Concepts used in this document

- ceiling function (round up) e.g., \( \lceil 1.5 \rceil = 2 \)

- Set, Subset

- \# of subsets that can be formed from a set of size \( n \) = \( 2^n \)

- \( \sum_{i=0}^{n} 2^i = 2^n - 1 \)

- Contrapositive, proof by contradiction
THE PIGEONHOLE PRINCIPLE

If $n$ holes are occupied by $n+1$ pigeons, then one hole is occupied by at least two pigeons.
THE PIGEONHOLE PRINCIPLE

If $n$ holes are occupied by $n+1$ pigeons, then one hole is occupied by at least two pigeons.
THE PIGEONHOLE PRINCIPLE

If \( n \) holes are occupied by \( >n \) pigeons, then one hole is occupied by at least two pigeons.
The Pigeonhole Principle

If $n$ holes are occupied by $p$ pigeons, $(p > n)$, then one hole is occupied by at least $\frac{p}{n}$ pigeons.
**THE PIGEONHOLE PRINCIPLE**

If $n$ holes are occupied by $p$ pigeons, ($p > n$), then one hole is occupied by at least $\frac{p}{n}$ pigeons.

$n = 6$ \{ at least 1.5 \}

$p = 9$ \{ at least 1.5 \}
THE PIGEONHOLE PRINCIPLE

If \( n \) holes are occupied by \( p \) pigeons, \((p>n)\), then one hole is occupied by at least \( \frac{p}{n} \) pigeons.

\[
\begin{align*}
n = 6 & \implies \text{at least } 1.5 \\
p = 9 & \implies \text{at least } 2 = \lceil \frac{p}{n} \rceil
\end{align*}
\]
THE PIGEONHOLE PRINCIPLE

If you have a pile of socks, of 3 colors, how many do you need to pick (randomly) to get a matching pair? (guaranteed)
THE PIGEONHOLE PRINCIPLE

If you have a pile of socks, of 3 colors, how many do you need to pick (randomly) to get a matching pair?
THE PIGEONHOLE PRINCIPLE

If you have a pile of socks, of 3 colors, how many do you need to pick (randomly) to get a matching pair?
THE PIGEONHOLE PRINCIPLE

If you have a pile of socks, of 3 colors, how many do you need to pick (randomly) to get a matching pair?
THE PIGEONHOLE PRINCIPLE

If you have a pile of socks, of 3 colors, how many do you need to pick (randomly) to get a matching pair?
THE PIGEONHOLE PRINCIPLE

If you have a pile of socks, of $n$ colors, how many do you need to pick (randomly) to get a matching pair?

$n+1$
THE PIGEONHOLE PRINCIPLE

If you have a pile of socks, of $n$ colors, how many do you need to pick (randomly) to get 3 matching?
THE PIGEONHOLE PRINCIPLE

If you have a pile of socks, of $n$ colors, how many do you need to pick (randomly) to get 3 matching?
THE PIGEONHOLE PRINCIPLE

If you have a pile of socks, of \( n \) colors, how many do you need to pick (randomly) to get 3 matching?
THE PIGEONHOLE PRINCIPLE

If you have a pile of socks, of \( n \) colors, how many do you need to pick (randomly) to get \( 3 \) matching?

Next pick does it.
THE PIGEONHOLE PRINCIPLE

If you have a pile of socks, of $n$ colors, how many do you need to pick (randomly) to get \(3\) matching?

\[2n+1\]

Worst scenario: pick 2 of each color = \(2n\) before getting a triple.
THE PIGEONHOLE PRINCIPLE

If you have a pile of socks, of $n$ colors, how many do you need to pick (randomly) to get $k$ of one type?

E.g., $n=5$
$k=6$
THE PIGEONHOLE PRINCIPLE

If you have a pile of socks, of $n$ colors, how many do you need to pick (randomly) to get $k$ of one type?

Worst scenario: pick $k-1$ of each color...
THE PIGEONHOLE PRINCIPLE

If you have a pile of socks, of $n$ colors, how many do you need to pick (randomly) to get $k$ of one type?

\[(k-1)n + 1\]

Worst scenario: pick $k-1$ of each color...
Prove: for any set of 5 points in a unit square, there are 2 points within distance $\leq \frac{\sqrt{2}}{2}$.
Prove: for any set of 5 points in a unit square, there are 2 points within distance $\leq \frac{\sqrt{2}}{2}$.

