# Assignment 4



**This assignment is to be done individually.**

---

**Important Note:** The university policy on academic dishonesty (cheating) will be taken very seriously in this course. You may not provide or use any solution, in whole or in part, to or by another student.

You are encouraged to discuss the concepts involved in the questions with other students. If you are in doubt as to what constitutes acceptable discussion, please ask! Further, please take advantage of office hours offered by the instructor and the TA if you are having difficulties with this assignment.

**DO NOT**:

- Give/receive code or proofs to/from other students
- Use search engines to find solutions for the assignment

**DO**:

- Meet with other students to discuss assignment (it is best not to take any notes during such meetings, and to re-work assignment on your own)
- Use online resources (e.g. Wikipedia) to understand the concepts needed to solve the assignment

**Submission Instructions:**

- You may type or write your answer as long as it is readable.
- Submit two files on CourSys
  1. `report.pdf`, which contains a write-up of your solutions to the assignment
  2. `Hanoi.c`, which contains the code you wrote in Problem 2.

---

# Assignment 4

## Problem 1

Prove, using mathematical induction, that the `SeriesSum` function in the code below outputs $\frac{n^2(n+1)^2}{4}$ for any positive integer $n$.

```
1  #include <stdio.h>
2
3  int SeriesSum(int n){
4      if(n == 1)
5          return 1;
6      else
7          return (n*n*n) + SeriesSum(n-1);
8  }
9
10 int main(){
11   int n;
12   scanf("%d",&n);
13   printf("%d",SeriesSum(n));
14 }
```

## Problem 2

You are given 3 towers/pegs/poles and $n$ disks of different sizes. All disks are initially on pole $A$ in the ascending order (smallest one on top). Write code to print all the single disk moves required to move all the disks from pole $A$ to pole $C$ while satisfying the following rules (15 marks):

- Only one disk can be moved at a time.
- A larger disk cannot be placed on a smaller disk at any time.
- Each move involves taking a disk at the top of any pole and placing it at another pole.

**input:** `n`

**output:** Movements (one per line)
[source pole] to [destination pole]

### sample:

input: `3`

output:

# Assignment 4

```
A to C
A to B
C to B
A to C
B to A
B to C
A to C
```

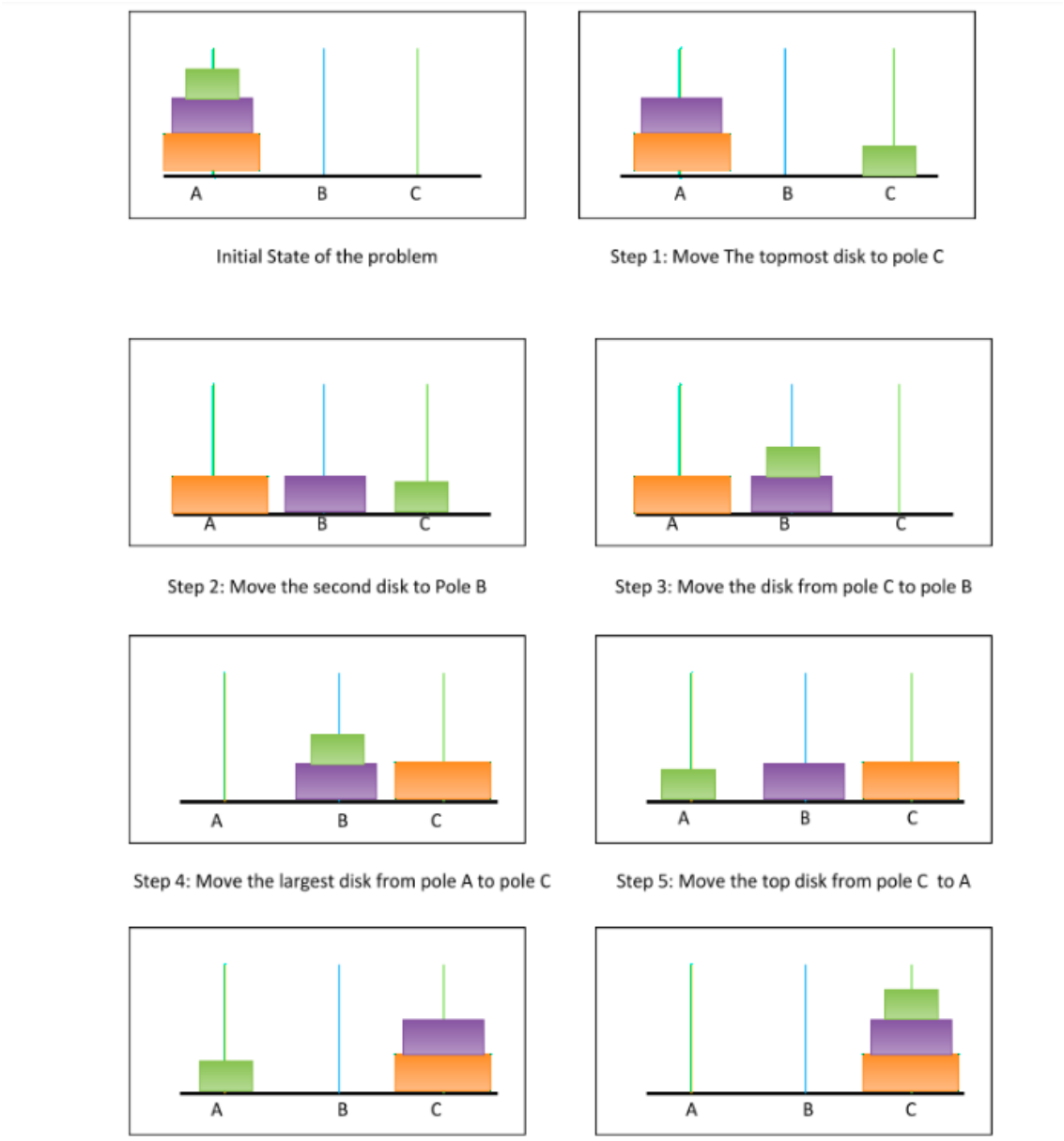


Figure 1: Solve the Tower of Hanoi problem with 3 disk