Induction II

1 Unstacking

Here is another wildly fun 6.042 game that's surely about to sweep the nation!

You begin with a stack of n boxes. Then you make a sequence of moves. In each move, you divide one stack of boxes into two nonempty stacks. The game ends when you have n stacks, each containing a single box. You earn points for each move; in particular, if you divide one stack of height $a + b$ into two stacks with heights a and b, then you score ab points for that move. Your overall score is the sum of the points that you earn for each move. What strategy should you use to maximize your total score?

As an example, suppose that we begin with a stack of $n = 10$ boxes. Then the game might proceed as follows:

Stack Heights										**Score**
$\underline{10}$										
5	$\underline{5}$									25 points
$\underline{5}$	3	2								6
$\underline{4}$	3	2	1							4
2	$\underline{3}$	2	1	2						4
$\underline{2}$	2	2	1	2	1					2
1	$\underline{2}$	2	1	2	1	1				1
1	1	$\underline{2}$	1	2	1	1	1			1
1	1	1	1	$\underline{2}$	1	1	1	1		1
1	1	1	1	1	1	1	1	1	1	1
								Total Score	$=$	45 points

On each line, the underlined stack is divided in the next step. Can you find a better strategy?

1.1 Strong Induction

We'll analyze the unstacking game using a variant of induction called *strong induction*. Strong induction and ordinary induction are used for exactly the same thing: proving that a predicate $P(n)$ is true for all $n \in \mathbb{N}$.

Principle of Strong Induction. Let $P(n)$ be a predicate. If

- $P(0)$ is true, and
- for all $n \in \mathbb{N}$, $P(0)$, $P(1)$, ..., $P(n)$ imply $P(n+1)$,

then $P(n)$ is true for all $n \in \mathbb{N}$.

The only change from the ordinary induction principle is that strong induction allows you to assume more stuff in the inductive step of your proof! In an ordinary induction argument, you assume that $P(n)$ is true and try to prove that $P(n+1)$ is also true. In a strong induction argument, you may assume that $P(0)$, $P(1)$, ..., $P(n-1)$, and $P(n)$ are *all* true when you go to prove $P(n+1)$. These extra assumptions can only make your job easier.

Despite the name, strong induction is actually no more powerful than ordinary induction. In other words, any theorem that can be proved with strong induction can also be proved with ordinary induction. However, an apeal to the strong induction principle can make some proofs a bit easier. On the other hand, if $P(n)$ is easily sufficient to prove $P(n+1)$, then use ordinary induction for simplicity.

1.2 Analyzing the Game

Let's use strong induction to analyze the unstacking game. We'll prove that your score is determined entirely by the number of boxes; your strategy is irrelevant!

Theorem 1. *Every way of unstacking n blocks gives a score of $n(n-1)/2$ points.*

There are a couple technical points to notice in the proof:

- The template for a strong induction proof is exactly the same as for ordinary induction.
- As with ordinary induction, we have some freedom to adjust indices. In this case, we prove $P(1)$ in the base case and prove that $P(1), \ldots, P(n-1)$ imply $P(n)$ for all $n \geq 2$ in the inductive step.

Proof. The proof is by strong induction. Let $P(n)$ be the proposition that every way of unstacking n blocks gives a score of $n(n-1)/2$.

Base case: If $n = 1$, then there is only one block. No moves are possible, and so the total score for the game is $1(1-1)/2 = 0$. Therefore, $P(1)$ is true.

Inductive step: Now we must show that $P(1), \ldots, P(n-1)$ imply $P(n)$ for all $n \geq 2$. So assume that $P(1), \ldots, P(n-1)$ are all true and that we have a stack of n blocks. The first move must split this stack into substacks with sizes k and $n-k$ for some k strictly between 0 and n. Now the total score for the game is the sum of points for this first move plus points obtained by unstacking the two resulting substacks:

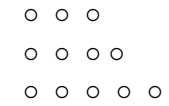
$$\begin{aligned}
\text{total score} &= (\text{score for 1st move}) \\
&\quad + (\text{score for unstacking } k \text{ blocks}) \\
&\quad + (\text{score for unstacking } n-k \text{ blocks}) \\
&= k(n-k) + \frac{k(k-1)}{2} + \frac{(n-k)(n-k-1)}{2} \\
&= \frac{2nk - 2k^2 + k^2 - k + n^2 - nk - n - nk + k^2 + k}{2} \\
&= \frac{n(n-1)}{2}
\end{aligned}$$

The second equation uses the assumptions $P(k)$ and $P(n-k)$ and the rest is simplification. This shows that $P(1), P(2), \ldots, P(n)$ imply $P(n+1)$.

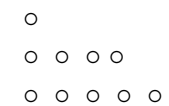
Therefore, the claim is true by strong induction. □

2 The Game of Nim

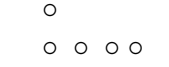
Nim is a game involving two players, some pennies, and mathematical induction. The game begins with a bunch of pennies, arranged in one or more rows. For example, here we have three rows of pennies:



The two players take turns. On each turn, a player must remove one or more pennies, all from a single row. The player who takes the last penny wins. For example, suppose that the first player removes two pennies from the first row:



Now the second player removes all the pennies from the last row, leaving this configuartion:



The first player then takes three pennies from the last row:

∘
∘

The second player is now in trouble; she must take exactly one of these two pennies. Either way, the first player takes the last penny and wins the game.

