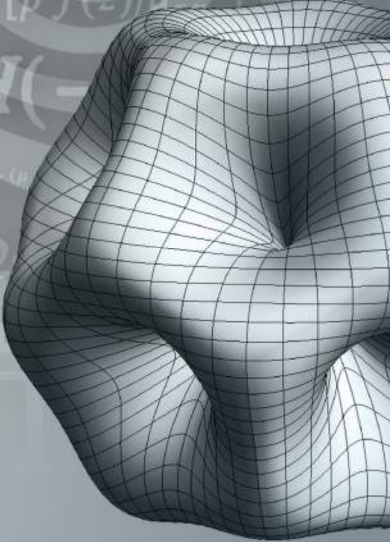
ISSN 2182-1976



Recreational Mathematics Magazine

A Recursive Solution to Bicolor Towers of Hanoi Problem

Prasad Vithal Chaugule
Former student, Dr. Babasaheb Ambedkar Technological University, India
M. Tech student, VJTI, Mumbai, India
prasadvchaugule@gmail.com

Number 4
September, 2015

Ludus

CIUHCT

A Recursive Solution to Bicolor Towers of Hanoi Problem

Prasad Vithal Chaugule
Former student, Dr. Babasaheb Ambedkar Technological University, India
M. Tech student, VJTI, Mumbai, India
prasadvchaugule@gmail.com

Dedicated to my father to whom it would never reach.

Abstract: *In this paper, we present a recursive algorithm to solve bicolor towers of Hanoi problem.*

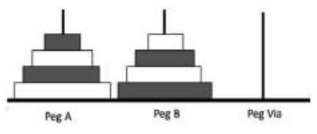
Key-words: Algorithms, recursion, towers of Hanoi.

1 Introduction

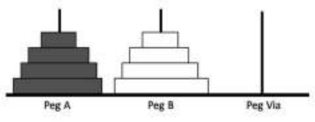
The bicolor towers of Hanoi problem is a variation of traditional towers of Hanoi [1] problem. It was offered to grade 3-6 students at *2ème Championnat de France des Jeux Mathématiques et Logiques* held in July 1988 [2]. Suppose there are three pegs, A, B and Via. Suppose there are two sets of disks α and β, where $\alpha = \{ \alpha_i \mid 1 \leqslant \mathrm{i} \leqslant \mathrm{n} \}$ and $\beta = \{ \beta_i \mid 1 \leqslant \mathrm{i} \leqslant \mathrm{n} \}$ such that color of every disk in α is white and color of every disk in β is black. Also, for every i, radius of α_i is i and radius of β_i is i. Suppose there are placed a finite number of n disks of alternating colors on pegs A and B with decreasing size from bottom to top. Peg Via is an auxillary peg. The goal of the problem is to make the towers of pegs A and B monochrome. The biggest disks at the bottom of the pegs A and B are required to swap positions. The rules of the problem are the following:

1. Only one disk can be moved at any time.
2. At no time can a larger disk be placed on a smaller disk. Same size disks can be placed over one another.

Figures 1a and 1b show the initial and final configuration of bicolor towers of Hanoi problem for $n = 4$. Out of the bicolor towers of Hanoi problem, we derive a new problem, which is same as the bicolor towers of Hanoi problem except that in this variation, we compel the output configuration to maintain the base disks of pegs A and B in their original place as was in its original configuration. We name this problem as easy bicolor towers of Hanoi problem. Figures 2a and 2b show the initial and final configuration of easy bicolor towers of Hanoi problem for $n = 4$. Hereafter, we refer the bicolor towers of Hanoi problem as the bicolor problem and the easy bicolor towers of Hanoi problem as the easy bicolor problem.

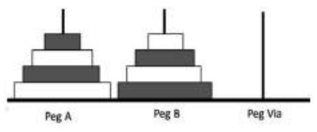


(a)

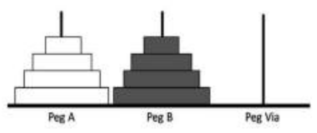


(b)

Figure 1: (a) Initial configuration of bicolor towers of Hanoi problem ($n = 4$). (b) Final configuration of bicolor towers of Hanoi problem (n=4).



(a)



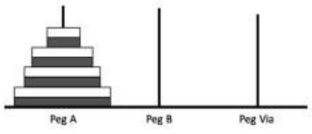
(b)

Figure 2: (a) Initial configuration of easy bicolor towers of Hanoi problem (n=4). (b) Final configuration of easy bicolor towers of Hanoi problem (n=4).

