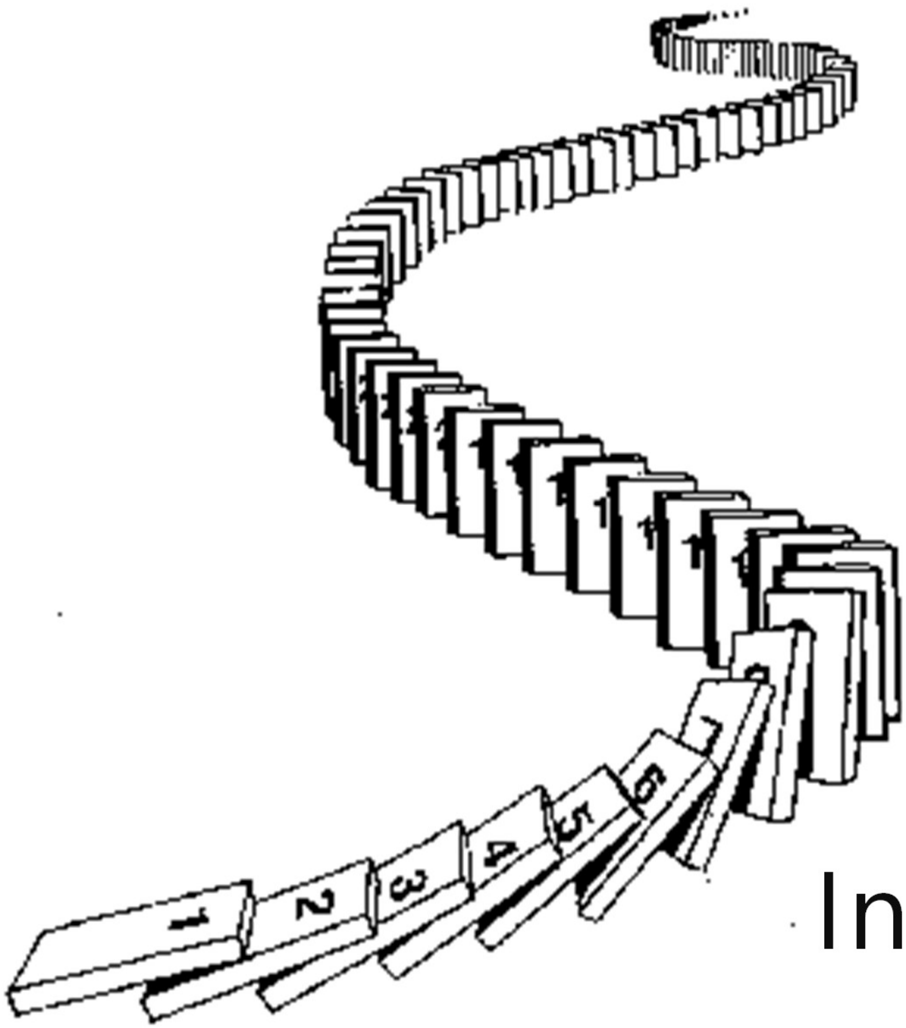


Induction

CSE 311 Summer 2024
Lecture 12

Midterm Announcements

- Homework 4 is Due this Friday
- Midterm will be in class on Monday Jully 22nd
- An Ed post is going to go out with more of a list of topics, but you can expect at least these questions:
  - A translation question
  - A set theory proof question
  - A number theory proof question
  - An induction proof question
- You can not expect these things:
  - Strong/Structural induction
  - Running through the extended Euclidian algorithm
  - Anything related to RSA or fast exponentiation
  - Write out any propositional logic chains of equivalences more than 1 or 2 steps

How do we know recursion works?

```
//Assume i is a nonnegative integer
//returns 2^i.
public int CalculatesTwoToTheI(int i){
     if(i == 0)
          return 1;
     else
          return 2*CaclulatesTwoToTheI(i-1);
}
```

Why does `CalculatesTwoToTheI(4)` calculate `2^4?`
Convince the people around you!

How do we know recursion works?

Something like this:

Well, as long as `CalculatesTwoToTheI(3) = 8`, we get 16…

Which happens as long as `CalculatesTwoToTheI(2) = 4`

Which happens as long as `CalculatesTwoToTheI(1) = 2`

Which happens as long as `CalculatesTwoToTheI(0) = 1`

And it is! Because that's what the base case says.

