CS1010 http://www.comp.nus.edu.sg/~cs1010/

Programming Methodology

UNIT 17

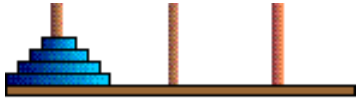
Recursion: Towers of Hanoi

Unit 17: Recursion: Towers of Hanoi

Objectives:

- Understand that many problems are more naturally solved with recursion, which can provide elegant solutions.
- Taste the classic example of recursion: Towers of Hanoi.

Tower Of Hanoi (1/17)



- This classical “Towers of Hanoi” puzzle has attracted the attention of computer scientists more than any other puzzles.
- Invented by Edouard Lucas, a French mathematician, in1883.
- There are 3 pegs (A, B and C) and a tower of n disks on the first peg A, with the smallest disk on the top and the biggest at the bottom. The purpose of the puzzle is to move the whole tower from peg A to peg C, with the following simple rules:
  - Only one disk (the one at the top) can be moved at a time.
  - A bigger disk must not rest on a smaller disk.

Tower Of Hanoi (2/17)

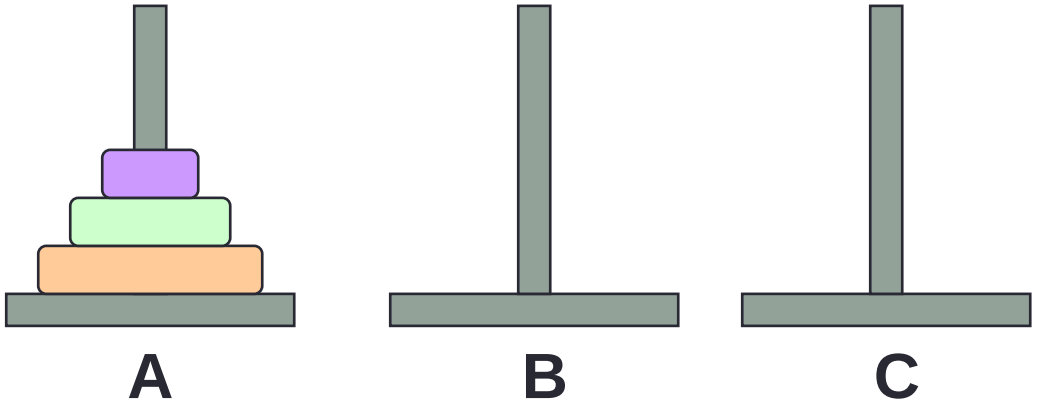
- Demo: Tower of Hanoi
- We attempt to write a program to produce instructions on how to move the disks from peg A to peg C to complete the puzzle.
- Example: A tower with 3 disks.
- Output produced by program:

```
Move disk from A to C
Move disk from A to B
Move disk from C to B
Move disk from A to C
Move disk from B to A
Move disk from B to C
Move disk from A to C
```

Tower Of Hanoi (3/17)

- Example: A tower with 3 disks.

  Move disk from A to C
  Move disk from A to B
  Move disk from C to B
  Move disk from A to C
  Move disk from B to A
  Move disk from B to C
  Move disk from A to C



Tower Of Hanoi (4/17)

- Example: A tower with 3 disks.

  Move disk from A to C

  Move disk from A to B

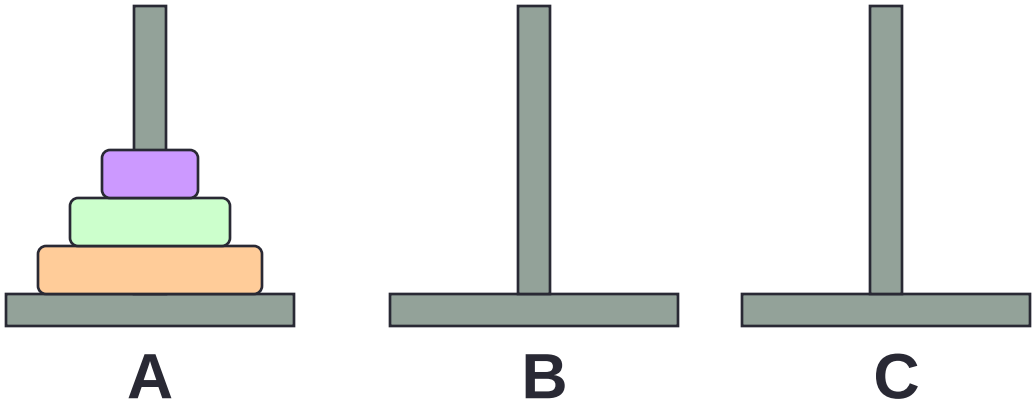
  Move disk from C to B

  Move disk from A to C

  Move disk from B to A

  Move disk from B to C

  Move disk from A to C



Tower Of Hanoi (5/17)

- Example: A tower with 3 disks.

