

Jaden Liu

Zliu259@ucsc.edu

10/31/2020

CSE13s Fall 2020

Assignment3: The Tower of Brahma

Design document

The tower of brahma is an old game. There are three pegs, and some disks on one of the pegs from small to big on the bottom. The aim of the game is to transport all the disks to the other peg with an rule: only small can be on the top of the big one.

In this lab, I implemented it in two ways: stack and recursive. I use stack to implement as the real pegs to imitate the number flow. For the recursive one, it is just print out the text without the real data change.

This program's argument is got from command line with three types:

- n x: which means get the number of disk
- s: print out the result by stack method
- r: print out the result by recursive method

Stack.c

```
#include stack.h
```

```
Typedef struct {  
    Name  
    Top  
    Capacity  
    Items  
} Stack
```

This file is to help struct a stack
previous to writing the main
program so it is easier to use in
tower.c

```
Stack * stack_create (int capacity , char name ); (function to create a stack)
```

```
Void stack_delete(stack *s){  
    Free(s.items)  
    Free(s)  
}
```

```
Void stack_push(stack *s, int i){  
    s.items[s.top] += i;  
    s.top += 1;  
}
```

```
Int stack_pop(Stack *s){  
    s.top-=1  
    return s.items[s.top]  
}
```

```
Bool stack_empty{  
    Return s.top=0  
}
```

```
Int stack_peek{  
    Return s.items[s.top]  
}
```

tower.c

```
int main(){
while (c=getopt(etc etc)){
switch (c){
case n:
if optarg==null
size = 5
else
size = optarg

case s:
print_stack

case r:
print_rec(tower)

}
}
}

void tower(int n, stack a, stack b, stack c){
almost the same as the recursion method,
just need to pop and push the value from stack}

void tower(int n, char a, char b, char c)
{
    if (n == 1) move(n, a, b);
    print(move_result)
    else
    {
        tower(n - 1, a, c, b);
        move(n, a, b);
        print(move_result)
        tower(n - 1, c, b, a);
    }
}
```

1. Getting the command from command line
2. Print out the result by the command input