tower_of_hanoi

October 18, 2021

Hoàng Trần Nhật Minh - 20204883

This notebook

[1]:
```
# TOWER OF HANOI: TRADITIONAL RECURSION PROBLEM
# EXPLICITLY VISUALIZED CODE
def hanoi_tower(n):
    '''run and return number of transfers'''
    def transfer(n, start, end, mid):
        if n == 1:
            nonlocal count
            count += 1
            map_ind = ['A', 'B', 'C']
            print(' '*18 + 'Step ' + str(count) + ': ' + map_ind[start] + ' ->␣
 ↪' + map_ind[end])
            move(start, end)
            print_board()
        else:
            transfer(n - 1, start, mid, end)
            transfer(1, start, end, mid)
            transfer(n - 1, mid, end, start)
    count = 0
    transfer(n, 0, 2, 1)
    return count

'''
a =
      0  1  2  3

0     4  3  2  1
1     0  0  0  0
2     0  0  0  0

printed :
                A  B  C

                1  .  .
                2  .  .
```

```
                3  .  .
                4  .  .
'''
def print_board():
    '''print current board'''
    print('|  A  B  C  |')

    for col in range(n - 1, -1, -1):
        print('|  ', end = '')
        for row in range(3):
            character = '.' if a[row][col] == 0 else a[row][col]
            print(str(character) + '  ', end = '')
        print('|')


def move(row_start, row_end):
    '''make a move'''
    def col_lastnonenull(row):
        for col_ind in range(n - 1, -1, -1):
            if a[row][col_ind] != 0:
                return col_ind

    def col_firstnull(row):
        for col_ind in range(n):
            if a[row][col_ind] == 0:
                return col_ind

    a[row_end][col_firstnull(row_end)] =␣
 ↪a[row_start][col_lastnonenull(row_start)]
    a[row_start][col_lastnonenull(row_start)] = 0

# MAIN:

# input n
n = int(input('n = '))

# initialize list of list, n = 4,
'''
a =
      0  1  2  3

0     4  3  2  1
1     0  0  0  0
2     0  0  0  0
'''
a = [[n - i for i in range(n)]]
for i in range(2):
```

```
    a.append([0 for j in range(n)])

# print the initialized board
print()
print_board()

# run, print the board every move and return the result
print('\nNumber of transfers: %i' % hanoi_tower(n))

# n SHOULD BE LESS THAN 5 for short output text
```

```
n = 4

|  A  B  C  |
|  1  .  .  |
|  2  .  .  |
|  3  .  .  |
|  4  .  .  |
                  Step 1: A -> B
|  A  B  C  |
|  .  .  .  |
|  2  .  .  |
|  3  .  .  |
|  4  1  .  |
                  Step 2: A -> C
|  A  B  C  |
|  .  .  .  |
|  .  .  .  |
|  3  .  .  |
|  4  1  2  |
                  Step 3: B -> C
|  A  B  C  |
|  .  .  .  |
|  .  .  .  |
|  3  .  1  |
|  4  .  2  |
                  Step 4: A -> B
|  A  B  C  |
|  .  .  .  |
|  .  .  .  |
|  .  .  1  |
|  4  3  2  |
                  Step 5: C -> A
|  A  B  C  |
|  .  .  .  |
|  .  .  .  |
|  1  .  .  |
|  4  3  2  |
```

```
                    Step 6: C -> B
|  A  B  C  |
|  .  .  .  |
|  .  .  .  |
|  1  2  .  |
|  4  3  .  |
                    Step 7: A -> B
|  A  B  C  |
|  .  .  .  |
|  .  1  .  |
|  .  2  .  |
|  4  3  .  |
                    Step 8: A -> C
|  A  B  C  |
|  .  .  .  |
|  .  1  .  |
|  .  2  .  |
|  .  3  4  |
                    Step 9: B -> C
|  A  B  C  |
|  .  .  .  |
|  .  .  .  |
|  .  2  1  |
|  .  3  4  |
                    Step 10: B -> A
|  A  B  C  |
|  .  .  .  |
|  .  .  .  |
|  .  .  1  |
|  2  3  4  |
                    Step 11: C -> A
|  A  B  C  |
|  .  .  .  |
|  .  .  .  |
|  1  .  .  |
|  2  3  4  |
                    Step 12: B -> C
|  A  B  C  |
|  .  .  .  |
|  .  .  .  |
|  1  .  3  |
|  2  .  4  |
                    Step 13: A -> B
|  A  B  C  |
|  .  .  .  |
|  .  .  .  |
|  .  .  3  |
|  2  1  4  |
```

```
                    Step 14: A -> C
|  A  B  C  |
|  .  .  .  |
|  .  .  2  |
|  .  .  3  |
|  .  1  4  |
                    Step 15: B -> C
|  A  B  C  |
|  .  .  1  |
|  .  .  2  |
|  .  .  3  |
|  .  .  4  |

Number of transfers: 15
```