# Chapter 15. Solving the Tower of Hanoi problem using the recursive method

## 15.1 Analysis of the Tower of Hanoi problem»

Given one rod with disks of different sizes and two empty rods. It is necessary to move the disks from one rod to another, you can only move one disk at a time, you can only install a smaller disk on a larger one. Implement a program that determines permutations of these disks using the least number of moves.

Let's consider an algorithm for solving the problem for four disks. It is necessary to move them from rod 1 to rod 3. First you need to release the base of the tower - the blue disk, following these steps (figure 15.1):

1. The orange disk moves from rod 1 to rod 2
2. The green disk moves from rod 1 to rod 3
3. The orange disc moves to pin 3 on top of the green disc, after which the top of the tower is already on the desired pin.
4. The purple disk moves to rod 2

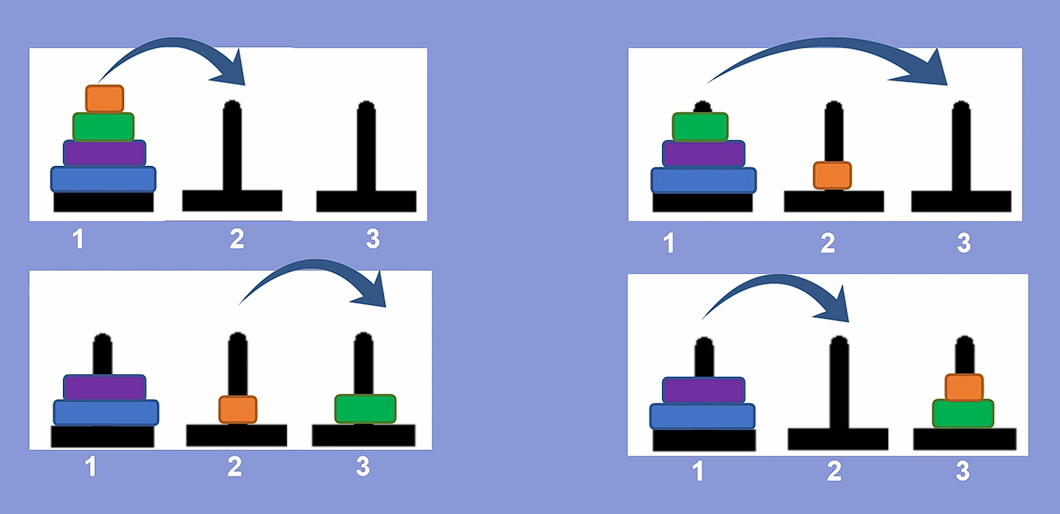


Figure 15.1 – Algorithm for releasing the base of the tower – the blue disk

As a result, the base of the tower – the blue disk – has become free, now it must be moved to rod 3, first it must be freed by performing the following steps:

1. The orange disk moves to rod 1
2. The green disk moves to rod 2
3. The orange disk moves to rod 2
4. The blue disk moves to rod 3

