Homework9

作者: 罗文杰

专业: 计算机科学与技术

学号: 3210102456

17.2

No.

Taking the code of LC-3 as an example, each time a recursive function is called using JSR, the program puts the address of the next instruction into R7, followed by the code pressing the contents of R7 onto the stack; when the function returns, its return address is the address previously put on the stack, which is the next address of the caller instruction, such that the address is different for each call.

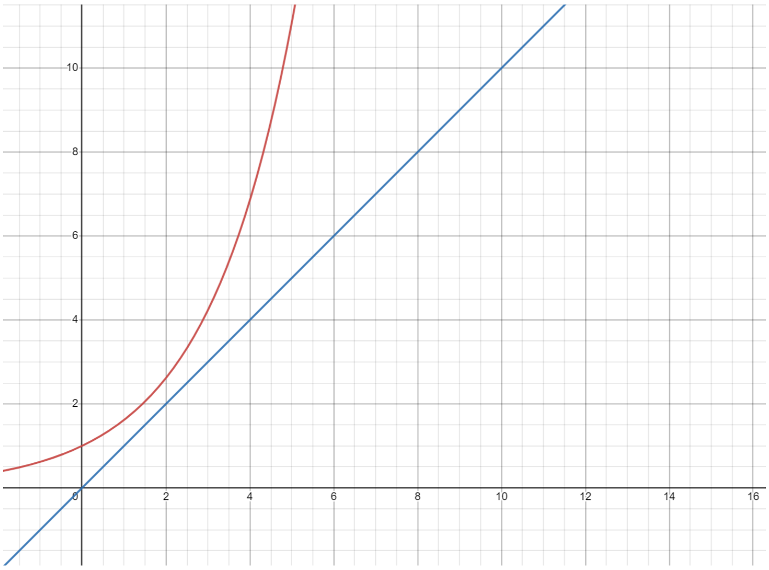
17.8

```
#include<stdio.h>

int main()
{
    int in;
    printf("which Fibonacci number?");
    scanf("%d", &in);

    int i , f1=1, f2=1;
    if(in==0||in==1) printf("that Fibonacci number is %d\n", f1);
    for(i=1;i<=in-2; i++){
        int t = f1;
        f1=f2;
        f2=t+f2;
    }
    printf(that Fibonacci number is %d\n", f2);
    return 0;
}
```

The running times of the recursive version of the Fibonacci and the iterative version on different values of n are shown below.



recursive version:red

iterative version:blue

Consider that under the iterative version, the running time of the function is a function on n, while under the recursive version, the running time of the function is a function on a^n, where a > 1, so the recursive version is significantly slower.

17.12

(a) No. Each time it is called, the argument to the function becomes smaller, so there is no such value.

(b) 6 function calls are made to func when the program is executed.

(c) 4

17.14

9

7

5

3

1

2

4

6

8

10