There are 4 quadrants (holes).
Prove: for any set of 5 points in a unit square, there are 2 points within distance \( \leq \frac{\sqrt{2}}{2} \).

There are 4 quadrants (holes). By pigeonhole, \( \geq 2 \) points are in one quadrant.

points = pigeons
Prove: for any set of 5 points in a unit square, there are 2 points within distance \( \leq \frac{\sqrt{2}}{2} \).

There are 4 quadrants (holes).

By pigeonhole, \( \geq 2 \) points are in one quadrant.

points = pigeons

Max distance in quadrant = \( \sqrt{(0.5)^2 + (0.5)^2} = \frac{\sqrt{2}}{2} \)
Prove: for any set of 5 points in a unit square, there are 2 points within distance \( \leq \frac{\sqrt{2}}{2} \).

There are 4 quadrants (holes). By pigeonhole, \( \geq 2 \) points are in one quadrant.

points = pigeons

Max distance in quadrant = \( \sqrt{(0.5)^2 + (0.5)^2} = \frac{\sqrt{2}}{2} \)

Works for other shapes & dimensions too.
Prove: if \( n \) teams play each other once (aka round robin), and every team wins at least once, [no ties allowed] then 2 teams will have the same number of wins.
Prove: if $n$ teams play each other once (aka round robin), and every team wins at least once, [no ties allowed] then 2 teams will have the same number of wins.

How many wins could a team have?
Prove: if $n$ teams play each other once (aka round robin), and every team wins at least once, [no ties allowed] then 2 teams will have the same number of wins.

How many wins could a team have? $\{1, 2, \ldots, n-1\}$
Prove: if \( n \) teams play each other once (aka round robin), and every team wins at least once, [no ties allowed] then 2 teams will have the same number of wins.

How many wins could a team have? \( \{1, 2, \ldots, n-1\} \)

\( n \) teams = pigeons

\( n-1 \) possible wins = holes
Definition: if A is a friend of B then B is a friend of A.
Definition: if A is a friend of B then B is a friend of A.

Prove: at any party with $n$ people,

two of them have the same number of friends present.
Definition: if $A$ is a friend of $B$ then $B$ is a friend of $A$.

Prove: at any party with $n$ people,

two of them have the same number of friends present.

How many friends could a person have at the party?
Definition: if A is a friend of B then B is a friend of A.

Prove: at any party with n people,

two of them have the same number of friends present.

How many friends could a person have at the party? \( \{0, 1, 2, \ldots, n-1\} \)
Definition: if A is a friend of B then B is a friend of A.

Prove: at any party with n people,
two of them have the same number of friends present.

How many friends could a person have at the party? \{0, 1, 2, ..., n-1\}

But if someone has no friends, then nobody has n-1 friends.
Definition: if $A$ is a friend of $B$ then $B$ is a friend of $A$.

Prove: at any party with $n$ people,

two of them have the same number of friends present.

How many friends could a person have at the party? $\{0, 1, 2, \ldots, n-1\}$

But if someone has no friends, then nobody has $n-1$ friends.

So the set is $\{0, 1, 2, \ldots, n-2\}$ or $\{1, 2, \ldots, n-1\}$.
**Definition**: if $A$ is a friend of $B$ then $B$ is a friend of $A$.

**Prove**: at any party with $n$ people, two of them have the same number of friends present.

How many friends could a person have at the party? $\{0, 1, 2, ..., n-1\}$

But if someone has no friends, then nobody has $n-1$ friends.

So the set is $\{0, 1, 2, ..., n-2\}$ or $\{1, 2, ..., n-1\}$

By pigeonhole, $\begin{bmatrix}$
people = pigeons$
$\text{valid \# friends = holes}$
$\end{bmatrix}$ either case works
$S = \{1, 2, 3, \ldots, 100\}$

Prove: if you are given any 51 numbers from $S$, you can find a pair that sums to 101.
$S = \{1, 2, 3, \ldots, 100\}$

Prove: if you are given any 51 numbers from $S$, you can find a pair that sums to 101.