There is a compact way to describe a configuration in Nim: list the number of pennies in each row. For example, the starting configuartion in the game above is $(3, 4, 5)$. The game then passed through the configurations $(1, 4, 5)$, $(1, 4, 0)$, and $(1, 1, 0)$.

2.1 Nimsums

A Harvard professor, Charles Bouton, discovered the optimal strategy for Nim in 1901. Bouton's strategy relies on a special mathematical operation called a *Nimsum*. The Nimsum of natural numbers $c_1, \ldots, c_k$ is itself a natural number that is computed as follows:

- Regard $c_1, \ldots, c_k$ as binary numbers.
- The i-th bit of the Nimsum is the ***xor*** of the i-th bits of $c_1, \ldots, c_k$.

(Xor is pronounced "eks-or" and is short for "exclusive-or". The xor of bits b_1 and b_2 is denoted $b_1 \oplus b_2$ and defined as follows:

b_1	b_2	$b_1 \oplus b_2$
0	0	0
0	1	1
1	0	1
1	1	0

As a consequence of this definition, the xor of bits $b_1, \ldots, b_k$ is 0 if the sum of the b_i is even and 1 if the sum is odd. For example, $1 \oplus 0 \oplus 1 \oplus 1 = 1$ because $1 + 0 + 1 + 1 = 3$ is odd, but $1 \oplus 1 \oplus 0 \oplus 0 = 0$ because $1 + 1 + 0 + 0 = 2$ is even.)

As an example, the Nimsum of 3, 4, and 5 is computed as follows:

$$\begin{array}{rcl} 3 & = & 011 \\ 4 & = & 100 \\ 5 & = & 101 \\ \hline & & 010 \quad = \quad 2 \end{array}$$

Here, we xor each column of bits to obtain the Nimsum 010, which is the binary representation of 2.

2.2 The Winning Strategy

Suppose that $(c_1, \ldots, c_k)$ is a configuration in Nim; that is, there are a total of k rows, and the i-th row contains c_i pennies. This configuration is called:

- *safe* if the Nimsum of $c_1, \ldots, c_k$ is nonzero
- *unsafe* if the Nimsum of $c_1, \ldots, c_k$ is zero

In this way, all possible configurations in Nim are now partitioned into two groups: the safe configurations and the unsafe configurations. Bouton's strategy is as follows:

- If you see a safe configuration on your turn, then select a move that leaves your opponent with an unsafe configuration.
- If you see an unsafe configuration on your turn, you're doomed. Every possible move leaves your opponent with a safe configuration; select among them arbitrarily and prepare to lose.

Thus, if the first player sees a safe position, she selects a move that leaves the second player an unsafe position. Every move available to the second player then leaves the first player with another safe position. This cycle repeats until the first player wins the entire game.

2.3 An Example

Let's use Bouton's strategy in one example game before trying to prove a general theorem. The bizarre French movie "Last Year at Marienbad" features a Nim game beginning in the configuration $(1, 3, 5, 7)$. Let's analyze this configuration by computing the Nimsum:

$$\begin{array}{rcl} 1 & = & 001 \\ 3 & = & 011 \\ 5 & = & 101 \\ 7 & = & 111 \\ \hline & & 000 \quad = \quad 0 \end{array}$$

Since the Nimsum is zero, this is a hopeless situation for the first player. In particular, no matter what she does, the second player is left with a safe configuration; that is, one in which the Nimsum is nonzero. For example, suppose that the first player removes the entire fourth row. Then the Nimsum becomes:

$$\begin{array}{rcl} 1 & = & 001 \\ 3 & = & 011 \\ 5 & = & 101 \\ 0 & = & 000 \\ \hline & & 111 \quad = \quad 7 \end{array}$$

This is a safe configuration. To maintain her advantage, the second player needs to leave the first player in an unsafe position; that is, a position with Nimsum zero. Only one move accomplishes this: removing three pennies from the third row:

$$\begin{array}{rcl} 1 & = & 001 \\ 3 & = & 011 \\ 2 & = & 010 \\ 0 & = & 000 \\ \hline & & 000 \quad = \quad 0 \end{array}$$

Now the first player is again confronted with an unsafe position, one with a Nimsum of zero. No matter what she does, the second player will be left with a safe position. For example, suppose that the first player takes all three pennies from the second row. Then the second player is left with:

$$\begin{array}{rcl} 1 & = & 001 \\ 0 & = & 000 \\ 2 & = & 010 \\ 0 & = & 000 \\ \hline & & 011 \quad = \quad 3 \end{array}$$

Sure enough, this is a safe position. One again, the second player needs to leave the first player in an unsafe position. Removing one penny from the third row accomplishes this:

$$\begin{array}{rcl} 1 & = & 001 \\ 0 & = & 000 \\ 1 & = & 001 \\ 0 & = & 000 \\ \hline & & 000 \quad = \quad 0 \end{array}$$

The first player is clearly going to lose. She must take one of the two remaining pennies, leaving the second player to take the other and win the game.

