

Homework 9

17.2 Yes. Because every time the place we call the function is same.

17.8

```
1  int fibo(int n)
2  {
3      int a = 1;
4      int b = 1;
5      int c = 0;
6      if (n == 0) return a;
7      else if (n == 1)
8          return b;
9      else{
10         for (int i = 1; i < n; i++){
11             c = a + b;
12             a = b;
13             b = c;
14         }
15         return c;
16     }
17 }
```

In the cases that n is small, the running time is similar. When the n is very big, the recursion is slower.

Recursions run by calling the function itself. When calling the function, address saving and parameter passing are required for each call. This is realized through a recursive **stack**.

Specifically, the contents to be saved each time the function is called include: local variables, formal parameters, call function address, and return value. Then, if you call n times recursively, you need to allocate n local variables, n formal parameters, n call function addresses, and n return values, which is bound to affect efficiency.

17.12

(a) No.

(b) 6 times.

17.14

9

7

5

3

1

2

4

6

8

10