Make 50 buckets: $\{1, 100\}, \{2, 99\}, \ldots, \{49, 52\}, \{50, 51\}$
$S = \{1, 2, 3, \ldots, 100\}$

Prove: if you are given any 51 numbers from $S$, you can find a pair that sums to 101.

Make 50 buckets: $\{1, 100\}, \{2, 99\}, \ldots, \{49, 52\}, \{50, 51\}$

By pigeonhole, your 51 numbers must include 2 in the same bucket.
Let \( L \) be a list of 32 8-digit decimal numbers.
Let $L$ be a list of 32 8-digit decimal numbers. Prove: there are 2 subsets of $L$ that have the same sum.
Let \( L \) be a list of 32 8-digit decimal numbers. Prove: there are 2 subsets of \( L \) that have the same sum.

Range of possible sums?
Let $L$ be a list of 32 8-digit decimal numbers. 

Prove: there are 2 subsets of $L$ that have the same sum.

Range of possible sums? $\rightarrow 0 \ldots \sum \rightarrow 0 \ldots 32 \cdot 10^8$

\[ \vdots \]

in fact, less than this
Let $L$ be a list of 32 8-digit decimal numbers.

Prove: there are 2 subsets of $L$ that have the same sum.

Range of possible sums? $\rightarrow 0 \ldots \sum_{L} \rightarrow 0 \ldots 32 \cdot 10^8$

How many subsets can we make?
Let \( L \) be a list of 32 8-digit decimal numbers. Prove: there are 2 subsets of \( L \) that have the same sum.

Range of possible sums? \( \rightarrow 0 \ldots \sum_{L} \rightarrow 0 \ldots 32 \cdot 10^8 \)

How many subsets can we make? \( \rightarrow 2^{32} \)

why?
Let $L$ be a list of 32 8-digit decimal numbers.

Prove: there are 2 subsets of $L$ that have the same sum.

Range of possible sums? $\rightarrow 0...\sum_{L} \rightarrow 0...32 \cdot 10^8$

How many subsets can we make? $\rightarrow 2^{32}$

(all combinations of in/out)
Let $L$ be a list of 32 8-digit decimal numbers.

Prove: there are 2 subsets of $L$ that have the same sum.

Range of possible sums? $\rightarrow 0 \ldots \sum_{L} \rightarrow 0 \ldots 32 \cdot 10^8$ 

$< 3,200,000,000$

How many subsets can we make? $\rightarrow 2^{32} = 4,294,967,296$

(all combinations of in/out)
Let $L$ be a list of 32 8-digit decimal numbers.

Prove: there are 2 subsets of $L$ that have the same sum.

Range of possible sums? $\rightarrow 0 \ldots \sum_{L} \rightarrow 0 \ldots 32 \cdot 10^8$

$< 3,200,000,000$ "holes"

How many subsets can we make? $\rightarrow 2^{32} = 4,294,967,296$ "pigeons"

(all combinations of in/out)

By pigeonhole, 2 subsets must have the same sum. \qed
Let \( L \) be a list of 32 8-digit decimal numbers.

Prove: there are 2 subsets of \( L \) that have the same sum.

Range of possible sums? \( 0 \leq \sum_{L} \leq 32 \cdot 10^8 \)

\( < 3,200,000,000 \) "holes"

How many subsets can we make? \( 2^{32} = 4,294,967,296 \) "pigeons"

(all combinations of in/out)

By pigeonhole, 2 subsets must have the same sum. \( \square \)

Note: if 2 subsets have common numbers we can remove them & get a solution with 2 disjoint subsets.
Let $L$ be a list of 32 8-digit decimal numbers.

Proved: there are 2 subsets of $L$ that have the same sum.

For 25-digit numbers, you only need $|L| \geq 90$
(see MCS, ch.15)
Let $L$ be a list of 32 8-digit decimal numbers.

Proved: there are 2 subsets of $L$ that have the same sum.

For 25-digit numbers, you only need $|L| \geq 90$

(see MCS, ch.15)

It is difficult to actually find a solution efficiently but it was easy to show that a solution exists.

This type of proof is called “non-constructive”
Let \( L \) be a list of 32 8-digit decimal numbers.

Proved: there are 2 subsets of \( L \) that have the same sum.

For 25-digit numbers, you only need \( |L| \geq 90 \)

(see MCS, ch.15)

It is difficult to actually find a solution efficiently
but it was easy to show that a solution exists.