How do we know recursion works?

There's really only two cases.

The Base Case is Correct

`CalculatesTwoToTheI(0) = 1 (which it should!)`

And that means `CalculatesTwoToTheI(1) = 2`, (like it should)

And that means `CalculatesTwoToTheI(2) = 4`, (like it should)

And that means `CalculatesTwoToTheI(3) = 8,` (like it should)

And that means `CalculatesTwoToTheI(4) = 16,` (like it should)

IF the recursive call we make is correct
THEN our value is correct.

How do we know recursion works?

The code has two big cases,

So our proof had two big cases

"The base case of the code produces the correct output"

"IF the calls we rely on produce the correct output THEN the current call produces the right output"

A bit more formally…

"The base case of the code produces the correct output"

"IF the calls we rely on produce the correct output THEN the current call produces the right output"

Let $P(i)$ be "`CalculatesTwoToTheI(i)` returns 2^i."

How do we know $P(4)$?

$P(0)$ is true.

And $P(0) \to P(1)$, so $P(1)$.

And $P(1) \to P(2)$, so $P(2)$.

And $P(2) \to P(3)$, so $P(3)$.

And $P(3) \to P(4)$, so $P(4)$.

A bit more formally…

This works alright for $P(4)$.

What about $P(1000)$? $P(1000000000)$?

At this point, we'd need to show that implication $P(k) \to P(k+1)$ for A BUNCH of values of k.

But the code is the same each time.

And so was the argument!

We should instead show $\forall k[P(k) \to P(k+1)]$.

Induction

Your new favorite proof technique!

How do we show $\forall n, P(n)$?

Show $P(0)$

Show $\forall k(P(k) \rightarrow P(k+1))$

Induction

```
//Assume i is a nonnegative integer
public int CalculatesTwoToTheI(int i){
       if(i == 0)
              return 1;
       else
              return 2*CaclulatesTwoToTheI(i-1);
}
```

Let $P(i)$ be "`CalculatesTwoToTheI(i)` returns 2^i."

Note that if the input i is 0, then the if-statement evaluates to true, and $1 = 2^0$ is returned, so $P(0)$ is true.

Suppose $P(k)$ holds for an arbitrary $k \geq 0$.

Consider the code run on $k + 1$. Since $k \geq 0$, $k + 1 \geq 1$ and we are in the else branch. By inductive hypothesis, `CalculatesTwoToTheI(`k`)` returns 2^k, so the code run on $k + 1$ returns $2 \cdot 2^k = 2^{k+1}$.

So $P(k + 1)$ holds.

Therefore $P(n)$ holds for all $n \geq 0$ by the principle of induction.

Making Induction Proofs Pretty

Let $P(i)$ be the predicate "`CalculatesTwoToTheI(i)` returns 2^i." We prove $P(n)$ holds holds for all natural numbers n by induction on n.

Base Case ($i = 0$) Note that if the input i is 0, then the if-statement evaluates to true, and $1 = 2^0$ is returned, so $P(0)$ is true.

Inductive Hypothesis: Suppose $P(k)$ holds for an arbitrary $k \geq 0$.

Inductive Step: Since $k \geq 0, k + 1 \geq 1$, so the code goes to the recursive case. We will return $2 \cdot$ `CalculatesTwoToTheI(k)`. By Inductive Hypothesis,

`CalculatesTwoToTheI(k)` = 2^k. Thus we return $2 \cdot 2^k = 2^{k+1}$.

So $P(k + 1)$ holds.

Therefore $P(n)$ holds for all $n \geq 0$ by the principle of induction.

Making Induction Proofs Pretty

All of our induction proofs will come in 5 easy(?) steps!

1. Define $P(n)$. State that your proof is by induction on n.
2. Show $P(0)$ i.e. show the base case
3. Suppose $P(k)$ for an arbitrary k.
4. Show $P(k+1)$ (i.e. get $P(k) \to P(k+1)$)
5. Conclude by saying $P(n)$ is true for all n by induction.

Some Other Notes

Always state where you use the inductive hypothesis when you're using it in the inductive step.

It's usually the key step, and the reader really needs to focus on it.

Be careful about what values you're assuming the Inductive Hypothesis for – the smallest possible value of k should assume the base case but nothing more.