  Move disk from A to C

  Move disk from A to B

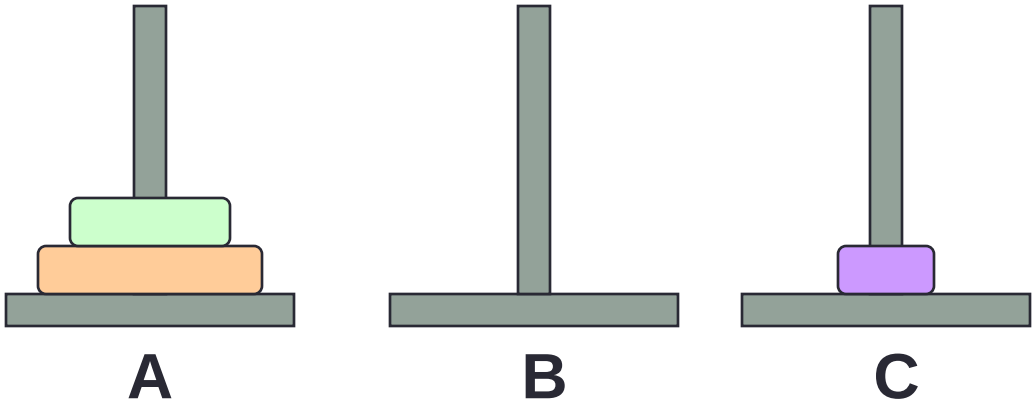
  Move disk from C to B

  Move disk from A to C

  Move disk from B to A

  Move disk from B to C

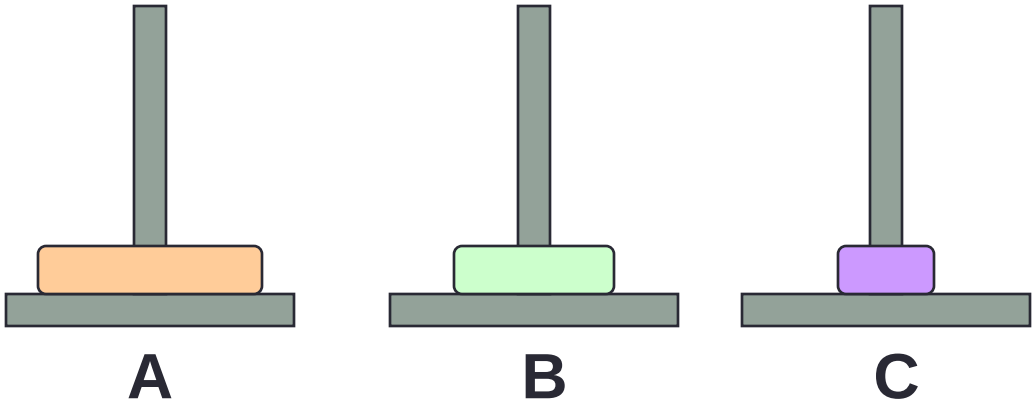
  Move disk from A to C



Tower Of Hanoi (6/17)

- Example: A tower with 3 disks.

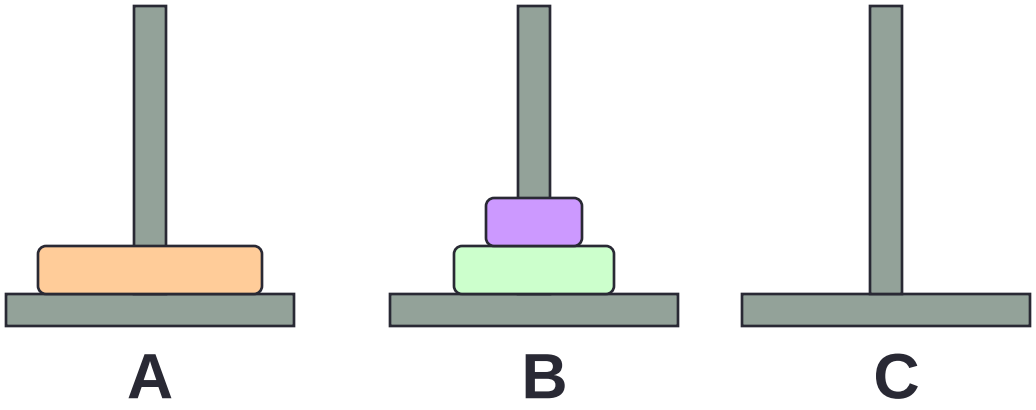
  Move disk from A to C
  Move disk from A to B
  Move disk from C to B
  Move disk from A to C
  Move disk from B to A
  Move disk from B to C
  Move disk from A to C