This type of proof is called “non-constructive”

This problem is related to applications from shipping/packaging to crypto.
Prove: for every $n$ there are integers $1 \leq a, b \leq 11$ s.t. $a \neq b$, and $10 \mid n^a - n^b$.

10 divides $n^a - n^b$.

$$\frac{n^a - n^b}{10} = \text{integer}.$$
Prove: for every \( n \) there are integers \( 1 \leq a, b \leq 11 \) s.t. \( a \neq b \), and \( 10 \mid n^a - n^b \)

Example 1: \( n = 3 \). Pick \( a = 5, b = 1 \). \( 3^5 - 3^1 = 240 \)
Prove: for every $n$ there are integers $1 \leq a, b \leq 11$ s.t. $a \neq b$, and $10 | n^a - n^b$

example 1: $n=3$. Pick $a=5$, $b=1$. $3^5 - 3^1 = 240$

example 2: $n=4$. Pick $a=5$, $b=3$. $4^5 - 4^3 = 960$
Prove: for every $n$ there are integers $1 \leq a, b \leq 11$ s.t. $a \neq b$, and $10 \mid n^a - n^b$

Example 1: $n=3$. Pick $a=5$, $b=1$. $3^5 - 3^1 = 240$

Example 2: $n=4$. Pick $a=5$, $b=3$. $4^5 - 4^3 = 960$

Example 3: $n=17$. Pick $a=6$, $b=2$. $17^6 - 17^2 = 24,137,280$
Prove: for every $n$ there are integers $1 \leq a, b \leq 11$ s.t. $a \neq b$, and $10|n^a - n^b$

Example 1: $n=3$. Pick $a=5$, $b=1$. $3^5 - 3^1 = 240$

Example 2: $n=4$. Pick $a=5$, $b=3$. $4^5 - 4^3 = 960$

Example 3: $n=17$. Pick $a=6$, $b=2$. $17^6 - 17^2 = 24,137,280$

We want $n^a - n^b$ to end with a 0.
Prove: for every $n$ there are integers $1 \leq a, b \leq 11$ s.t. $a \neq b$, and $10 \mid n^a - n^b$

example 1: $n=3$. Pick $a=5$, $b=1$. $3^5 - 3^1 = 240$
example 2: $n=4$. Pick $a=5$, $b=3$. $4^5 - 4^3 = 960$
example 3: $n=17$. Pick $a=6$, $b=2$. $17^6 - 17^2 = 24,137,280$

We want $n^a - n^b$ to end with a 0. $\rightarrow$ $n^a$ & $n^b$ end with same digit.
Prove: for every $n$ there are integers $1 \leq a, b \leq 11$ s.t. $a \neq b$, and $10 \mid n^a - n^b$

example 1: $n=3$. Pick $a=5, b=1$. $3^5 - 3^1 = 240$

example 2: $n=4$. Pick $a=5, b=3$. $4^5 - 4^3 = 960$

example 3: $n=17$. Pick $a=6, b=2$. $17^6 - 17^2 = 24,137,280$

We want $n^a - n^b$ to end with a 0. $\rightarrow n^a$ & $n^b$ end with same digit.

Compute $n^0, n^1, n^2, \ldots, n^{10}$

e.g. $n=3$: 1, 3, 9, 27, 81, 243, 729, 2187, 6561, 19683, 59049
Prove: for every $n$ there are integers $1 \leq a, b \leq 11$ s.t. $a \neq b$, and $10 \mid n^a - n^b$.

**Example 1:** $n=3$. Pick $a=5$, $b=1$. $3^5 - 3^1 = 240$

**Example 2:** $n=4$. Pick $a=5$, $b=3$. $4^5 - 4^3 = 960$

**Example 3:** $n=17$. Pick $a=6$, $b=2$. $17^6 - 17^2 = 24,137,280$

We want $n^a - n^b$ to end with a 0. $\rightarrow n^a$ & $n^b$ end with same digit.

Compute $n^0$, $n^1$, $n^2$, ..., $n^{10}$ and place each in a bucket by last digit.

