Pearploy Chaicharoensin 6381278

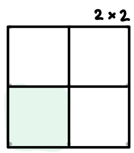
Task1: MissingTile

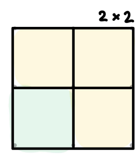
Theorem: Any 2^n -by- 2^n grid with one painted cell can be tiled using L-shaped triominoes such that the entire grid is covered by triominoes but no triominoes overlap with each other nor the painted cell.

Proof: By Induction

Predicate: P(i) : Any 2^i -by- 2^i grid with one painted cell can be fully tiled using L-shaped triominoes.

Base Case: P(1): 2 -by- 2 grid with one painted square can be filled with L-shaped triominoes. TRUE

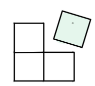




Induction Step: Assume that P(k) is true for all k>0, 2^k * 2^kgrid with one painted square can be filled with L-shaped triominoes.

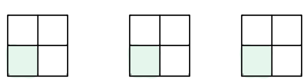
Want to show: P(k+1): 2^{k+1} -by- 2^{k+1}grid with one painted square can be filled with L-shaped triominoes as well.

Consider 2^{k+1} -by- 2^{k+1}grid divided into 4 pieces and one of them is colored.



Since 2^{k+1} * 2^{k+1} = 2^{2k+2} , if divide it by 4 we'll get $\frac{2^{2k+2}}{4} = \frac{2^{2k}*2*2}{4}$

$=2^{2k}$



$=2^k * 2^k$

So now we have only 3 of $2^k * 2^k$*grids left, which we can apply* the inductive hypothesis to each grid and it will work.

Therefore, by mathematical induction, we can conclude that any 2^n -by- 2^n grid with one painted cell can be fully tiled using L-shaped triominoes, where n can be any positive integers.

Pearploy Chaicharoensin 6381278

Task4: TailSumofSquares

Consider the following snippet of Java code:

```
int sumHelper(int n , int a) {
        if (n==0) return a;
        else return sumHelper(n-1, a + n*n);

}
 int sumSqr(int n) { return sumHelper(n, 0); }
```

YourTask: Prove that for $n \geq 1$, sumSqr(n) $\hookrightarrow 1^2+2^2+3^2+\cdots+n^2$. To prove this, use induction to show that sumHelper computes the "right thing." *(Hint: How did we prove* fact_helper *in class?)*

Theorem: sumHelper(n, 0) $\hookrightarrow 1^2+2^2+3^2+\cdots+n^2$,for $n \geq 1$.

Proof: P(n) : sumSqr(n) $\hookrightarrow 1^2+2^2+3^2+\cdots+n^2$,for $n \geq 1$.

Base Case: P(1): sumSqr(1) $\hookrightarrow 1^2$ == 1 TRUE

Induction Step: Assume P(n) is true, thus $1^2+2^2+3^2+\cdots+n^2 = \frac{n(n+1)(2n+1)}{6}$.

Want to show: P(n+1) is also true: sumSqr($n+1$) $\hookrightarrow 1^2+2^2+3^2+\cdots+n^2+(n+1)^2 = \frac{(n+1)(n+2)(2n+3)}{6}$

$$
\begin{aligned}
1^2+2^2+3^2+\cdots+n^2+(n+1)^2 &= 1^2+2^2+3^2+\cdots+n^2+(n+1)^2 \\
&= \frac{n(n+1)(2n+1)}{6} + (n+1)^2 \\
&= \frac{n(n+1)(2n+1)+6(n+1)^2}{6} \\
&= \frac{n(2n+1)+6(n+1)}{6} * (n+1) \\
&= \frac{(2n^2+n)+(6n+6)}{6} * (n+1) \\
&= \frac{2n^2+7n+6}{6} * (n+1) \\
&= \frac{(n+1)(n+2)(2n+3)}{6}
\end{aligned}
$$

Therefore, by mathematical induction, we can conclude that for $n \geq 1$, sumSqr(n) $\hookrightarrow 1^2+2^2+3^2+\cdots+n^2$, sumHelper is able to compute correctly.

Task5: MysteriousFunction

Consider the following Python function foo, which takes as input an integer $n \geq 1$ and returns a tuple of length 2 of integers:

```
def foo(n):
        assert n>=1
        if n == 1:
                return (1, 2)

        else:
                p, q = foo(n-1)

                return (q + p*n*(n+1), q*n*(n+1))
```

Proof: for $n \geq 1$, foo(n) ↪ (p,q) such that $\frac{p}{q} = 1 - \frac{1}{n+1}$

Base Case: foo(1) ↪ (1,2) TRUE

foo(2) ↪ (8,12) TRUE

Induction Step: Assume foo(n) is true, thus $\frac{p_n}{q_n} = 1 - \frac{1}{n+1} = \frac{n}{n+1}$

Then $p_n = n$ and $q_n = n + 1$

Want to show: foo(n+1) is also true, thus $\frac{p_{n+1}}{q_{n+1}} = 1 - \frac{1}{n+2} = \frac{n+1}{n+2}$

Since foo(n+1) is one step after foo(n) it means that

$$\frac{p_{n+1}}{q_{n+1}} = \frac{q_n+p_n*(n+1)*(n+2)}{q_n*(n+1)*(n+2)} = \frac{(n+1)+(n*(n+1)*(n+2))}{(n+1)*(n+1)*(n+2)}$$

$$= \frac{n^3+3n^2+3n+1}{(n+1)^2(n+2)}$$

$$= \frac{(n+1)^3}{(n+1)^2(n+2)}$$

$$= \frac{(n+1)}{(n+2)}$$

Therefore, by mathematical induction, we can conclude that for $n \geq 1$, foo(n) ↪ (p,q) such that $\frac{p}{q} = 1 - \frac{1}{n+1}$

Task7: Midway Tower of Hanoi

You'll begin by showing a useful property. Prove, using mathematical induction, that for any $n \geq 0$,solve_hanoi (n, ...) generates exactly $2^n - 1$ lines of instructions.

Theorem: For any n disks stack on each other where the largest is always at the bottom and the smallest one is at the top. It will generate exactly $2^n - 1$ moves.

Predicate: P(i) : any i disks take $2^i - 1$ lines of instructions.

Base Case: P(1): $2^1 - 1=1$ TRUE since 1 desk only takes 1 move from one pec to another.

Induction Step: Assume P(n) = $2^n - 1$ is true.

Want to show: P(n+1) = $2^{n+1} - 1$ is also true.
Before we will get to move the last piece of the desk(largest), in this case it is n+1 one, we will have to move the one before first: (n+1)-1 ones. But since desk n is larger than previous moved desks, moving to an empty pec is the only choice if we follow the rule, then it will take an extra 1 step.

$2^{n+1} - 1$ = move n disks out + move the desk (n+1) + move n desks to pile on top of the current dest n+1
= P(n) + 1 + P(n)
= $(2^n - 1) + 1 + (2^n - 1)$
= $2 * 2^n$-1
= $2^{n+1} - 1$

Therefore, by mathematical induction, we can conclude that any i well stacked disks take $2^i - 1$ moves from one pec to another.