Tower Of Hanoi (7/17)

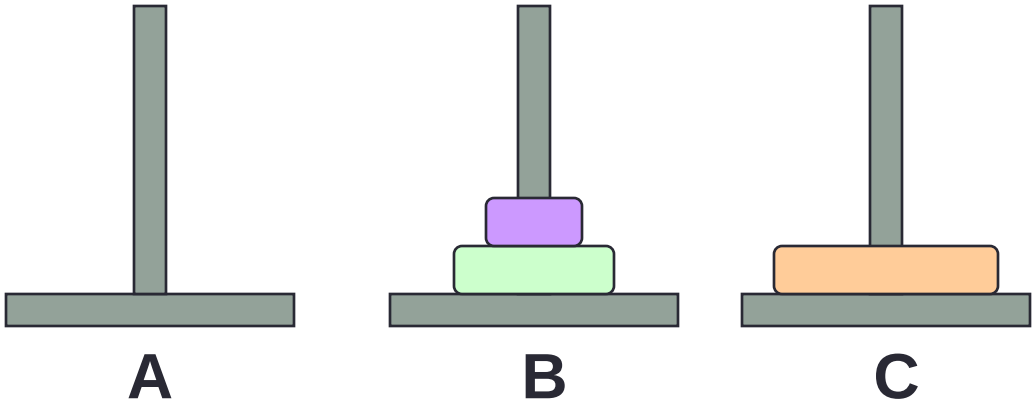
- Example: A tower with 3 disks.

  Move disk from A to C
  Move disk from A to B
  Move disk from C to B
  Move disk from A to C
  Move disk from B to A
  Move disk from B to C
  Move disk from A to C



Tower Of Hanoi (8/17)

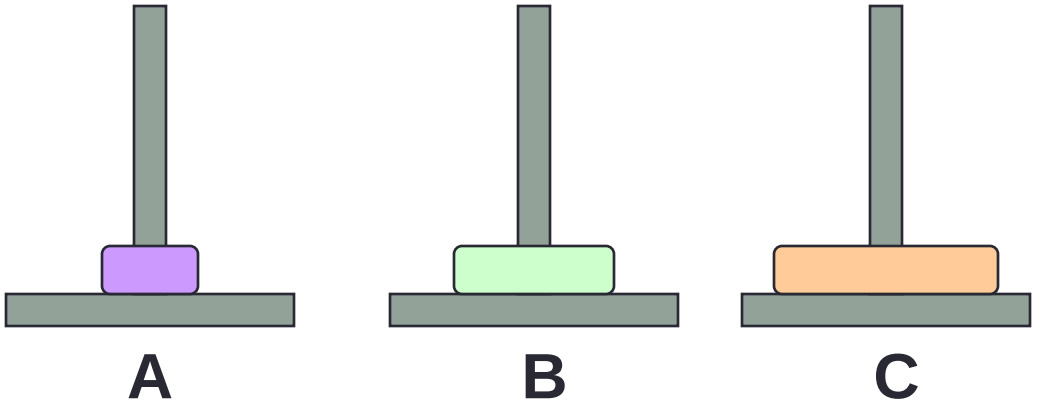
- Example: A tower with 3 disks.
  - Move disk from A to C
  - Move disk from A to B
  - Move disk from C to B
  - Move disk from A to C
  - **Move disk from B to A**
  - Move disk from B to C
  - Move disk from A to C



Tower Of Hanoi (9/17)

- Example: A tower with 3 disks.