The Principle of Induction (formally)

$$\text{Principle of Induction}\quad \frac{P(0);\forall k(P(k)\rightarrow P(k+1))}{\therefore\quad \forall n(P(n))}$$

Informally: if you knock over one domino, and every domino knocks over the next one, then all your dominoes fell over.

More induction!

More Induction

Induction doesn't only work for code!

Show that $\sum_{i=0}^{n} 2^i = 1 + 2 + 4 + \cdots + 2^n = 2^{n+1} - 1$.

More Induction

Induction doesn't only work for code!

Show that $\sum_{i=0}^{n} 2^i = 1 + 2 + 4 + \cdots + 2^n = 2^{n+1} - 1$.

Let $P(n) =$"$\sum_{i=0}^{n} 2^i = 2^{n+1} - 1$."

We show $P(n)$ holds for all natural numbers n by induction on n.

Base Case ( )

Inductive Hypothesis:

Inductive Step:

$P(n)$ holds for all $n \geq 0$ by the principle of induction.

More Induction

Induction doesn't only work for code!

Show that $\sum_{i=0}^{n} 2^i = 1 + 2 + 4 + \cdots + 2^n = 2^{n+1} - 1$.

Let $P(n) =$"$\sum_{i=0}^{n} 2^i = 2^{n+1} - 1$."

We show $P(n)$ holds for all natural numbers n by induction on n.

Base Case $(n = 0)$ $\sum_{i=0}^{0} 2^i = 1 = 2 - 1 = 2^{0+1} - 1$.

Inductive Hypothesis: Suppose $P(k)$ holds for an arbitrary $k \geq 0$.

Inductive Step: We show $P(k+1)$. Consider the summation $\sum_{i=0}^{k+1} 2^i = 2^{\mathrm{k}+1} + \sum_{i=0}^{k} 2^i = 2^{k+1} + 2^{k+1} - 1$, where the last step is by IH.

Simplifying, we get: $\sum_{i=0}^{k+1} 2^i = 2^{k+1} + 2^{k+1} - 1 = 2 \cdot 2^{k+1} - 1 = 2^{(k+1)+1} - 1$.

$P(n)$ holds for all $n \geq 0$ by the principle of induction.

Induction Template

Induction Template

1. Define $\mathrm{P}(n)$. State that your proof is by induction on n.
2. Base Case: Show $\mathrm{P}(b)$ is true for your base case b.
3. Inductive Hypothesis: Suppose $\mathrm{P}(k)$ holds for an arbitrary integer $k \geq b$.
4. Inductive Step: Prove $\mathrm{P}(k+1)$ (using the Inductive Hypothesis).
5. Conclusion: Conclude by saying $\mathrm{P}(n)$ holds for all integers $n \geq b$ by induction.

Induction Examples

Prove that the sum of the first n positive integers is $\frac{n(n+1)}{2}$.

Examples

$n = 3$ Sum: $1 + 2 + 3 = 6$

Formula: $\frac{3(3+1)}{2} = \frac{3 \cdot 4}{2} = 6$

$n = 5$ Sum: $1 + 2 + 3 + 4 + 5 = 15$

Formula: $\frac{5(5+1)}{2} = \frac{5 \cdot 6}{2} = 15$

Carl Friedrich Gauss
(1777-1855)

Prove that the sum of the first n positive integers is $\frac{n(n+1)}{2}$.

1. Let $\mathrm{P}(n)$ be "$1 + 2 + 3 + \cdots + n = \frac{n(n+1)}{2}$". We prove $\mathrm{P}(n)$ for all ________________ by induction.
2. Base Case:

3. Inductive Hypothesis: Suppose $\mathrm{P}(k)$ holds for

4. Inductive Step: We aim to show $\mathrm{P}(k+1)$. Observe that:
   $1 + 2 + 3 + \cdots + (k+1) = 1 + 2 + 3 + \cdots + k + (k+1)$

   So $\mathrm{P}(k+1)$ holds.
5. Conclusion: Thus $\mathrm{P}(n)$ holds for all integers $n \geq 1$ by induction.

