

This article is written by students. It may include omissions and imperfections,
as far as possible reported by our reviewers in the editing notes.

Turning over coins

Year 2022-2023

Alexia Bucșa, Bianca Tavaszi, Carlos Pop, Clara Bob, Gloria Dobra, Maria Breaz, Rianna Tancel (10th grade) and Beatrice Teoc, Briana Filimon, Emma Vasii, Irina Paul, Irina Terec, Sara Crișan (11th grade).

School : Colegiul Național Emil Racoviță, Cluj-Napoca

Teacher : Ariana Văcărețu

Researcher : Dr. George Țurcaș, , Babeș-Bolyai University of Cluj-Napoca

Abstract

This topic is about sorting a stack of coins. First of all, we present the topic statement. Then, we show the steps in our approach and our findings. At the end, we link the [programs](#) we created based on the algorithm we discovered, as well as a [web application](#).

Table des matières

1	The problem	1
2	Methods and results	2
2.1	First steps into the problem	2
2.2	Total number of possible combinations	2
2.3	Total number of possible turns	3
2.4	Algorithm	4
2.5	Average number of turns	5
2.6	Another version of the problem	6
2.7	Programs	6
2.8	App	7
3	Conclusions	7
4	Bibliography	7
	Notes d'édition	7

1. The problem

We have a certain number of stacked coins. A coin has a red side and a black side. A stack of coins can be turned over partly, but only by starting from the top of the stack. We considered the front side to be the one coloured in black, except for the special condition, where we do not specify which side will the coins be facing at the end.

How many times do you have to repeat this operation to get all the coins on the front side? How should we proceed in general?



A related problem is the "Pancake Sorting Problem". Instead of a stack of coins, a stack of differently sized pancakes needs to be arranged so that the smallest pancake is on the top and the pancakes get larger from top to bottom. A famous paper treating this subject is "[Bounds for Sorting by Prefix Reversal](#)", *Discrete Mathematics*, Volume 27, Issue 1, 1979, Pages 47-57, co-authored by William H. Gates and Christos H. Papadimitriou.

2. Methods and results

2.1. First steps into the problem

Each of us tried to visualise the problem, so we made some coins out of various materials such as plastic and foam. The plastic ones were 3D printed, whereas the foam ones were cut from glitter foam sheets. The plastic ones have one side white instead of red because this color was not available.



2.2. Total number of possible combinations

We tried out cases on paper and then created a [Google sheets](#) file containing the possible combinations of n coins [1].

We noticed that for one coin there are two possible combinations, which can be written as $(1 + 1)^1$ or 2^1 , for two coins there are four possible combinations, which can be written as $(1 + 1)^2$ or 2^2 , for three coins there are 8 possible combinations, which can be written as $(1 + 1)^3$ or 2^3 , for four coins there are 16 possible combination, which can be written as $(1 + 1)^4$ or 2^4 and so on.

represents the black side of the coin and 0 represents the red side. A 'change' consists in the alternations between 1 and 0. In other words, a 'change' is when two coins are facing differently. For example, the string 100101 has 5 'changes' : 1 00 1 0 1 [3]. Therefore, if a stack of coins has x 'changes', we would have to make (x-1) turns if the bottom coin is 1 and x turns if the bottom coin is 0 to reach our desired goal, which is to have all coins with their black side up. This is because we have to make one final turn after our stack consists of just 0's.

2.4. Algorithm

We will present an algorithm which takes as input a sequence of 1's and 0's, models the coins and turns them into a sequence consisting of just 1's, while performing only the transformations that were specified in the description of the problem. Our algorithm checks whether or not the top coin is the same as the second one. If they are different, the algorithm turns the top coin and goes on. If not, it goes on and checks if the second and third coin are different, in which case it turns the first two and then moves on.