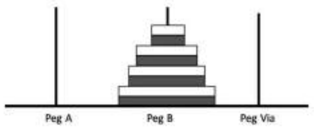
2 Double Towers of Hanoi Problem

We will study another variation to traditional towers of Hanoi problem which is double towers of Hanoi problem. Suppose there are three pegs, A, B and Via. Suppose there are two sets of disks α and β, where $\alpha = \{ \alpha_i \mid 1 \leqslant \mathrm{i} \leqslant \mathrm{n} \}$ and $\beta = \{ \beta_i \mid 1 \leqslant \mathrm{i} \leqslant \mathrm{n} \}$ such that color of every disk in α is white and color of every disk in β is black. Also, for every i, radius of α_i is i and radius of β_i is i. Suppose there are placed a finite number of $2n$ disks on peg A with non-increasing size from bottom to top such that for every same size pair, the black disk is always placed below the white disk. Peg Via is an auxillary peg. The goal of the problem is to move all the disks from peg A to peg B. In the output configuration on peg B, for every same size disk pair, the black disk is supposed to get placed below the white disk. The rules of the problem remains the same. Figures 3a and 3b show the initial and final configuration of double towers of Hanoi problem for n=4. Algorithm 1 solves the double towers of Hanoi problem with a small fault in the output configuration. The fault in the output configuration is that the position of the bottommost same size disk pair gets swapped, that is, the white disk get placed below the black disk as shown in Figures 4a and 4b.

To overcome this fault, we present enhanced double towers of Hanoi algorithm which calls the double towers of Hanoi routine twice to overcome this fault. The correctness of algorithm 1 (with the fault) is straightforward whereas the correctness of algorithm 2 is shown in Figures 5a and 5b.

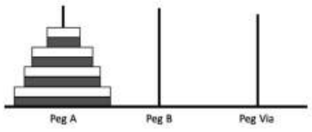


(a)

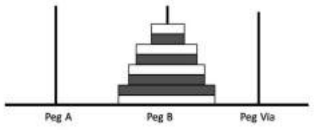


(b)

Figure 3: (a) Initial configuration of double towers of Hanoi problem (n=4). (b) Final configuration of double towers of Hanoi problem (n=4).



(a)



(b)

Figure 4: (a) Initial configuration of double towers of Hanoi problem (n=4). (b) Final configuration of double towers of Hanoi problem with a fault due to algorithm 1 (n=4).

Algorithm 1 Algorithm for Double Towers of Hanoi Problem

```
1: procedure DoubleTowersOfHanoi($A$, $B$, $Via$, $n$)
2:     if $n == 1$ then
3:         print: Move from A to B
4:         print: Move from A to B
5:     else
6:         DoubleTowersOfHanoi($A$, $Via$, $B$, $n - 1$)
7:         print: Move from A to B
8:         print: Move from A to B
9:         DoubleTowersOfHanoi($Via$, $B$, $A$, $n - 1$)
10:    end if
11: end procedure
```

The recurrence relation of algorithm 1 is

$$A_1(n) = \begin{cases} 2 & n = 1 \\ 2.A_1(n-1) + 2 & n > 1 \end{cases}$$

Algorithm 2 Enhanced Algorithm for Double Towers of Hanoi Problem

```
1: procedure EnhancedDoubleTowersOfHanoi($A$, $B$, $Via$, $n$)
2:     DoubleTowersOfHanoi($A$, $Via$, $B$, $n$)          ▷ step 1
3:     DoubleTowersOfHanoi($Via$, $B$, $A$, $n$)          ▷ step 2
4: end procedure
```

The recurrence relation of algorithm 2 is

$$\boxed{A_2(n) = \; 2.A_1(n) \qquad n \geqslant 1}$$

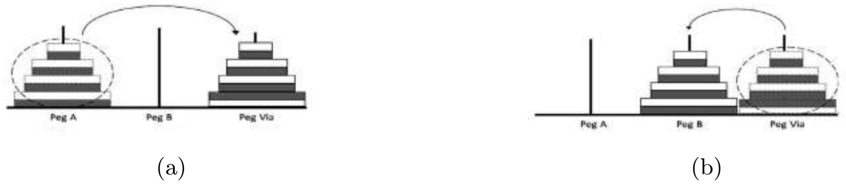
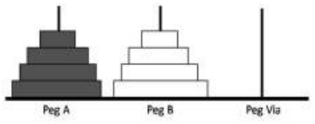


Figure 5: (a) Configuration after step 1 of algorithm 2 (n=4). (b) Configuration after step 2 of algorithm 2 (n=4).

