

|  |  |
| --- | --- |
| name | id |
| Mohamed Hossam El-Din El-Sayed | 1808581 |
| Mohamed Hesham Alsaied Albendary | 1808442 |
| Ahmed Salah Abdelhak | 1803820 |
| Ahmed Reda Ibrahim Taha Elawam | 1802745 |
| Mohamed Ashraf Farouk Ibrahim | 1805188 |
| Mohamed Ahmed Mahmoud Asran | 1808040 |

Ain Shams University  
Faculty of Engineering  
Computer and Systems Engineering  
CSE332s: Design and Analysis of Algorithm

Contents

[Task 1 1](#_Toc102002707)

[Assumptions 1](#_Toc102002708)

[Problem Description 1](#_Toc102002709)

[Detailed Solution 2](#_Toc102002710)

[Complexity Analysis 5](#_Toc102002711)

[Comparison With Another Algorithm//TODO: INSERT THE REFERENCE 6](#_Toc102002712)

[Sample Output 6](#_Toc102002713)

[Conclusion 7](#_Toc102002714)

//TODO: INSERT TABLE OF FIGURES, TABLES

# Task 1

## Assumptions

User will input correct data type whenever they are asked for input (ex. When asked to enter number of rows, they won’t enter a string for example). Also, the centers of the coins are assumed to be at the points of an equilateral triangular lattice

## Problem Description

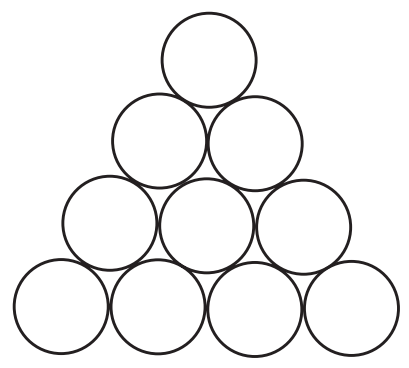
Inverting a Coin Triangle Consider an equilateral triangle formed by closely packed pennies or other identical coins like the one shown in Figure 1. It’s required to use iterative improvement method to design an algorithm to flip the triangle upside down in the minimum number of moves if on each move it’s possible to slide one coin at a time to its new position.

Figure 1

## Detailed Solution

A trivial solution to flip a triangle with rows is to just move all the coins to their new positions, which takes number of steps equal to number of coins in that triangle or 10 steps when applied to Figure 1(obviously inefficient!)

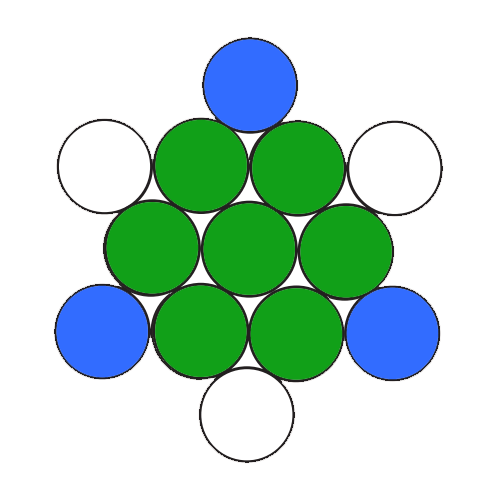
To flip the pyramid in the least number of moves, we will try to overlap both starting and ending positions together, this way the coins marked in green in Figure 2,will not be moved at all. And the ones marked in blue only will be moved to their new positions “Marked with white”. So, in Figure 1, we will need to make only 3 moves