  Move disk from A to C
  Move disk from A to B
  Move disk from C to B
  Move disk from A to C
  Move disk from B to A
  Move disk from B to C
  Move disk from A to C

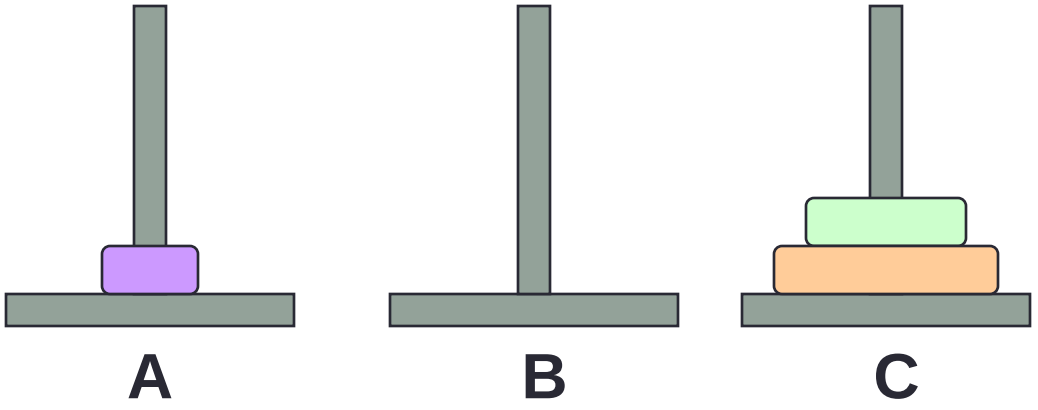


Tower Of Hanoi (10/17)

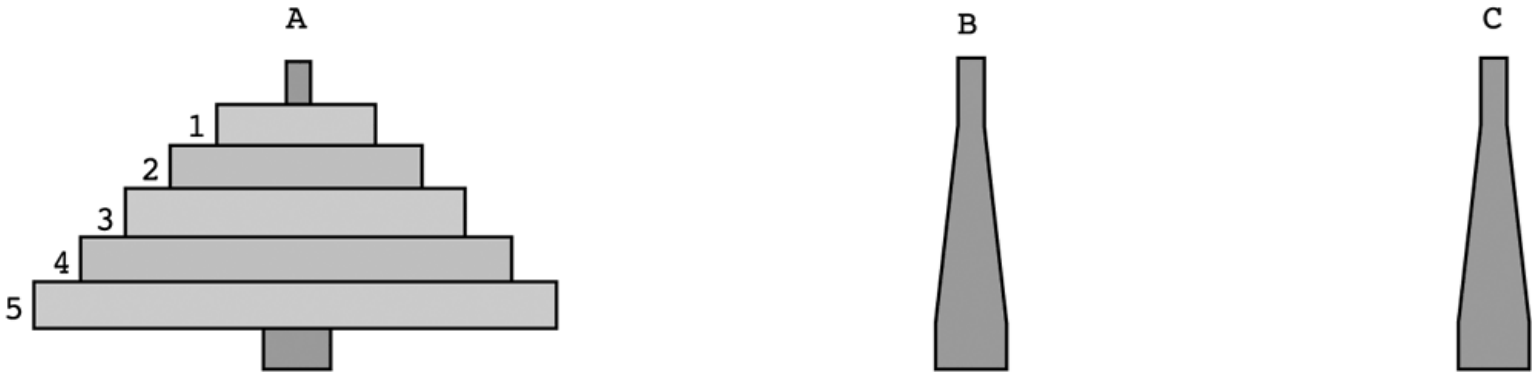
- Example: A tower with 3 disks.

  Move disk from A to C
  Move disk from A to B
  Move disk from C to B
  Move disk from A to C
  Move disk from B to A
  Move disk from B to C
  Move disk from A to C

Voilà!



Tower Of Hanoi (11/17)