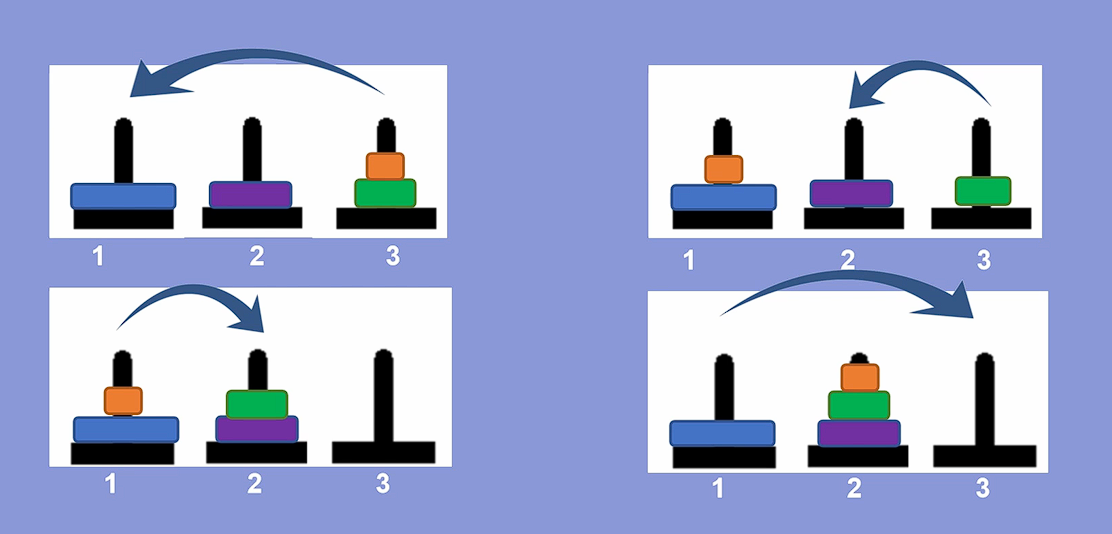


Figure 15.2 – Algorithm for moving the base of the tower - the blue disk to the desired rod

The base of the tower has been successfully moved to rod 3, after which it is necessary to move the next largest disk is purple, for this you need to perform the following steps (figure 15.3):

1. The orange disk moves to rod 3
2. The green disk moves to pin 1
3. The orange disk moves to rod 1
4. The purple disk moves to rod 3

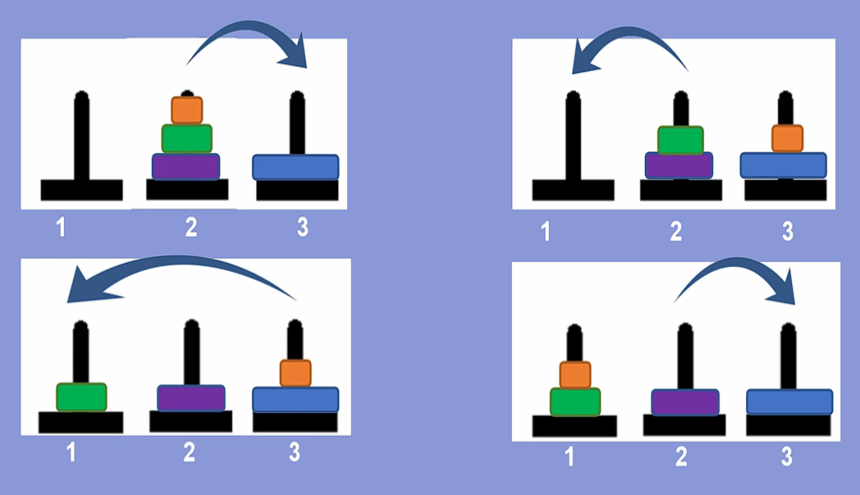


Figure 15.3 – Algorithm for moving the purple disk to the desired rod

To complete the solution to the problem, you need to move the remaining two disks to rod 3; to do this, you must perform the following steps (figure 15.4):

1. The orange disk moves to rod 2
2. The green disk moves to pin 3
3. The orange disk moves to rod 3

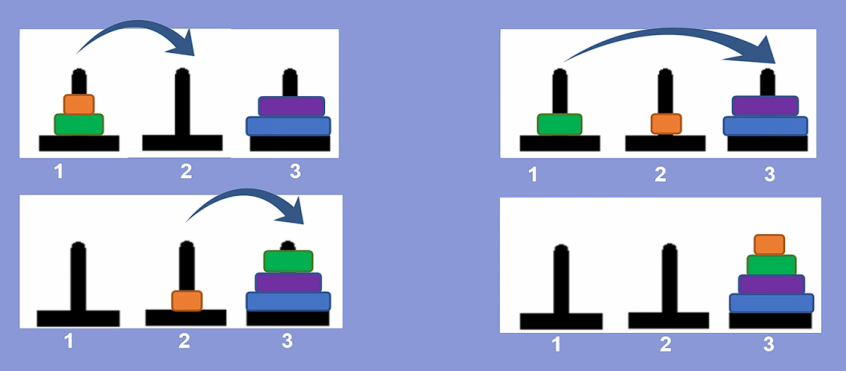


Figure 15.4 – Algorithm for moving the green and orange disks to the desired rod

Thus, the Tower of Hanoi problem with four disks was solved.

