Goals

• Write a strong inductive proof
• Identify when multiple cases are needed
Algorithm 1: \text{Sum}(A)

\textbf{Input} : List \( A \) of integers
\textbf{Output} : Sum of the elements of \( A \).

1 \( l = \text{length}(A) \);

// Base Case
2 \textbf{if} \( l \) equals 1 \textbf{then}
3 \hphantom{2} \text{return} \( A[1] \);
4 \textbf{else}
5 \hphantom{2} \text{// Recursive step}
6 \hphantom{3} \text{mid} = l/2, \text{rounded to next lowest integer if not an integer};
7 \hphantom{3} \text{return} \text{Sum}(A[1 : \text{mid}]) + \text{Sum}(A[\text{mid} + 1 : l]);
8 \hphantom{2} \text{// \( A[a : b] \) is a list of \( a \)th to \( b \)th elements of \( A \) inclusive.}
9 \text{end}
Proof By Strong Induction

Prove it takes \( n - 1 \) breaks to reduce an \( n \)-square chocolate bar to \( n \) individual pieces.

(Inductive step: Let \( k \geq \_ \). Assume for strong induction that \( P(j) \) is true for all \( j \) such that \( \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \).)