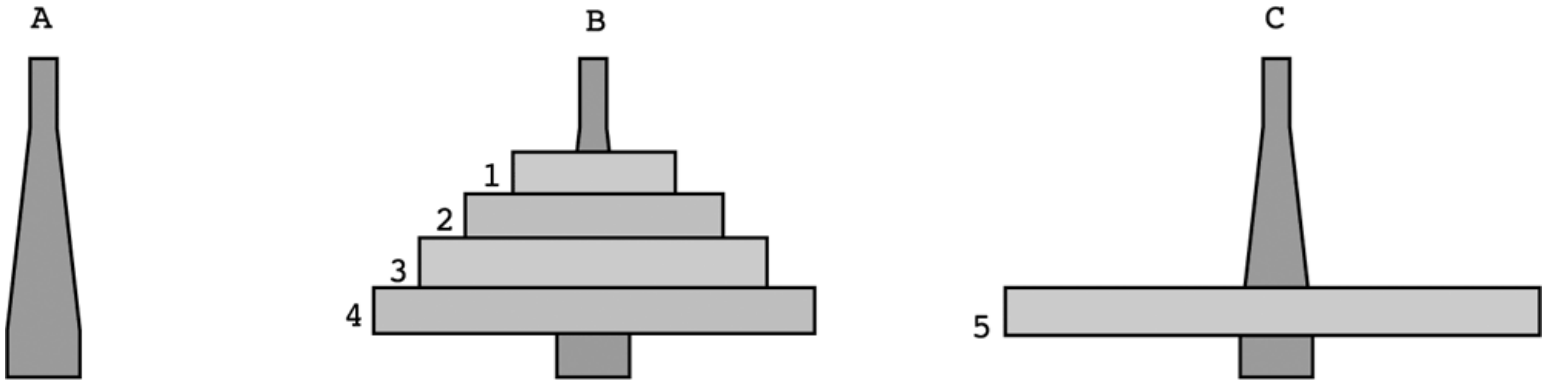


Can be interpreted as:

1. move four disks from peg A to peg B
2. move disk 5 from peg A to peg C
3. move four disks from peg B to peg C

Tower Of Hanoi (12/17)

Towers after steps 1 and 2:



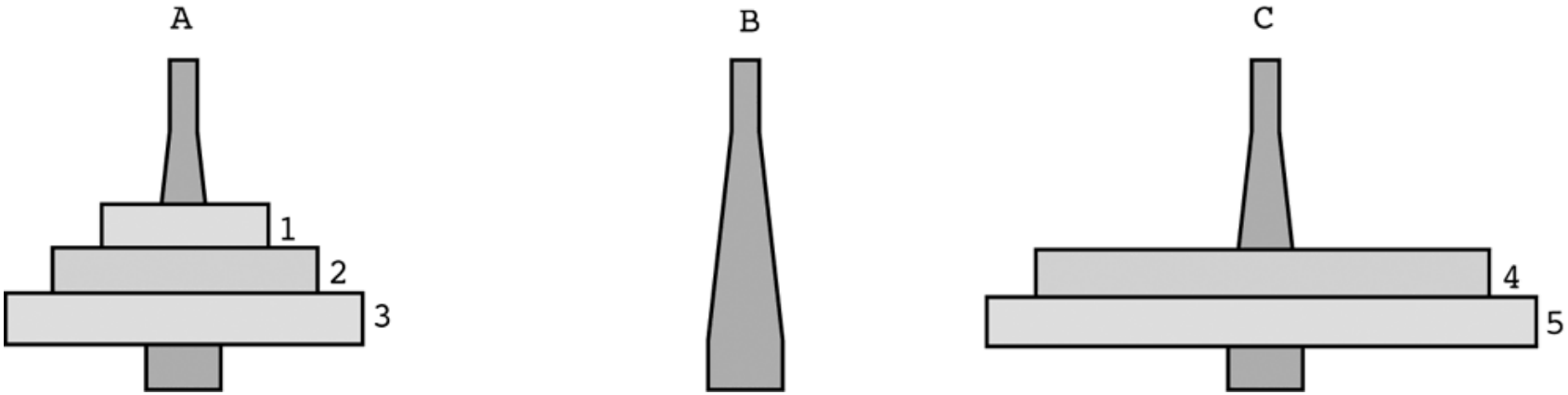
But how do we execute step 1? Or step 3?

Step 3 can be interpreted as:

- 3.1 move three disks from peg B to peg A
- 3.2 move disk 4 from peg B to peg C
- 3.3 move three disks from peg A to peg C

Tower Of Hanoi (13/17)

Towers after steps 1, 2, 3.1 and 3.2:



Can you start to visualise how to solve this using recursion?

Tower Of Hanoi (14/17)

- Algorithm:

```
if (n > 0)
    move n – 1 disks from the source peg to the
    temp peg using the dest peg

    move disk n from the source peg to the dest peg

    move n – 1 disks from the temp peg to the dest
    peg using the source peg
```

Tower Of Hanoi (15/17)