E.g. $n=3$: $1, 3, 9, 27, 81, 243, 729, 2187, 6561, 19683, 59049$
Prove: for every $n$ there are integers $1 \leq a, b \leq 11$ s.t. $a \neq b$, and $10 | n^a - n^b$

example 1: $n=3$. Pick $a=5$, $b=1$. $3^5 - 3^1 = 240$

example 2: $n=4$. Pick $a=5$, $b=3$. $4^5 - 4^3 = 960$

example 3: $n=17$. Pick $a=6$, $b=2$. $17^6 - 17^2 = 24,137,280$

We want $n^a - n^b$ to end with a 0. $\rightarrow n^a$ & $n^b$ end with same digit.

Compute $n^0$, $n^1$, $n^2$, ..., $n^{10}$ and place each in a bucket by last digit.

e.g. $n=3$: 1, 3, 9, 27, 81, 243, 729, 2187, 6561, 19683, 59049

Proved by pigeonhole. 11 powers (pigeons), 10 buckets (holes)
FILE ZIPPING

Objective: reduce storage (#bits) for any given file
FILE ZIPPING

Objective: reduce storage (#bits) for any given file

files A
    B
↑       zip via any single zipping algorithm

A.zip
↑
B.zip

ideally small
FILE ZIPPING

Objective: reduce storage (#bits) for any given file

files A B → zip via any single zipping algorithm → A.zip → unzip when needed → A B
FILE ZIPPING

Objective: reduce storage (#bits) for any given file

files $A$ $\rightarrow$ zip via any single zipping algorithm $\rightarrow$ A.zip $\rightarrow$ unzip $\rightarrow$ A

B $\rightarrow$ B.zip $\rightarrow$ when needed $\rightarrow$ B

if A.zip = B.zip, can't unzip reliably
FILE ZIPPING

Objective: reduce storage (#bits) for any given file

files A → zip via any single zipping algorithm → A.zip → unzip → A

B → zipping algorithm → B.zip → when needed → B

if A.zip = B.zip, can't unzip reliably

Claim: Whatever zip algorithm you choose (with reliable unzip), there is some n-bit file X for which X.zip uses \( \geq n \) bits
FILE ZIPPING

Objective: reduce storage (#bits) for any given file

files A  B → zip via any single zipping algorithm → A.zip → unzip → A

B.zip → when needed → B

if A.zip = B.zip, can't unzip reliably

Claim: Whatever zip algorithm you choose (with reliable unzip), there is some n-bit file X for which X.zip uses \( \geq n \) bits.

Contrapositive: if all .zip files use < n bits, unzip won't work reliably.
FILE ZIPPING

Objective: reduce storage (#bits) for any given file

files A \rightarrow \text{zip via any single zipping algorithm} \rightarrow \text{unzip} \rightarrow \text{A}

B \rightarrow \text{B} \rightarrow \text{when needed} \rightarrow \text{unzip reliably}

Claim: Whatever zip algorithm you choose (with reliable unzip), there is some n-bit file X for which X.zip uses \geq n bits

Contrapositive: if all .zip files use < n bits, unzip won’t work reliably.

Look at all n-bit files. How many?
**FILE ZIPPING**

Objective: reduce storage (#bits) for any given file

A → zip via any single zipping algorithm → A.zip → unzip
B → when needed → B.zip

if A.zip = B.zip, can't unzip reliably

Claim: Whatever zip algorithm you choose (with reliable unzip), there is some n-bit file X for which X.zip uses \( \geq n \) bits.

Contrapositive: if all .zip files use \(<n\) bits, unzip won't work reliably.

Look at all n-bit files. How many? \( 2^n \) \( \{0,1\}^n \)
FILE ZIPPING

Objective: reduce storage (#bits) for any given file

files A B → zip via any single zipping algorithm → A.zip → unzip → B.zip → when needed → A B

if A.zip = B.zip, can’t unzip reliably

Claim: Whatever zip algorithm you choose (with reliable unzip), there is some n-bit file X for which X.zip uses \( \geq n \) bits

Contrapositive: if all .zip files use \(< n \) bits, unzip won’t work reliably.

Look at all n-bit files. How many? \( 2^n \)

How many zip files with \(< n \) bits?
FILE ZIPPING  

Objective: reduce storage (#bits) for any given file

files \( A \) → zip via any single zipping algorithm → \( A.\text{zip} \) → unzip → \( A \)

\( B.\text{zip} \) → when needed → \( B \)

if \( A.\text{zip} = B.\text{zip} \), can't unzip reliably

Claim: Whatever zip algorithm you choose (with reliable unzip), there is some \( n \)-bit file \( X \) for which \( X.\text{zip} \) uses \( \geq n \) bits

Contrapositive: if all .zip files use \( < n \) bits, unzip won't work reliably.