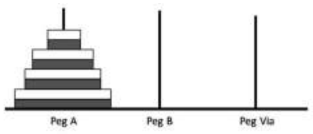
3 The Merge Problem

In this section, we study the merge problem. Suppose there are three pegs, A, B and Via. Suppose there are two sets of disks α and β, where $\alpha = \{ \alpha_i \mid 1 \leqslant \mathrm{i} \leqslant \mathrm{n} \}$ and $\beta = \{ \beta_i \mid 1 \leqslant \mathrm{i} \leqslant \mathrm{n} \}$ such that color of every disk in α is white and color of every disk in β is black. Also, for every i, radius of α_i is i and radius of β_i is i. Suppose there are placed a finite number of n disks on peg A from set β with decreasing size from bottom to top and there are placed a finite number of n disks on peg B from set α with decreasing size from bottom to top. Peg Via is an auxillary peg. The goal of the problem is to place all the *2n* disks on peg A with non-increasing size from bottom to top such that for every same size pair, the black disk is always placed below the white disk. The rules of the problem remains the same.

Figures 6a and 6b show the initial and final configuration of the merge problem for n=4. Algorithm 3 solves the merge problem.



(a)



(b)

Figure 6: (a) Initial configuration of the merge problem (n=4). (b) Final configuration of the merge problem (n=4).

Algorithm 3 Algorithm for Merge Problem

```
procedure MergeProblem($A$, $B$, $Via$, $n$)
    if $n == 1$ then
        print: Move from B to A
    else
        MergeProblem($A$, $B$, $Via$, $n-1$)                ▷ step 1
        DoubleTowersOfHanoi($A$, $Via$, $B$, $n-1$)         ▷ step 2
        print: Move from B to A                             ▷ step 3
        DoubleTowersOfHanoi($Via$, $A$, $B$, $n-1$)         ▷ step 4
    end if
end procedure
```

The recurrence relation of algorithm 3 is

$$A_3(n) = \begin{cases} 1 & n = 1 \\ A_3(n-1) + 2.A_1(n-1) + 1 & n > 1 \end{cases}$$

The correctness of algorithm 3 is shown in Figures 7a, 7b, 7c and 7d.

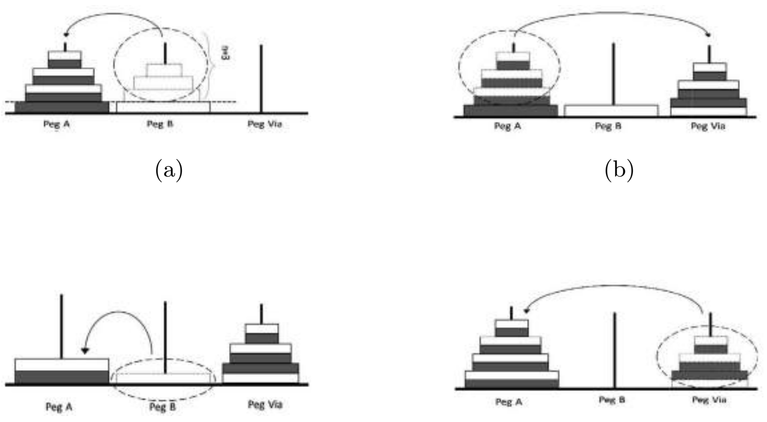


Figure 7: (a) Configuration after step 1 of algorithm 3 (n=4). (b) Configuration after step 2 of algorithm 3 (n=4). (c) Configuration after step 3 of algorithm 3 (n=4). (d) Configuration after step 4 of algorithm 4 (n=4).