2.4 Proof of Correctness

Now let's prove that Bouton's strategy works. We'll need two lemmas.

Lemma 2. *If a player sees an unsafe configuration on her turn, then every possible move leaves her opponent with a safe configuration.*

Proof. Suppose that the player sees an unsafe configuration; that is, a configuration $(c_1, \ldots, c_k)$ with Nimsum zero. Then the player must remove pennies from some row j, leaving $c'_j < c_j$ pennies behind. In binary, the numbers c_j and c'_j must differ in some bit position. But then the Nimsum of the resulting configuration has a 1 in that bit position. Therefore, the Nimsum is nonzero, meaning that the resulting configuration is safe. □

Lemma 3. *If a player sees a safe configuration on her turn, then she has a move that leaves her opponent with an unsafe configuration.*

Proof. Suppose that the player sees a configuration $(c_1, \ldots, c_k)$ with Nimsum $s \neq 0$.

Let i be the position of the most-significant 1 in the binary representation of s. Since the i-th position of s is nonzero, there must be some row size c_j that is nonzero in the i-th position as well. Suppose that the player removes all but c'_j pennies from the j-th row, where c'_j differs from c_j in exactly those positions where s has a 1.

We must check that c'_j, the number of pennies the player leaves behind, is less than c_j, the number of pennies originally in the j-th row. This is true because c_j has a 1 in the i-th position, c'_j has a 0 in this position, and the two numbers agree in all higher positions.

Modifying the game configuration by changing c_j by c'_j reduces the Nimsum to zero, because c_j and c'_j disagree in exactly those positions where the original Nimsum had a 1. □

An example may clarify the preceding argument. Suppose that a player sees the configuration $(1, 2, 4, 4)$. We can compute the Nimsum as follows:

$$\begin{array}{rcl} 1 & = & 001 \\ 2 & = & 010 \\ 4 & = & 100 \\ 4 & = & 100 \\ \hline & & 011 \quad = \quad 3 \end{array}$$

The Nimsum is 011 in binary. The most significant 1 is in the second position from the right. The number of pennies in the second row, 010 in binary, also has a 1 in this position. We should leave 001 pennies in this row, since this differs from the number of pennies there now (010) in every position where the Nimsum (011) has a 1. Now we compute the Nimsum of the resulting configuration:

$$\begin{array}{rcl} 1 & = & 001 \\ 1 & = & 001 \\ 4 & = & 100 \\ 4 & = & 100 \\ \hline & & 000 \quad = \quad 0 \end{array}$$

As expected, this is an unsafe position.

We're ready to prove the main result.

Theorem 4. *If the current player has a safe position, then she can guarantee a win. Otherwise, the other player can guarantee a win.*

The proof uses induction, with the statement of the theorem as the induction hypothesis. This hypothesis is quite complicated! Suppose that we define three smaller propositions:

$$\begin{aligned} A &= \text{current player has a safe position} \\ B &= \text{current player can guarantee a win} \\ C &= \text{next player can guarantee a win} \end{aligned}$$

In these terms, the induction hypothesis is $(A \to B) \wedge (\overline{A} \to C)$. Induction hypotheses containing implications are always a bit tricky, but this one has *two* implications! Some square-bracketed comments are included in the proof to clairfy the structure.

Proof. The proof is by strong induction on n, the total number of pennies remaining. Let $P(n)$ be the proposition that in all Nim configurations with n pennies, if the current player has a safe position, then she can guarantee a win and, otherwise, the other player can guarantee a win.

Base case. Suppose that $n = 0$; that is, no pennies remain. This is an unsafe position. Therefore, the first implication is trivially true because the if-part is false. The second implication is also true, because when the current player is left with no pennies, the other player has just won the game. [In other words, the proposition $(A \to B) \wedge (\overline{A} \to C)$ is true because A is false and C is true.]

Inductive step. We must show that for all $n \geq 0$, the propositions $P(0), \ldots, P(n)$ imply $P(n+1)$. Assume $P(0), \ldots, P(n)$ and consider a configuration with $n+1$ pennies remaining. Now we must show two things:

1. *If the configuration is safe, the current player can guarantee a win.* Suppose that the configuration is safe. By Lemma 3, the current player has a move that leaves the next player with an unsafe configuration involving fewer than $n+1$ pennies. If $m < n+1$ pennies remain, then assumption $P(m)$ implies that the player *after* the next player (which is the current player) can guarantee a win. [This shows $A \to B$.]

2. *If the configuration is unsafe, the next player can guarantee a win.* Suppose that the configuration is unsafe. By Lemma 2, every move leaves the next player with a safe configuration involving $m < n+1$ pennies. The assumption $P(m)$ then implies that the next player can guarantee a win. [This shows $\overline{A} \to C$.]

Putting these arguments together gives $P(n+1)$. [That is, $A \to B$ and $\overline{A} \to C$ together imply $(A \to B) \wedge (\overline{A} \to C)$.] Therefore, the theorem follows by the principle of induction. □