Look at all \( n \)-bit files. How many? \( 2^n \)

How many zip files with \( < n \) bits? \( \sum_{i=0}^{n-1} 2^i = ? \)
FILE ZIPPING

Objective: reduce storage (#bits) for any given file

files A → zip via any single zipping algorithm → A.zip → unzip → A
B → B.zip → when needed → B

if A.zip = B.zip, can't unzip reliably

Claim: Whatever zip algorithm you choose (with reliable unzip), there is some n-bit file X for which X.zip uses \( \geq n \) bits.

Contrapositive: if all .zip files use \(< n\) bits, unzip won't work reliably.

Look at all n-bit files. How many? \( 2^n \)

How many zip files with \(< n\) bits? \( \sum_{i=0}^{n-1} 2^i = 2^n - 1 \)
FILE ZIPPING  

Objective: reduce storage (#bits) for any given file

files A → zip via any single zipping algorithm → A.zip → unzip → A

B → B.zip

when needed → B

if A.zip = B.zip, can’t unzip reliably

Claim: Whatever zip algorithm you choose (with reliable unzip), there is some n-bit file X for which X.zip uses \( \geq n \) bits.

Contrapositive: if all .zip files use <n bits, unzip won’t work reliably.

Look at all n-bit files. How many? \( 2^n \)

How many zip files with <n bits? \( \sum_{i=0}^{n-1} 2^i = 2^n - 1 \)

If every n-bit file zips to a <n-bit file, ...?
FILE ZIPPING

Objective: reduce storage (#bits) for any given file

files A → zip via any single zipping algorithm → A.zip → unzip when needed → A

B.zip → when needed → B

if A.zip = B.zip, can’t unzip reliably

Claim: Whatever zip algorithm you choose (with reliable unzip), there is some n-bit file X for which X.zip uses \( \geq n \) bits

Contrapositive: if all .zip files use \(< n \) bits, unzip won’t work reliably.

Look at all n-bit files. How many? \( 2^n \) (pigeons)

How many zip files with \(< n \) bits? \( \sum_{i=0}^{n-1} 2^i = 2^n - 1 \) (holes)

If every n-bit file zips to a \(< n \)-bit file, \( \exists A, B \) s.t. A.zip = B.zip
**FILE ZIPPING**  
Objective: reduce storage (#bits) for any given file

Files A → zip via any single zipping algorithm → A.zip → unzip → A  
B → B.zip → when needed → B

If A.zip = B.zip, can’t unzip reliably.

Claim: Whatever zip algorithm you choose (with reliable unzip), there is some n-bit file X for which X.zip uses \( \geq n \) bits.

Contrapositive: if all .zip files use \(< n \) bits, unzip won’t work reliably.

Look at all n-bit files. How many?  
\( 2^n \) (pigeons)

How many zip files with \(< n \) bits?  
\[ \sum_{i=0}^{n-1} 2^i = 2^n - 1 \] (holes)

If every n-bit file zips to a \(< n \)-bit file, \( \exists A, B \) s.t. A.zip = B.zip
Consider 5 points on a grid. (size & dimensions don't matter)

diagram:

example:
Consider 5 points on a grid. (size & dimensions don't matter)

Prove: ∃ 2 points such that their midpoint is on the grid.
Consider 5 points on a grid. (size & dimensions don't matter)

Prove: ∃ 2 points such that their midpoint is on the grid.

Example:

\[ \times \text{ segment doesn't pass through grid point} \]
Consider 5 points on a grid. (size & dimensions don’t matter)

Prove: ∃ 2 points such that their midpoint is on the grid.

Example:

X segment doesn’t pass through grid point
Consider 5 points on a grid. (size & dimensions don't matter)

Prove: \exists 2 points such that their midpoint is on the grid.

Example:

\[ \times \text{ segments don't pass through grid point} \]
Consider 5 points on a grid. (size & dimensions don't matter)

Prove: $\exists$ 2 points such that their midpoint is on the grid.

