing these crossings always terminate?

Problem 6

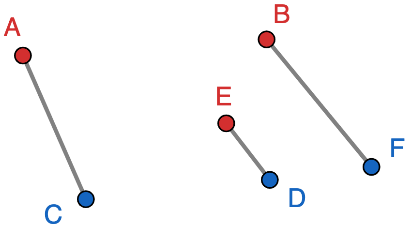
- Let try out a small example, say $n = 3$:



- Oops, there is a crossing! What can we do?
- Let's try uncrossing that one crossing.
- Hmm, another one. Maybe we can keep undoing them...
- Will uncrossing these crossings always terminate?

Problem 6

- Let try out a small example, say $n = 3$:



- Oops, there is a crossing! What can we do?
- Let's try uncrossing that one crossing.
- Hmm, another one. Maybe we can keep undoing them...
- Will uncrossing these crossings always terminate?
- Yes! The sum of the lengths of segments is going down each time.

Proof

Proof

Consider the matching between the red and blue points such that the total length of segments is minimized. We will show that no crossings can occur. Suppose towards contradiction that some segments AD and BC cross, where A, B are red and C, D are blue.

Proof

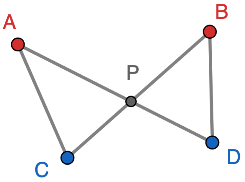
Consider the matching between the red and blue points such that the total length of segments is minimized. We will show that no crossings can occur. Suppose towards contradiction that some segments AD and BC cross, where A, B are red and C, D are blue.

Claim. The matching in which AC and BD are matched has lower total length of segments.

Proof

Consider the matching between the red and blue points such that the total length of segments is minimized. We will show that no crossings can occur. Suppose towards contradiction that some segments AD and BC cross, where A, B are red and C, D are blue.

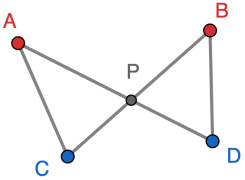
Claim. The matching in which AC and BD are matched has lower total length of segments.
Let P be the intersection of AD and BC, as shown.



Proof

Claim. The matching in which AC and BD are matched has lower total length of segments.

Let P be the intersection of AD and BC, as shown.

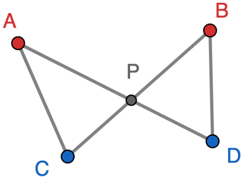


The Triangle Inequality states that in a triangle, the sum of the lengths of two of the sides must be more than the length of the third. Therefore:

$$AC < PA + PC \qquad \text{and} \qquad BD < PD + PB.$$

Proof

Claim. The matching in which AC and BD are matched has lower total length of segments.
Let P be the intersection of AD and BC, as shown.



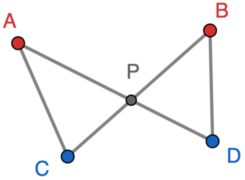
The Triangle Inequality states that in a triangle, the sum of the lengths of two of the sides must be more than the length of the third. Therefore:

$$AC < PA + PC \qquad \text{and} \qquad BD < PD + PB.$$

Summing these inequalities, we get $AC + BD < AD + BC$, which proves the claim.

Proof

Claim. The matching in which AC and BD are matched has lower total length of segments.
Let P be the intersection of AD and BC, as shown.



The Triangle Inequality states that in a triangle, the sum of the lengths of two of the sides must be more than the length of the third. Therefore:

$$AC < PA + PC \qquad \text{and} \qquad BD < PD + PB.$$

Summing these inequalities, we get $AC + BD < AD + BC$, which proves the claim.

Since we assumed the matching had minimal total length of segments, we have a contradiction, and conclude there are no intersections. ■

Next time!

Polynomials and algebra

Bonus problem

Consider a rectangular array with m rows and n columns whose entries are real numbers. It is permissible to reverse the signs of all the numbers in any row or column. Prove that after a number of these operations we can make the sum of numbers along each line (row or column) nonnegative.