4 The Split Problem

In this section, we study the split problem. Suppose there are three pegs, A, B and Via. Suppose there are two sets of disks α and β, where $\alpha = \{ \alpha_i \mid 1 \leqslant \mathrm{i} \leqslant \mathrm{n} \}$ and $\beta = \{ \beta_i \mid 1 \leqslant \mathrm{i} \leqslant \mathrm{n} \}$ such that color of every disk in α is white and color of every disk in β is black. Also, for every i, radius of α_i is i and radius of β_i is i. Suppose there are placed a finite number of $2n$ disks on peg A with non-increasing size from bottom to top such that for every same size pair, the black disk is always placed below the white disk. Peg Via is an auxillary peg. The goal of the problem is to place n disks on peg A from set β with decreasing size from bottom to top and n disks on peg B from set α with decreasing size from bottom to top. The rules of the problem remains the same. Figures 8a and 8b show the initial and final configuration of the split problem for n=4. Algorithm 4 solves the split problem. The correctness of algorithm 4 is shown in Figures 9a, 9b, 9c and 9d.

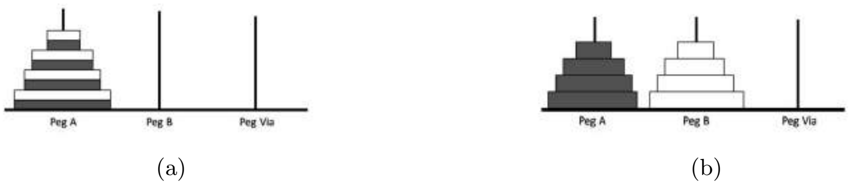


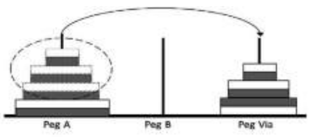
Figure 8: (a) Initial configuration of the split problem (n=4). (b) Final configuration of the split problem (n=4).

Algorithm 4 Algorithm for Split Problem

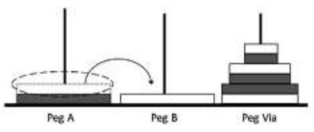
```
1: procedure SplitProblem($A$, $B$, $Via$, $n$)
2:     if $n == 1$ then
3:         print: Move from A to B
4:     else
5:         DoubleTowersOfHanoi($A$, $Via$, $B$, $n-1$)          ▷ step 1
6:         print: Move from A to B                              ▷ step 2
7:         DoubleTowersOfHanoi($Via$, $A$, $B$, $n-1$)          ▷ step 3
8:         SplitProblem($A$, $B$, $Via$, $n-1$)                 ▷ step 4
9:     end if
10: end procedure
```

The recurrence relation of algorithm 4 is

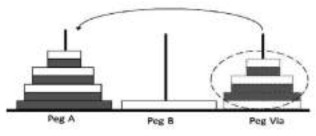
$$\boxed{A_4(n) = \begin{cases} 1 & n = 1 \\ A_4(n-1) + 2.A_1(n-1) + 1 & n > 1 \end{cases}}$$



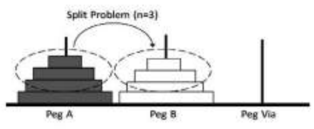
(a)



(b)



(c)



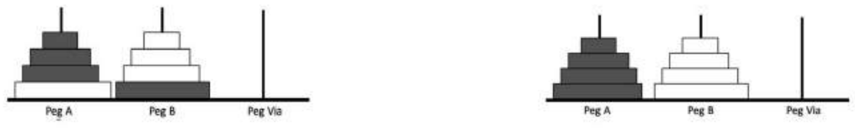
(d)

Figure 9: (a) Configuration after step 1 of algorithm 4 (n=4). (b) Configuration after step 2 of algorithm 4 (n=4). (c) Configuration after step 3 of algorithm 4 (n=4). (d) Configuration after step 4 of algorithm 4 (n=4).

5 The Swap Base Disk Problem

In this section, we study the swap base disk problem. Suppose there are three pegs, A, B and Via. Suppose there are two sets of disks α and β, where $\alpha = \{ \alpha_i \mid 1 \leqslant \text{i} \leqslant \text{n} \}$ and $\beta = \{ \beta_i \mid 1 \leqslant \text{i} \leqslant \text{n} \}$ such that color of every disk in α is white and color of every disk in β is black. Also, for every i, radius of α_i is i and radius of β_i is i. Suppose there are placed a finite number of n disks on peg

A from set $\beta^* = \{ \beta_i \mid 1 \leqslant \text{i} \leqslant \text{n-1} \} \cup \{ \alpha_n \}$ with decreasing size from bottom to top and there are placed a finite number of n disks on peg B from set $\alpha^* = \{ \alpha_i \mid 1 \leqslant \text{i} \leqslant \text{n-1} \} \cup \{ \beta_n \}$ with decreasing size from bottom to top. Peg Via is an auxillary peg. The goal of the problem is to make the towers on peg A and peg B monochrome by swapping the base disks of towers on peg A and peg B. The rules of the problem remains the same. Figures 10a and 10b show the initial and final configuration of the swap base disk problem. Algorithm 5 solves the swap base disk problem. The correctness of algorithm 5 is shown in Figures 11a, 11b, 11c, 11d,11e,11f, 11g and 11h.