Example:

× segment does pass through grid points but midpoint isn't on grid.
Consider 5 points on a grid. (size & dimensions don't matter)

Prove: ∃ 2 points such that their midpoint is on the grid.

Example:

✔ success

(try to shift points & avoid the midpoint claim)
Prove: among any 5 grid points, \( \geq 2 \) have midpoint on grid.
Prove: among any 5 grid points, ≥2 have midpoint on grid.

Odd v Even
Prove: among any 5 grid points, at least 2 have midpoint on grid.

4 grid position types for (x,y):
odd v even: (0,E) (E,0) (0,0) (E,E)
Prove: among any 5 grid points, ≥ 2 have midpoint on grid.

4 grid position types for \((x, y)\):

Odd v Even: \((0, 0)\) \((0, E)\) \((E, 0)\) \((E, E)\)

Pigeonhole: 2 points \(A, B\) \(\rightarrow\) same type.
Prove: among any 5 grid points, 2 have midpoint on grid.

4 grid position types for \((x,y)\):
- Odd vs Even: \((0,E)\) \((E,0)\) \((0,0)\) \((E,E)\)

Pigeonhole: 2 points \(A,B\) of same type.

For each coordinate \(c = \{x,y\}\):
\[
\text{midpoint}(A,B)_c = \frac{1}{2}(A_c + B_c)
\]
Prove: among any 5 grid points, ≥2 have midpoint on grid.

4 grid position types for \((x, y)\):
Odd v Even: \((0, E)\) \((E, 0)\) \((0, 0)\) \((E, E)\)

Pigeonhole: 2 points \(A, B\) → same type.
For each coordinate \(c = \{x, y\}\)
midpoint\((A, B)_c\) = \(\frac{1}{2}(A_c + B_c)\)
If same type, \((A_c + B_c)\) → even
(e.g., odd + odd = even)
Prove: among any 5 grid points, ≥ 2 have midpoint on grid.

4 grid position types for \((x,y)\):
Odd v Even: \((0,E)\) \((E,0)\) \((0,0)\) \((E,E)\)

Pigeonhole: 2 points \(A, B\) → same type.

For each coordinate \(c = \{x, y\}\)
\[
\text{midpoint}(A, B)_c = \frac{1}{2}(A_c + B_c)
\]

If same type, \((A_c + B_c)\) → even → \(2 \cdot k\) (e.g., odd + odd = even) \((k = \text{integer})\)
Prove: among any 5 grid points, \( \geq 2 \) have midpoint on grid.

4 grid position types for \((x,y)\):
- Odd v Even: \((0,E)(E,0)(0,0)(E,E)\)

Pigeonhole: 2 points \(A,B\) \(\rightarrow\) same type.

For each coordinate \(c=\{x,y\}\):

\[
\text{midpoint}(A,B)_c = \frac{1}{2}(A_c + B_c)
\]

If same type, \((A_c + B_c) \rightarrow \text{even} \rightarrow 2 \cdot k\)

(e.g., odd + odd = even) \(k = \text{integer}\)

So midpoint = \(\frac{1}{2}2k = \text{integer}\) \(\Box\)
sequence of distinct numbers

17  3  6  92  8  22  31  27  18  13  45  68  33  72  49
Given a sequence of distinct numbers, a subsequence of size \( k \) is \( k \)-monotone if it is either increasing or decreasing.

17 3 6 92 8 22 31 27 18 13 45 68 33 72 49
Given a sequence of distinct numbers, a subsequence of size $k$ is $k$-monotone if it is either increasing or decreasing.

$17\ 3\ 6\ 92\ 8\ 22\ 31\ 27\ 18\ 13\ 45\ 68\ 33\ 72\ 49$

2-monotone
Given a sequence of distinct numbers, a subsequence of size $k$ is $k$-monotone if it is either increasing or decreasing.

17 3 6 92 8 22 31 27 18 13 45 68 33 72 49

5-monotone
Given a sequence of distinct numbers, a subsequence of size $k$ is $k$-monotone if it is either increasing or decreasing.
Given a sequence of distinct numbers, a subsequence of size $k$ is $k$-monotone if it is either increasing or decreasing.