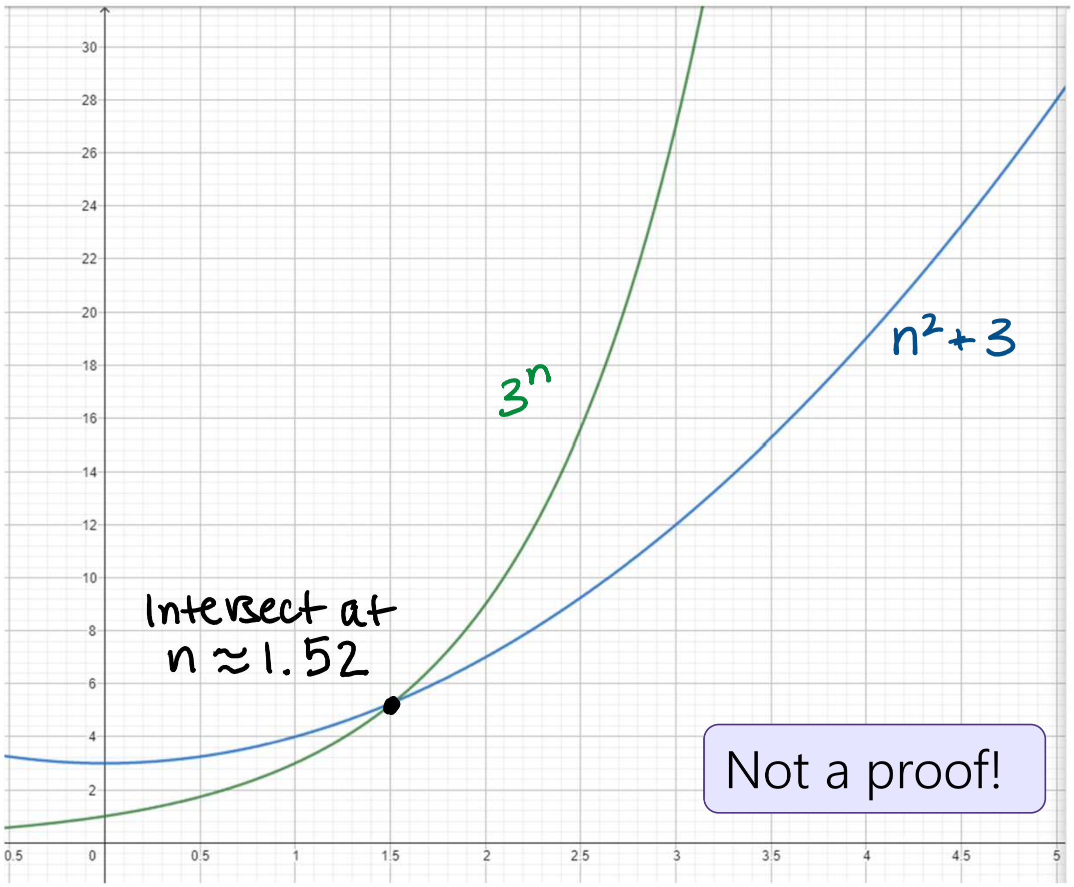
Prove that the sum of the first n positive integers is $\frac{n(n+1)}{2}$.

1. Let $\mathrm{P}(n)$ be "$1 + 2 + 3 + \cdots + n = \frac{n(n+1)}{2}$". We prove $\mathrm{P}(n)$ for all integers $\boldsymbol{n} \geq \mathbf{1}$ by induction.
2. Base Case: The LHS evaluates to 1. The RHS evaluates to $\frac{1(1+1)}{2} = 1$. Since $\mathrm{L}HS = RHS$, the base case holds.
3. Inductive Hypothesis: Suppose $\mathrm{P}(k)$ holds for an arbitrary integer $k \geq 1$. That is, $1 + 2 + 3 + \cdots + k = \frac{k(k+1)}{2}$.
4. Inductive Step: We aim to show $\mathrm{P}(k+1)$. Observe that:

$$\begin{aligned} 1 + 2 + 3 + \cdots + (k+1) &= 1 + 2 + 3 + \cdots + k + (k+1) \\ &= \frac{k(k+1)}{2} + (k+1) && \text{By the IH} \\ &= \frac{k(k+1)}{2} + \frac{2(k+1)}{2} \\ &= \frac{(k+1)(k+2)}{2} \\ &= \frac{(k+1)\big((k+1)+1\big)}{2} \end{aligned}$$

   So $\mathrm{P}(k+1)$ holds.
5. Conclusion: Thus $\mathrm{P}(n)$ holds for all integers $n \geq 1$ by induction.

Prove that $3^n \geq n^2 + 3$ for all integers $n \geq 2$.



Not a proof!

Prove that $3^n \geq n^2 + 3$ for all integers $n \geq 2$.