For example, $x = 5$:

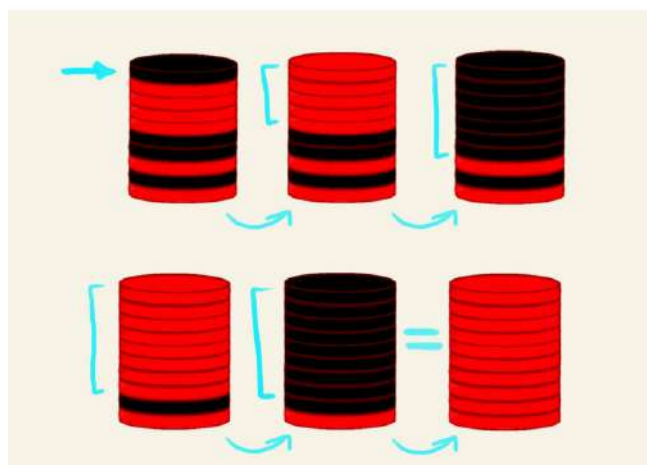
0	1	0	1	0	1	
0	1	0	1	0	1	
1	1	0	1	0	1	
0	0	0	1	0	1	
1	1	1	1	0	1	
1	1	1	1	0	1	
0	0	0	0	0	1	
	1	2	3	4	5	turns

CASE ONE : the bottom coin is 0, so we will make 5 turns

1	0	1	0	1	
0	0	1	0	1	
1	1	1	0	1	
0	0	0	0	1	
1	1	1	1	1	
1	1	1	1	1	
1	1	1	1	1	
	1	2	3	4	turns

CASE TWO : the bottom coin is 1, so we will make 4 turns

Another example :



This image shows the coins represented only by the side facing up. In this case there are 10 coins and they are initially arranged 1000011010. The first coin is flipped, then the first 5 coins, followed by the

first 7 and the first 8, and finally the first 9. The resulting stack has all the coins with their red side up. For all the coins to be with their black side up, the stack needs to be flipped one more time. This final action is not depicted in the image.

2.5. Average number of turns

Performing some experiments, we conjecture a formula for the average number of turns needed to arrange n coins with the front side up. We believe that for n coins the average number of turns is $\frac{n}{2}$.

For example, $n = 3$:

CASE ONE	$\begin{array}{c c} 1 & \\ \hline 1 & \\ \hline 1 & \end{array}$	\Rightarrow number of turns = 0
CASE TWO	$\begin{array}{c cc} 0 & 1 & \\ \hline 0 & 1 & \\ \hline 0 & 1 & \end{array}$	\Rightarrow number of turns = 1
CASE THREE	$\begin{array}{c cc} 0 & 1 & \\ \hline 1 & 1 & \\ \hline 1 & 1 & \end{array}$	\Rightarrow number of turns = 1
CASE FOUR	$\begin{array}{c cc} 0 & 1 & \\ \hline 0 & 1 & \\ \hline 1 & 1 & \end{array}$	\Rightarrow number of turns = 1
CASE FIVE	$\begin{array}{c ccc} 1 & 0 & 1 & \\ \hline 0 & 0 & 1 & \\ \hline 0 & 0 & 1 & \end{array}$	\Rightarrow number of turns = 2
CASE SIX	$\begin{array}{c ccc} 1 & 0 & 1 & \\ \hline 0 & 0 & 1 & \\ \hline 1 & 1 & 1 & \end{array}$	\Rightarrow number of turns = 2
CASE SEVEN	$\begin{array}{c ccc} 1 & 0 & 1 & \\ \hline 1 & 0 & 1 & \\ \hline 0 & 0 & 1 & \end{array}$	\Rightarrow number of turns = 2
CASE EIGHT	$\begin{array}{c cccc} 0 & 1 & 0 & 1 & \\ \hline 1 & 1 & 0 & 1 & \\ \hline 0 & 0 & 0 & 1 & \end{array}$	\Rightarrow number of turns = 3

Therefore, the following cases can be observed :

- 1 case $\binom{3}{0}$ with 0 turns
- 3 cases $\binom{3}{1}$ with 1 switch
- 3 cases $\binom{3}{2}$ with 2 turns
- 1 case $\binom{3}{3}$ with 3 turns

The average number of turns for this example ($n = 3$) is :

$$\frac{0 \cdot 1 + 1 \cdot 3 + 2 \cdot 3 + 3 \cdot 1}{1 + 3 + 3 + 1} = \frac{0 \cdot \binom{3}{0} + 1 \cdot \binom{3}{1} + 2 \cdot \binom{3}{2} + 3 \cdot \binom{3}{3}}{\binom{3}{0} + \binom{3}{1} + \binom{3}{2} + \binom{3}{3}} = \frac{12}{8} = \frac{3}{2}$$

We can calculate the average number of turns for a stack of n coins by dividing the total number of turns by the total number of cases, obtaining the following formula :

$$\frac{\text{total number of turns}}{\text{total number of cases}} = \frac{0 \cdot \binom{n}{0} + 1 \cdot \binom{n}{1} + 2 \cdot \binom{n}{2} + \dots + n \cdot \binom{n}{n}}{\binom{n}{0} + \binom{n}{1} + \binom{n}{2} + \dots + \binom{n}{n}} = \frac{n}{2}$$

2.6. Another version of the problem

One of the teams analysed a different version of this problem, adding the restriction that, at each step, at least two coins have to be turned at once. This changes the algorithm and adds cases which have no solution.