Figure 2

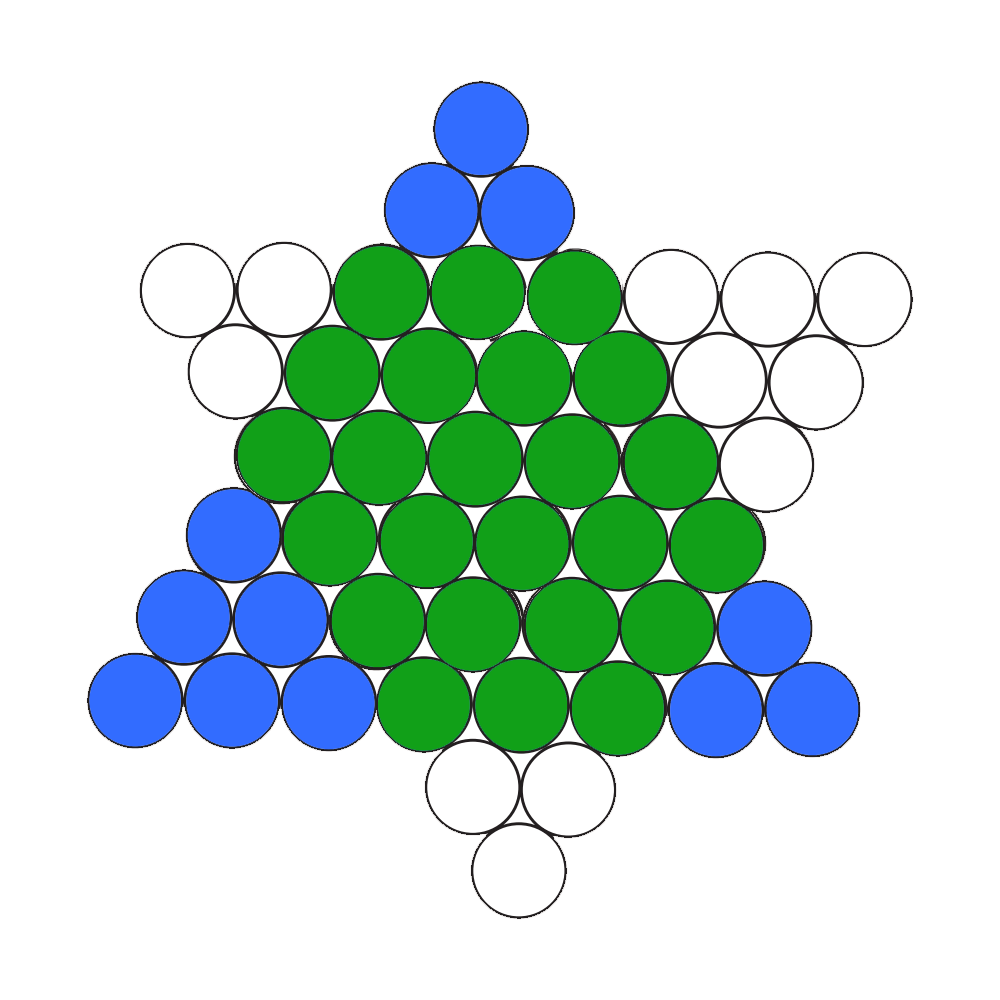
Let’s draw this triangle with 8 rows

Figure 3

If we keep on drawing more and more triangles, we will find out that

1. The small blue triangle in the upper corner can always be moved to the corresponding small white triangle in the bottom
2. The small blue triangle in the bottom left corner can always be moved to the corresponding small white triangle in the upper right corner
3. The small blue triangle in the bottom right corner can always be moved to the corresponding small white triangle in the upper left corner

Table 1 shows the relation between number of rows and the number of rows of each of the small triangles

Table 1

|  |  |  |  |
| --- | --- | --- | --- |
| Number of rows | Number of rows in left triangle | Number of rows in upper triangle | Number of rows in right triangle |
| 1 | 0 | 0 | 0 |
| 2 | 1 | 0 | 0 |
| 3 | 1 | 1 | 0 |
| 4 | 1 | 1 | 1 |
| 5 | 3 | 1 | 1 |
| 6 | 3 | 3 | 1 |
| 7 | 3 | 3 | 3 |
| 8 | 6 | 3 | 3 |

It’s easy to prove that using data obtained in Table 1

Pseudo-code

ALGORITHM InvertTriangle(n)

    //INPUT: n - number of rows

    //OUTPUT: Inverted triangle

    //REQUIREMENTS: n > 0

    a <- ⌊(n + 1) / 3⌋

    for i <- 1 to a do

        move one coin from bottom left triangle to upper right

        print triangle

    b <- ⌊(n) / 3⌋

    for i <- 1 to b do

        move one coin from top triangle to bottom

        print triangle

    c <- ⌊(n - 1) / 3⌋

    for i <- 1 to c do

        move one coin from bottom right triangle to upper left

        print triangle

    steps <- a + b + c

    return steps

## Complexity Analysis

Complexity for each function

printTriangle:

init:

sum:

options:

mainmenu:

To calculate total complexity, recall that we make moves and in each move, we print the triangle. So, in total complexity is , since are all functions in , total complexity is . However, the problem can be solved in if we calculate only number of steps as without printing the triangle after each step

## Comparison With Another Algorithm//TODO: INSERT THE REFERENCE

Another way to calculate number of steps which I found online is by using this formula , this formula and my solution give the same number of steps for all values of

## Sample Output

Figure 4

Now we will try but since it will produce a very long output, I’ll go to options and disable step by step output so that we can see only number of steps as shown in Figure 5

## Conclusion

Figure 5

Inverting a triangle of coins using this algorithm uses iterative improvement since on each step we are getting closer to the goal result by doing the same step over and over (moving one coin between two small triangles)