The presented solution allows us to identify patterns that can be used to create a general solution to the Tower of Hanoi problem. It was revealed that the problem should be solved not from the beginning, but from the end. To move all the disks to the desired rod, you must first move the bottom disk onto it; this can only be done when n – 1 disks are on a free rod [8].

General algorithm for solving the problem:

1. n – 1 disks are moved to a free rod (rod 2)
2. The nth disk moves to the desired rod (rod 3)
3. n – 1 disks per desired rod (rod 3)

To move n – 1 disks, you need:

1. n – 2 disks are moved to a free rod (rod 2)
2. n – 1 disks are moved to the desired rod (rod 3)
3. n – 2 discs are moved to the desired rod (rod 3)

The recursive algorithm continues until n reaches zero.

## 15.2 Solving the Tower of Hanoi problem for a given number of disks.

Based on the general algorithm, algorithms for solving the problem for four, three and two disks are presented.

Algorithm for solving the problem for four disks (figure 15.5):

1. Three discs are placed on a free rod (rod 2)
2. The fourth disk is transferred to the desired rod (rod 3)
3. Three discs are transferred to the desired rod (rod 3)

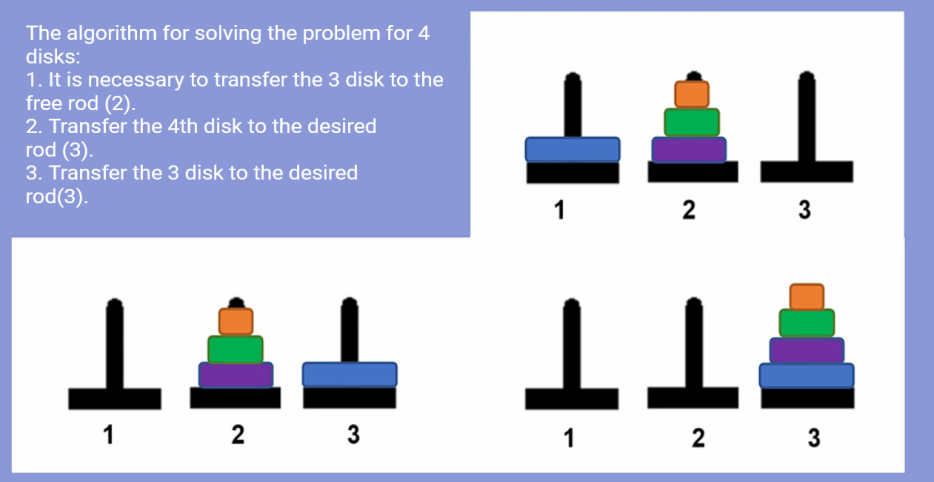


Figure 15.5 – Algorithm for solving the Tower of Hanoi problem for four disks

Algorithm for solving the problem for three disks (figure 15.6):

1. Two disks are placed on a free rod (rod 2)
2. The third disk is transferred to the desired rod (rod 3)
3. The two discs are placed on the desired rod (rod 3)

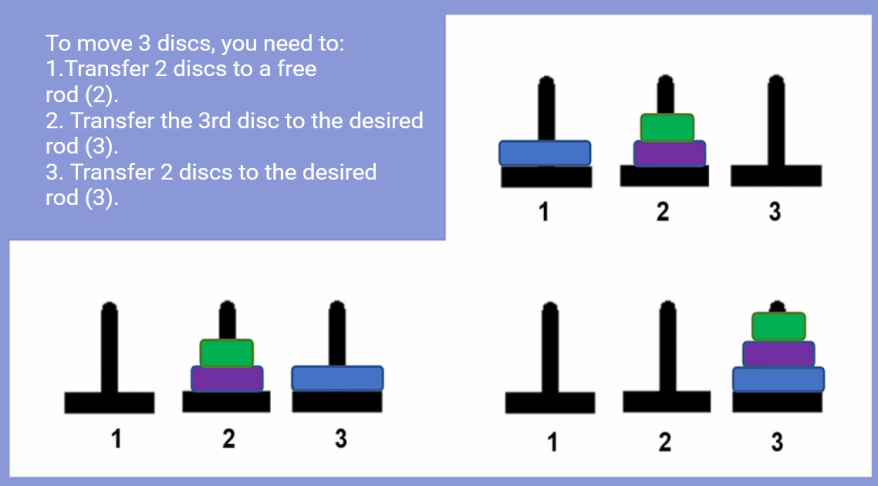


Figure 15.6 – Algorithm for solving the Tower of Hanoi problem for three disks

Algorithm for solving the problem for two disks (figure 15.7):