17 3 6 92 8 22 31 27 18 13 45 68 33 72 49

Geometric view: (6,22) (8,27)
Create 2D point set
Given a sequence of distinct numbers, a subsequence of size \( k \) is \( k \)-monotone if it is either increasing or decreasing.

Geometric view:

Create 2D point set
How large must a sequence be, to guarantee a "large" monotone subsequence?
How large must a sequence be, to guarantee a "large" monotone subsequence?

Claim: A \((n+1)\)-monotone subsequence exists in every sequence of size \(n^2+1\).
How large must a sequence be, to guarantee a "large" monotone subsequence?

Claim: A \((n+1)\)-monotone subsequence exists in every sequence of size \(n^2+1\).

Notice, we can't do better:

\[ n = 4 \quad \Rightarrow \quad n^2 = 16 \text{ pts} \]

can't find \((n+1)\)-monotone

= 5-monotone subsequence
How large must a sequence be, to guarantee a "large" monotone subsequence?

Claim: A \((n+1)\)-monotone subsequence exists in every sequence of size \(n^2+1\).

Notice, we can't do better:

\[
\begin{align*}
n &= 4 & n^2 &= 16 \text{ pts:} \\
\text{can't find } (n+1)\text{-monotone} \\
\text{= 5-monotone subsequence}
\end{align*}
\]
Prove: A \((n+1)\)-monotone subsequence exists in every sequence of size \(n^2+1\).
Prove: A \((n+1)\)-monotone subsequence exists in every sequence of size \(n^2+1\).

Define \(u_i, d_i\): sizes of longest upward & downward chains ending at position \(i\).
Prove: A \((n+1)\)-monotone subsequence exists in every sequence of size \(n^2 + 1\).

Define \(u_i, d_i\): sizes of longest upward & downward chains ending at position \(i\).

Assuming no \((n+1)\)-monotone chain, for all \(i\): \(u_i, d_i \leq n\)
Prove: A \((n+1)\)-monotone subsequence exists in every sequence of size \(n^2+1\).

Define \(u_i, d_i\): sizes of longest upward & downward chains ending at position \(i\).

Assuming no \((n+1)\)-monotone chain, for all \(i\): \(u_i, d_i \leq n \Rightarrow \leq n^2\) holes.

By pigeonhole, 2 of the \(n^2+1\) points have equal \((u, d)\).
Prove: A \((n+1)\)-monotone subsequence exists in every sequence of size \(n^2+1\).

Define \(u_i, d_i\): sizes of longest upward & downward chains ending at position \(i\).

Assuming no \((n+1)\)-monotone chain, for all \(i\): \(u_i, d_i \leq n \Rightarrow \leq n^2\) holes

By pigeonhole, 2 of the \(n^2+1\) points have equal \((u, d)\).

But... position \(j = i+k\) will extend \(u_i\) or \(d_i\) \(\rightarrow\) either \(u_j > u_i\) or \(d_j > d_i\) \((k > 0)\)
Prove: A \((n+1)\)-monotone subsequence exists in every sequence of size \(n^2+1\).

Define \(u_i, d_i\): sizes of longest upward & downward chains ending at position \(i\).

Assuming no \((n+1)\)-monotone chain, for all \(i\): \(u_i, d_i \leq n \Rightarrow \leq n^2\) holes

By pigeonhole, 2 of the \(n^2+1\) points have equal \((u, d)\).

But...

position \(j = i+k\) will extend \(u_i\) or \(d_i\) → either \(u_j > u_i\) or \(d_j > d_i\)

\(\Rightarrow\) So for all \(i, j\): \((u_i, d_i) \neq (u_j, d_j)\)
Prove: A \((n+1)\)-monotone subsequence exists in every sequence of size \(n^2+1\).

Define \(u_i, d_i\): sizes of longest upward & downward chains ending at position \(i\).

Assuming no \((n+1)\)-monotone chain, for all \(i\): \(u_i, d_i \leq n \Rightarrow \leq n^2\) holes.

By pigeonhole, 2 of the \(n^2+1\) points have equal \((u, d)\).

\underbrace{\text{But...}} \\
\text{position } j = i + k \text{ will extend } u_i \text{ or } d_i \rightarrow \text{ either } u_j > u_i \text{ or } d_j > d_i

\downarrow \text{ So for all } i, j: (u_i, d_i) \neq (u_j, d_j) \text{ Contradiction.}