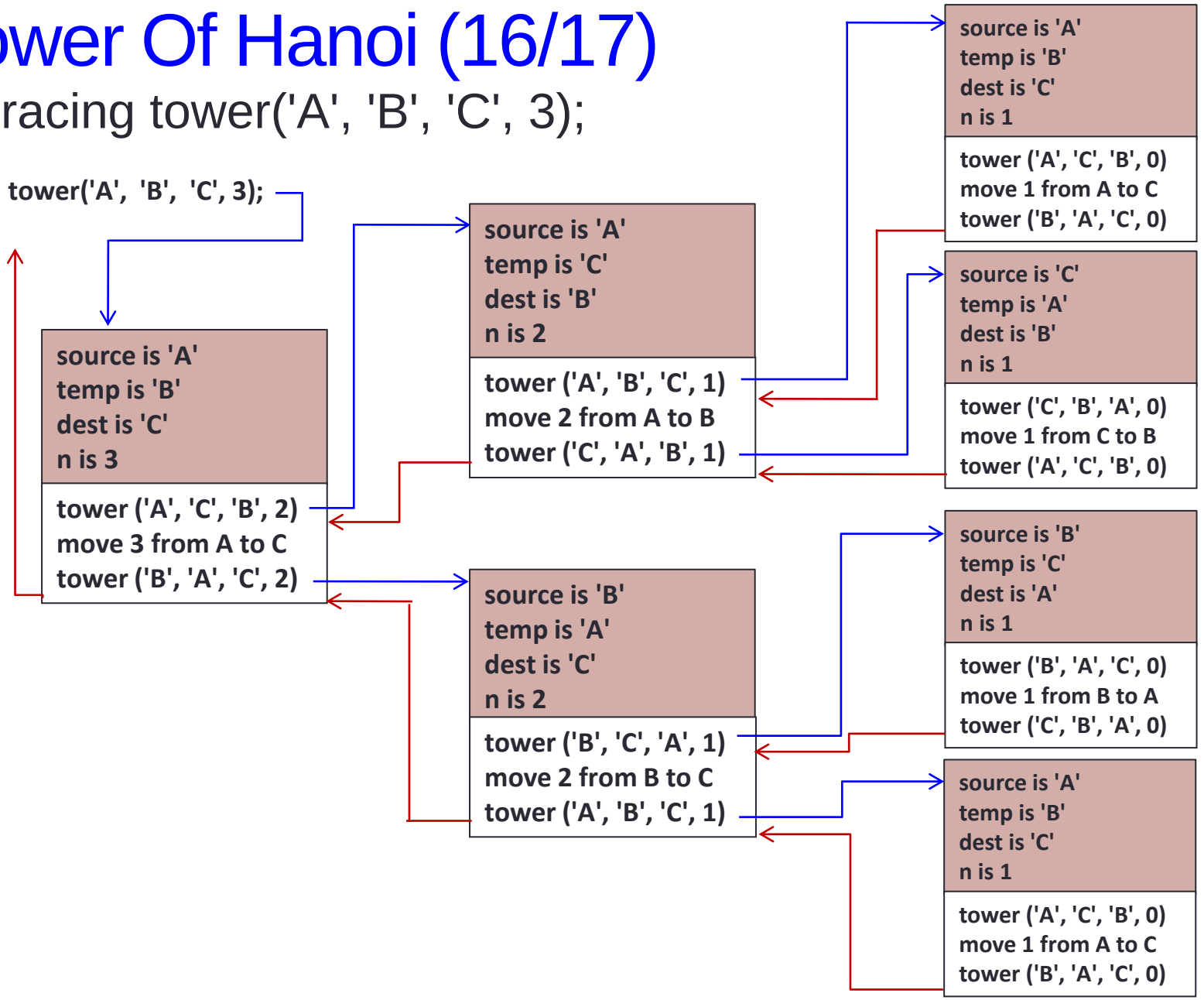
Unit17_TowersOfHanoi.c

```
#include <stdio.h>
void tower(char, char, char, int);
int main(void) {
   int disks;
   printf("Number of disks: ");
   scanf("%d", &disks);
   tower('A','B','C', disks);
   return 0;
}

// Display instructions for moving n disk from source to dest
// using temp as an auxiliary. Disks are numbered 1 to n
// (smallest to largest).
void tower(char source, char temp, char dest, int n) {
   if (n > 0) {
      tower(source, dest, temp, n-1);
      printf("Move disk %d from peg %c to peg %c\n",
             n, source, dest);
      tower(temp, source, dest, n-1);
   }
}
```

Tower Of Hanoi (16/17)

- Tracing tower('A', 'B', 'C', 3);



Tower Of Hanoi (17/17)

- Output generated by tower('A', 'B', 'C', 3);

```
Move disk 1 from peg A to peg C
Move disk 2 from peg A to peg B
Move disk 1 from peg C to peg B
Move disk 3 from peg A to peg C
Move disk 1 from peg B to peg A
Move disk 2 from peg B to peg C
Move disk 1 from peg A to peg C
```

End of File