1. The first disk moves to the free rod (rod 2)
2. The second disk is moved to the desired rod (rod 3)
3. The first disk is moved to the desired rod (rod 3)

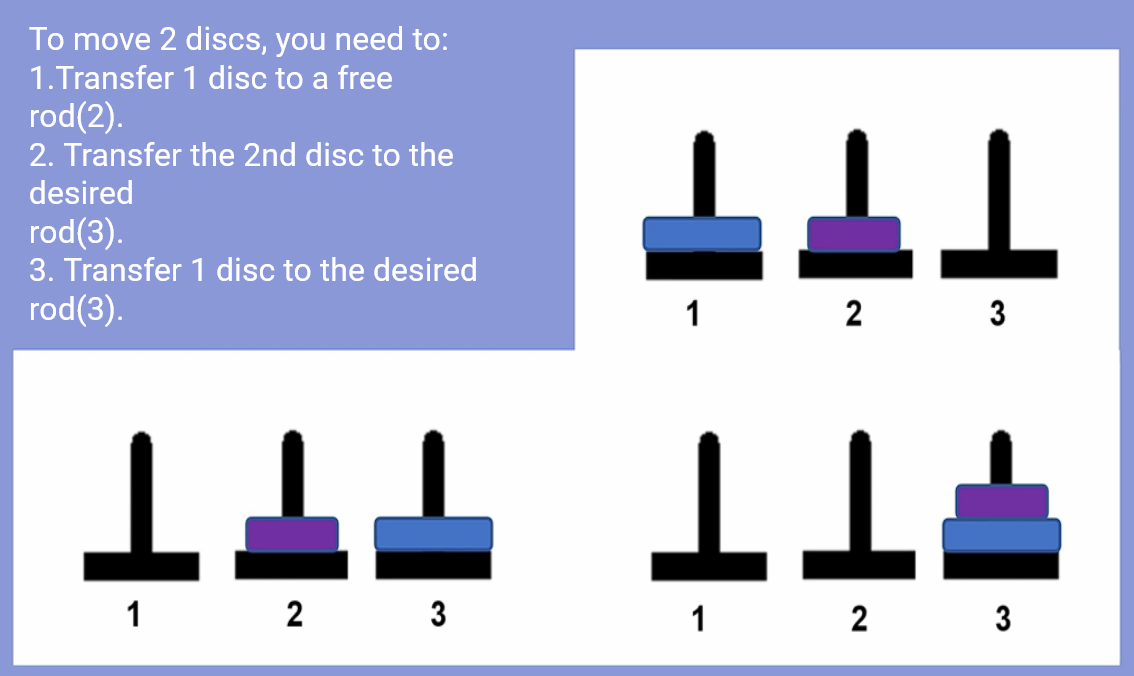


Figure 15.7 – Algorithm for solving the Tower of Hanoi problem for two disks

## 15.3 Code for solving the Tower of Hanoi problem

To write code for solving the problem, a program block diagram was drawn up, shown in figure 15.8.

