

# Exercise 9: Recursion

S Milton Rajendram

16-29 March 2018 (Mon-Thurs)

## 1 Text processing

Read a text from stdin. Define functions for doing each of the following operations. Test the functions individually.

1. Store the lines in as an array of lines, each line a C-string.
2. Store each line as an array of words, each word a C-string.
3. Count the number of lines.
4. Count the number of words.
5. Define a function to “search and replace” a word by another word.
6. Capitalize the first letter of each line.

## 2 Tower of Hanoi

There are three poles fixed in the ground. On the first of these poles, 8 discs are placed, each of different size, in decreasing order of size. How will you move the discs from its pole to the clockwise pole (`cw_pole`) according to the rule that no disc may ever be above a smaller disc. Figures 1.

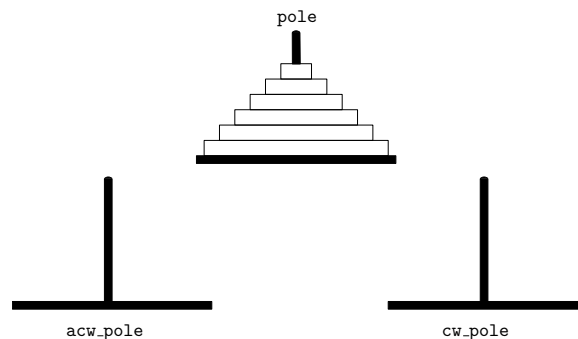


Figure 1: Tower of Hanoi, pole, clockwise pole, anti-clockwise pole

We can solve the problem recursively.

- Base case: There is no disc in the pole.
- Recursion step: Reduce the size of the tower to  $n - 1$  discs. Move the tower of top  $n - 1$  discs to the anti-clockwise pole. Move the exposed disc ( $n$ ) on the pole to the clockwise pole. Then, move the tower of  $n - 1$  discs from anti-clockwise pole to the clockwise pole. This idea is illustrated in Figure 2. Define `hanoi()`. Let the function print the sequence of moves on the `stdout`.

1: 1 -> 2

2: 1 -> 3

...

```
move_tower (n, pole, cw pole, acw pole)
-- pre:  tower of size n on pole,
--       towers in cw and acw poles are broader than the tower on pole
-- post: tower of size n on cw pole
if n > 0
    move_tower (n-1, pole, acw pole, cw pole)
    move_disk (pole, cw pole)
    move_tower (n-1, acw pole, cw pole, pole)
```

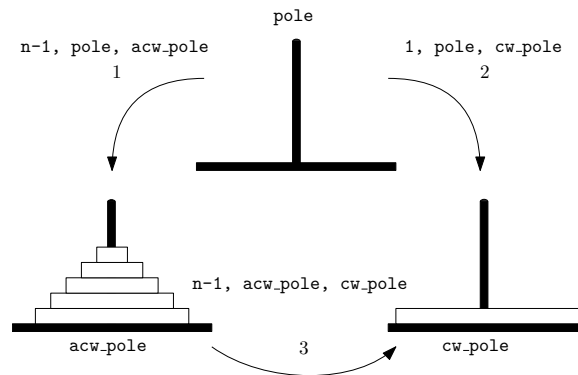


Figure 2: Tower of Hanoi: move tower in two recursive steps