$$\sum_{i=1}^{200}\left(\sum_{j=1}^{300}(5i^2-j)\right) = \sum_{i=1}^{200}\left(\sum_{j=1}^{300}5i^2 - \sum_{j=1}^{300}j\right)$$

$$\sum_{i=1}^{200}\left(\left(5i^2-1\right)+\left(5i^2-2\right)+\cdots+\left(5i^2-300\right)\right)$$

$$= \sum_{i=1}^{200}\left(300\cdot 5i^2 - 1 - 2 - 3 - \cdots - 300\right)$$

$$= \sum_{i=1}^{200}\left(1500i^2 - \left(1+2+3+\cdots+300\right)\right)$$

$$= \sum_{i=1}^{200}\left(1500i^2 - \frac{300(301)}{2}\right) = \left(\sum_{i=1}^{200}1500i^2\right) - \left(\sum_{i=1}^{200}\frac{300\cdot 301}{2}\right)$$

$$= \left(1500\sum_{i=1}^{200}i^2\right) - \frac{200\cdot 300\cdot 301}{2}$$

by Rosen Table 2

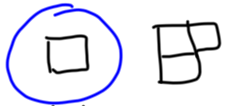
$$= 1500\left(\frac{200\cdot 201\cdot 401}{6}\right) - 200\cdot 150\cdot 301$$

Prove by induction: (p.277, 6ed)

P(n) = "An $2^n \times 2^n$ checkerboard can be tiled by with one corner missing."

(a) P(0) is the statement:
"A 1×1 checkerboard can be tiled by , w/ one corner missing."

(b) P(0) true



(c) what is ind. hyp? P(k)
= "A $2^k \times 2^k$ can be tiled .... 1 left over."

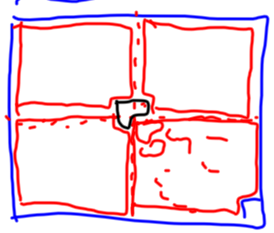
(d) What to show? P(k+1):
"A $2^{k+1} \times 2^{k+1}$ can be tiled ... 1 left over."

(e) Show P(k) → (P(k+1)):

Must show I can tile

2^{k+1}

2^{k+1}



but I know I 2^k can tile 2^k

(f) We know P(0);
We know $\forall k \geq 0$, $P(k) \rightarrow P(k+1)$
So by induction $\forall n. P(n)$.

Strong induction:

P(n): "A chocolate bar w/ n contiguous squares can be broken into singletons requires exactly n-1 breaks."

Base case P(1) = "______ "

Ind. step Want to show $P(k) \to P(k+1)$.

I have a k+1 piece bar; want to show k breaks req'd.

Faulty: I break off one piece to get a singleton plus a k-piece bar. This took one step, plus (by P(k)), k-1 other others were req'd, for a total of k.

Flaw: What if we make a first break different from one piece?

Make any break — into sizes j and (k+1)-j.

If I know P(j) (requires j-1)
and P(k+1-j) (requires k+1-j-1),

Then it means my k+1-piece bar required $1 + (j-1) + (k+1-j-1)$
$= k$ breaks.

This is "strong induction":

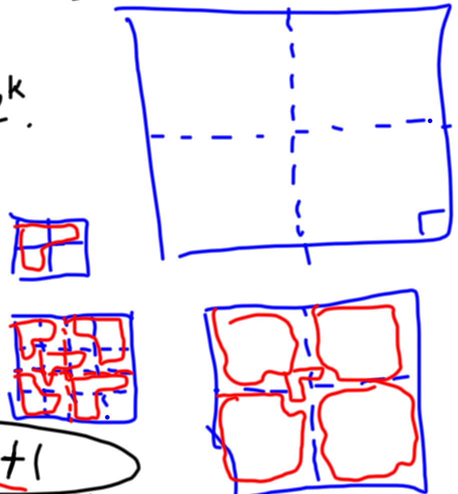
- P(0),
- $\forall k \ (P(0) \wedge P(1) \wedge P(2) \cdots \wedge P(k)) \to P(k+1)$

Then conclude $\boxed{\forall n.\ P(n)}$.

Recursive formulas,
recursive data
structural induction &
recursive algorithms.

Q: In tiling a $\boxed{2^k \times 2^k}$ room w/ tiles,
how many tiles are needed?

Let $f(k)$ be # of tiles needed for $2^k \times 2^k$.