(a) (b)

Figure 10: (a) Initial configuration of swap base disk problem (n=4). (b) Final configuration of swap base disk problem (n=4).

Algorithm 5 Algorithm for Swap Base Disk Problem

```
 1: procedure SwapBaseDiskProblem($A$, $B$, $Via$, $n$)
 2:     if $n == 1$ then
 3:         print: Move from A to Via
 4:         print: Move from B to A
 5:         print: Move from Via to B
 6:     else
 7:         MergeProblem($A$, $B$, $Via$, $n-1$)                        ▷ step 1
 8:         print: Move from B to Via                                   ▷ step 2
 9:         DoubleTowersOfHanoi($A$, $Via$, $B$, $n-1$)                 ▷ step 3
10:         print: Move from A to B                                     ▷ step 4
11:         DoubleTowersOfHanoi($Via$, $B$, $A$, $n-1$)                 ▷ step 5
12:         print: Move from Via to A                                   ▷ step 6
13:         EnhancedDoubleTowersOfHanoi($B$, $A$, $Via$, $n-1$)         ▷ step 7
14:         SplitProblem($A$, $B$, $Via$, $n-1$)                        ▷ step 8
15:     end if
16: end procedure
```

The recurrence relation of algorithm 5 is

$$A_5(n) = \begin{cases} 3 & n = 1 \\ A_3(n-1) + A_4(n-1) + A_2(n-1) + 2.A_1(n-1) + 3 & n > 1 \end{cases}$$

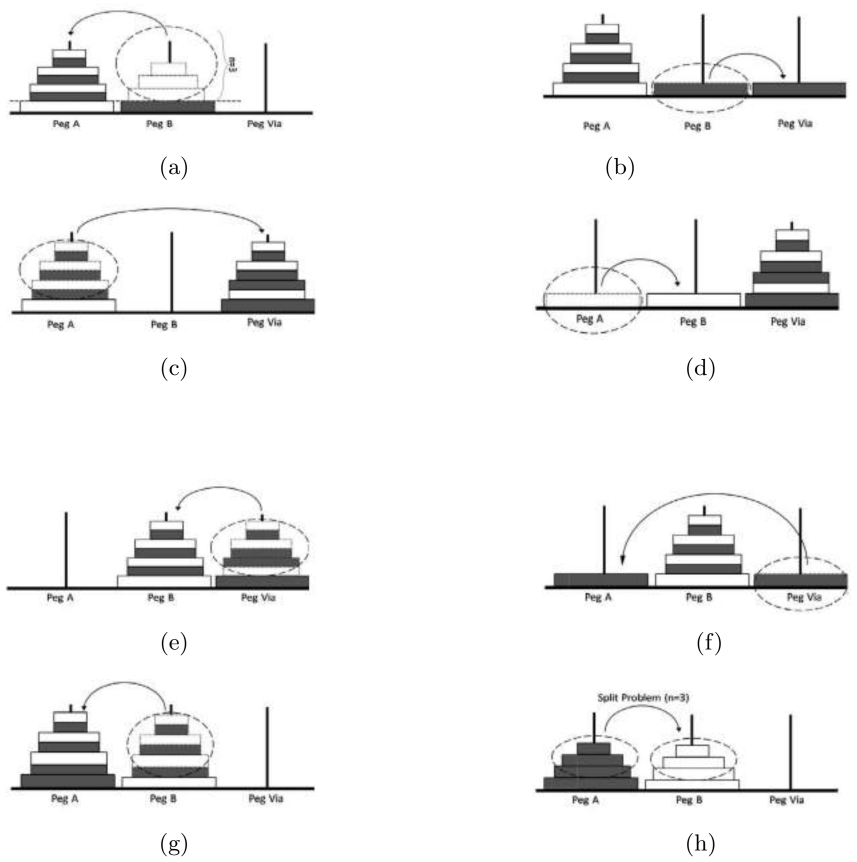


Figure 11: (a) Configuration after step 1 of algorithm 5 (n=4). (b) Configuration after step 2 of algorithm 5 (n=4). (c) Configuration after step 3 of algorithm 5 (n=4). (d) Configuration after step 4 of algorithm 5 (n=4). (e) Configuration after step 5 of algorithm 5 (n=4). (f) Configuration after step 6 of algorithm 5 (n=4). (g) Configuration after step 7 of algorithm 5 (n=4). (h) Configuration after step 8 of algorithm 5 (n=4).