To solve the stack, we first check if there are at least two consecutive coins facing the same way. If not, the case is not solvable, given the restriction. Then we search for the first group of at least two consecutive coins facing the same way and we flip those coins and the ones before them. We continue until all the coins are facing the same way.

The total number of turns : After the first move, we count the number of changes. The total number of turns is the number of changes plus 1.

Otherwise, when there are no consecutive coins facing the same way, we flip the first three coins and we arrive to the previous situation.

The total number of turns : Because we made one more move compared to the previous case, we will count the changes after the second move. The total number of turns is the number of changes plus 2. It is also equal to the number of changes before making any switch.

As we said before there are some cases which cannot be solved, given the restriction. These are 10, 01, 101, 010, because no matter how we flip the coins there will never be any consecutive coins facing the same way.

Notice that here we do not specify which side we want the coins to be facing at the end.

For example, for 7 coins : [4]

1	1	0	1	0	1	
0	1	0	1	0	1	
1	0	0	1	0	1	
0	1	1	1	0	1	
0	0	0	0	0	1	
1	1	1	1	1	1	
1	1	1	1	1	1	
	1	2	3	4	5	turns

Firstly, we find the first two consecutive coins facing the same way, the ones on 4th and 5th positions. Then we flip the first 5 coins. We continue in the same matter until all the coins are with their black side up.

Another example, for 7 coins :

1	0	1	0	1	0	1	
0	1	1	0	1	0	1	
1	0	0	0	1	0	1	
0	0	1	1	1	0	1	
1	1	1	1	1	0	1	
0	0	0	0	0	0	1	
1	1	1	1	1	1	1	
	1	2	3	4	5	6	turns

Because no consecutive coins are facing the same way, the first step is to flip the first 3 coins. Then we proceed similarly to the first example.

2.7. Programs

Each team wrote a program in C++ or Python which takes as input the number of coins and their order in the stack. The [programs](#) arrange the stack, show the steps and the total number of moves. There is also the [program](#) for the special condition that at least two coins have to be turned.

2.8. App

We programmed a [web application](#) using JavaScript, HTML and CSS to make it easier to visualize solving a stack of coins. To be able to create 3D objects (the coins) and handle them, as well as to use lights and cameras, we used Babylon.js, which is a javascript open-source game and rendering engine. The app shows each step and the total number of moves required to arrange the stack.

To access the app click on the hyperlink above, go to homebase.ro/game or scan the following QR code.



3. Conclusions

In conclusion, we found a formula for the total number of possible combinations of n coins, as well as for the total number of possible turns. In addition, we found an algorithm to arrange the stack, we created computer programs and a web application.

4. Bibliography

Binomial theorem : https://en.wikipedia.org/wiki/Binomial_theorem

Pascal's triangle : https://en.wikipedia.org/wiki/Pascal%27s_triangle

Book "Introduction to Counting and Probability" by David Patrick, 2005 :

<https://artofproblemsolving.com/store/book/intro-counting>

Pancake sorting problem : https://en.wikipedia.org/wiki/Pancake_sorting

Article "Bounds for Sorting by Prefix Reversal", Discrete Mathematics, Volume 27, Issue 1, 1979, Pages 47-57,, co-authored by William H. Gates & Christos H. Papadimitriou :

<http://home.ustc.edu.cn/ustcsh/py2016/data/GP79.pdf>

Notes d'édition

[1] Readers will be well advised to open the link of the Google sheet to better understand the situation.

[2] This proof by induction is not very illuminating, and neither is the binomial formula. The reader may find it easier to convince himself of the truth of this formula by seeing it as the counting of the leaves of a binary tree.

[3] It is not very clear and seems to be 4. Or perhaps there is ambiguity about the initial conditions.

[4] In this example, a drawing could indicate the location of the cut to help understand the movements. For example, it is not very clear that the first movement cuts at 1010011, although this is explained later in the text.