1. Let $\mathrm{P}(n)$ be "$3^n \geq n^2 + 3$". We prove $\mathrm{P}(n)$ for all ________________ by induction.
2. Base Case ($n = 2$):

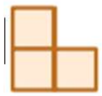
3. Inductive Hypothesis: Suppose $\mathrm{P}(k)$

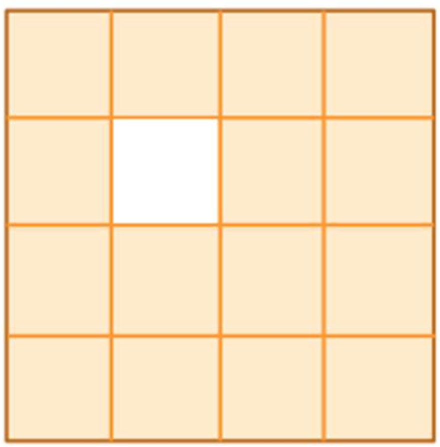
4. Inductive Step: We aim to show $\mathrm{P}(k + 1)$. Observe that:

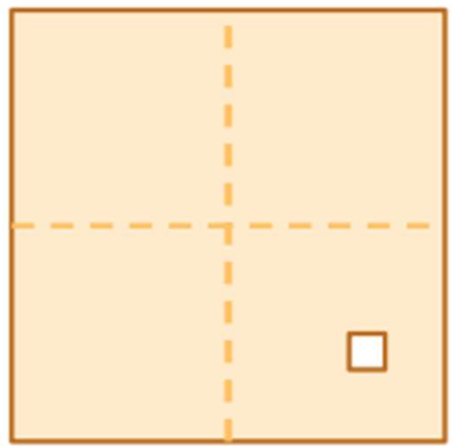
$$3^{k+1} =$$

   So $\mathrm{P}(k + 1)$ holds.
5. Conclusion: Thus $\mathrm{P}(n)$ holds for all integers $n \geq 2$ by induction.

Checkerboard Tiling

- Imagine a $2^n \times 2^n$ checkerboard with a single square removed.
- Can you tile the board with  pieces? You may rotate and flip the pieces around.





- Claim: All $2^n \times 2^n$ boards with one square removed can be tiled with  pieces

Checkerboard Tiling: Base Case

- Consider all 2×2 boards with one piece missing.

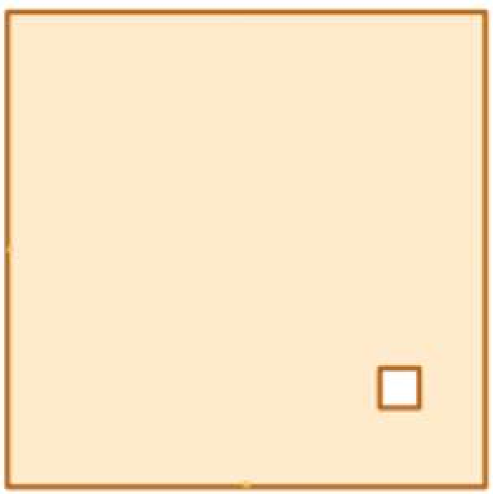
- We can definitely tile these with one piece!

Checkerboard Tiling: Inductive Hypothesis

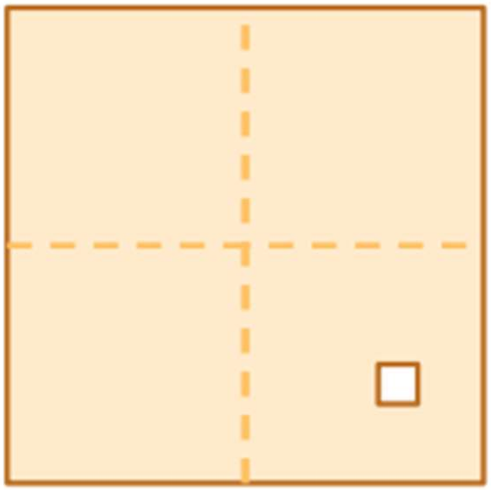
- Assume you could tile any $2^k \times 2^k$ board with one piece missing.
-

Checkerboard Tiling: Inductive Step

- Now consider a $2^{k+1} \times 2^{k+1}$ board with one piece missing.