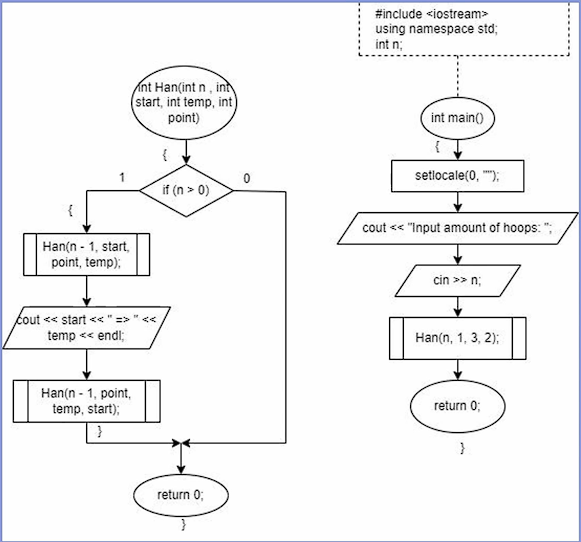


Figure 15.8 – Flowchart of the program for solving the “Tower of Hanoi” problem

A Han function is created that takes as parameters the integer variables int n (number of disks), int start (initial rod), int point (auxiliary rod) and int temp (auxiliary rod). First, the body of the function checks to see if the number of disks is equal to zero. If it is non-zero, the function recursively calls itself to move n - 1 disks to the auxiliary rod, then prints a message about which rod the disks were moved from to which. After displaying information about which rod the disk was moved from, the function calls itself again, but this time to move n - 1 disks to the desired rod, the program code is shown in figure 15.9. As you can see, the function calls itself more than once [13]. This simple recursion is called cascade.

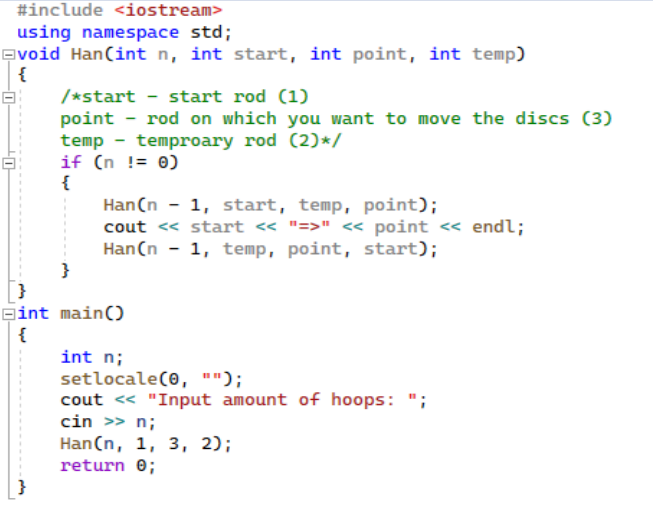


Figure 15.9 – Code for solving the “Tower of Hanoi” problem

Below is the result of the program for solving the Tower of Hanoi problem with three and four disks (figure 15.10)

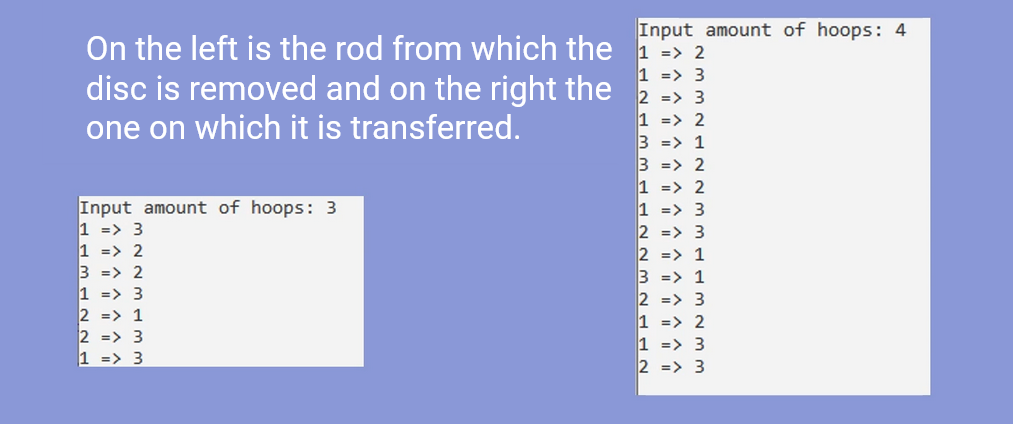


Figure 15.10 – The result of the program – the sequence of movements of the disks to the desired rod