6 The Easy Bicolor Towers of Hanoi

In this section, we present the algorithm for solving the easy bicolor problem. Algorithm 6 solves the easy bicolor problem. The correctness of algorithm 6 is shown in Figure 12.

Algorithm 6 Algorithm for Easy Bicolor Problem

```
1: procedure EasyBicolorProblem($A$, $B$, $Via$, $n$)
2:     if $n == 1$ then
   ▷ do nothing
3:     else
4:         BicolorProblem($A$, $B$, $Via$, $n-1$)                ▷ step 1
5:     end if
6: end procedure
```

The recurrence relation of algorithm 6 is

$$A_6(n) = \begin{cases} 0 & n = 1 \\ A_7(n-1) & n > 1 \end{cases}$$

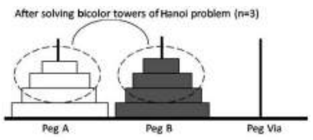


Figure 12: Configuration after step 1 of algorithm 6 (n=4).

7 The Bicolor Towers of Hanoi

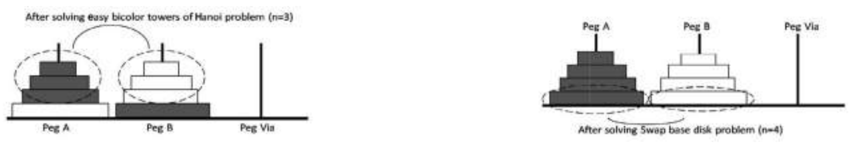
In this section, we present the algorithm for solving the bicolor problem. Algorithm 7 solves the bicolor problem. The correctness of algorithm 7 is shown in Figures 13a and 13b.

Algorithm 7 Algorithm for Bicolor Problem

```
procedure BicolorProblem($A$, $B$, $Via$, $n$)
    if $n == 1$ then
        print: Move from A to Via
        print: Move from B to A
        print: Move from Via to B
    else
        EasyBicolorProblem($A$, $B$, $Via$, $n-1$)        ▷ step 1
        SwapBaseDiskProblem($A$, $B$, $Via$, $n$)        ▷ step 2
    end if
end procedure
```

The recurrence relation of algorithm 7 is

$$\boxed{A_7(n) = \begin{cases} 3 & n = 1 \\ A_6(n-1) + A_5(n) & n > 1 \end{cases}}$$


After solving easy bicolor towers of Hanoi problem (n=3)
Peg A
Peg B
Peg Via
Peg A
Peg B
Peg Via
After solving Swap base disk problem (n=4)

(a) (b)

Figure 13: (a) Configuration after step 1 of algorithm 7 (n=4). (b) Configuration after step 2 of algorithm 7 (n=4).

8 Conclusion

We have studied the recursive solution to bicolor towers of Hanoi problem for size n.

Appendix A

A Recursive Solution to the Traditional Towers of Hanoi Problem

Algorithm 8 Algorithm for traditional Towers of Hanoi Problem

```
procedure TowersOfHanoi($A$, $B$, $Via$, $n$)
    if $n == 1$ then
        print: Move from A to B
    else
        TowersOfHanoi($A$, $Via$, $B$, $n - 1$)
        print: Move from A to B
        TowersOfHanoi($Via$, $B$, $A$, $n - 1$)
    end if
end procedure
```

The recurrence relation of algorithm 8 is

$$\boxed{A_8(n) = \begin{cases} 1 & n = 1 \\ 2.A_8(n-1) + 1 & n > 1 \end{cases}}$$

References

[1] Rosen, K., H. *Discrete Mathematics & and Its Applications: With Combinatorics and Graph Theory*, McGraw-Hill Offices, 2010.

[2] `http://www.cut-the-knot.org/recurrence/BiColorHanoi.shtml`