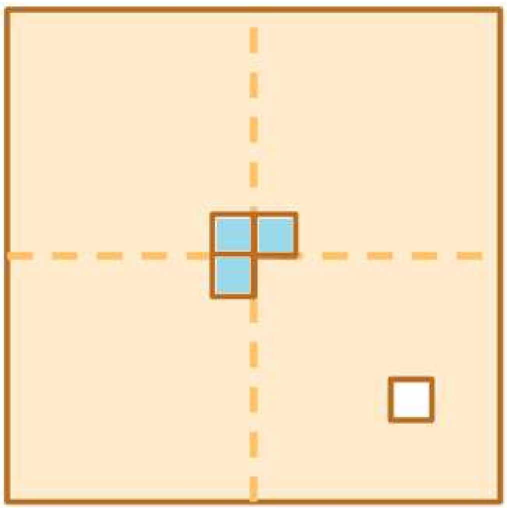


Checkerboard Tiling: Inductive Step

- Now consider a $2^{k+1} \times 2^{k+1}$ board with one piece missing.
- 
- Divide the board into four quadrants of dimension $2^k \times 2^k$.

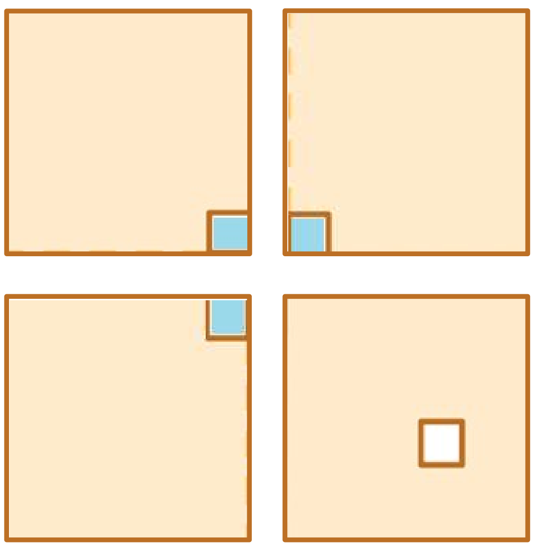
Checkerboard Tiling: Inductive Step

- Now consider a $2^{k+1} \times 2^{k+1}$ board with one piece missing.



- Place a single piece to occupy the three quadrants that aren't missing a piece.

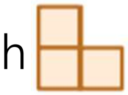
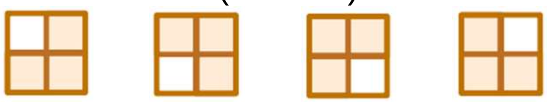
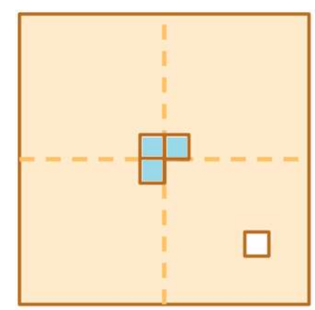
Checkerboard Tiling: Inductive Step

- Now consider a $2^{k+1} \times 2^{k+1}$ board with one piece missing.
- 

- Each quadrant is now a $2^k \times 2^k$ board with one piece missing.

We can tile each of these by the IH.

Checkerboard Tiling

1. Let $\mathrm{P}(n)$ be "all $2^n \times 2^n$ boards with one square removed can be tiled with  pieces."
   We prove $\mathrm{P}(n)$ for all integers $n \geq 1$ by induction.
2. Base Case ($n = 1$): Observe that we can tile all 2×2 checkerboards:
   
   So the base case holds.
3. Inductive Hypothesis: Suppose $\mathrm{P}(k)$ holds for an arbitrary integer $k \geq 1$. That is, assume we can tile all $2^k \times 2^k$ checkerboards with one piece missing.
4. Inductive Step: We aim to show $\mathrm{P}(k+1)$. Consider an arbitrary $2^{k+1} \times 2^{k+1}$ checkerboard. We can divide the board into four quadrants, with one piece missing in one quadrant. Now place a single piece to occupy the three quadrants that aren't missing a piece. We now have four $2^k \times 2^k$ quadrants that are effectively each missing a piece. By the IH, we can tile each quadrant. Thus we can tile the entire checkerboard. So $\mathrm{P}(k+1)$ holds.
   

- 5. Conclusion: Thus $\mathrm{P}(n)$ holds for all integers $n \geq 1$ by induction.