$f(0) = 0$

$f(1) = 1$

$f(2) = 5$

$f(3) = 21$

same

$f(k) = 4 \cdot f(k-1) + 1$

$\boxed{a_k = 4a_{k-1} + 1}$

$f(k-1) + f(k-1) + f(k-1) + f(k-1) + 1$

```
int f (int k) {
  if (k > 0)
     return 4*f(k-1) +1;
  else
     return 0;
}
```

Def'n: A SList is either:
- new EmptyList()

or
- new ConsList(int n, SList rest).

```
abstract class SList {
  int length();
  boolean allPos();
}
class EmptyList extends SList {
}
class ConsList extends SList {
  int n;
  SList rest;
}
```

```
SList l0 = new EmptyList();
SList l1 = new ConsList(7, l0);
SList l2 = new ConsList(3, l1);
```

Def'n of length of a SList:
- length of EmptyList is 0.
- length ConsList(n, l) is 1+length(l)

Write code:

```
class EmptyList {
  int length() { return 0; }
}
class ConsList {
  :
  int length() { return 1+(this.rest).length(); }
}
```

(1 → int, this.rest → SList)

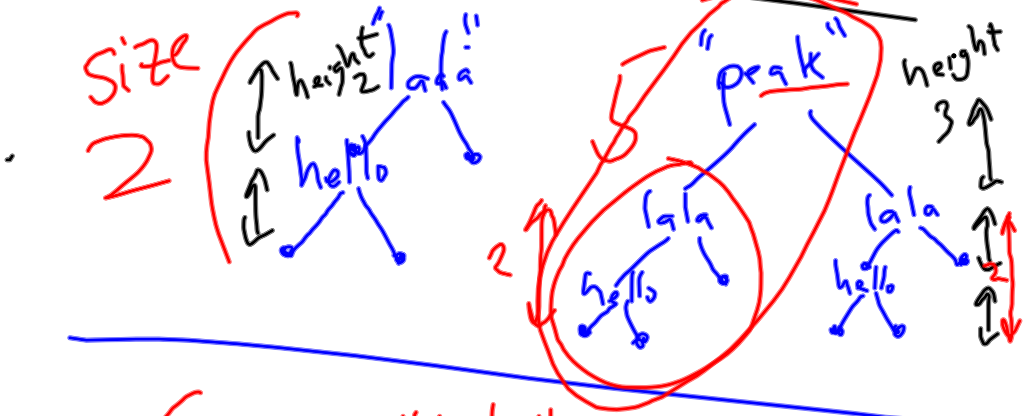
Write method allPos() — are all numbers in the list positive?

Def'n: a Tree is

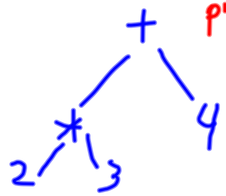
- empty — Size: 0
- new Branch(String s, Tree l, Tree r)

Def'n of size

1 + l.size() + r.size()



(2 * 3) + 4



prefix + * 2 3 4

post 2 3 * 4 +

infix 2*3 + 4

Proof on trees:

$P(t) = $ "t.size() $< 2^{t.\text{height}()}$"

How to prove?

Use structural induction.

Base case: $P(\text{new EmptyTree}())$ holds.

$0 < 2^0 = 1$ true ✓

Inductive step:

$P(\text{k.left}) \wedge P(\text{k.right}) \to P(\text{k})$

t

1. $\text{k.left.size}() < 2^{\text{k.left.height}()}$ by ind hyp.
2. $\text{k.right.size}() < 2^{\text{k.right.height}}$
3. $\text{k.l.s}() + \text{k.r.size}() < 2^{\text{k.l.h}()} + 2^{\text{k.r.h}()}$
4. $1 + \sim < 1 + \sim$
15. $1 + \text{k.l.s}() + \text{k.r.s}() < 2^{1+\max(\text{k.lef.h}(), \text{k.r.h}())}$

16. $1 + \text{k.left.size}() + \text{k.right.size}() < 2^{\text{k.height}}$
17. $\text{k.size}() < 2^{\text{k.height}()}$ by defn of k.size()

$\sim < 1 + 2^{\text{k.l.h}} + 2^{\text{k.r.h}}$

$\leq 1 + 2^{\max(\text{k.l.h}, \text{k.r.h})} + 2^{\max(\text{k.l.h}, \text{k.r.h})}$

$= 1 + 2 \cdot 2^{\max(\sim)}$

$= 1 + 2^{1 + m(\sim)}$

$= 1 + 2^{\text{k.height}}$ by code