

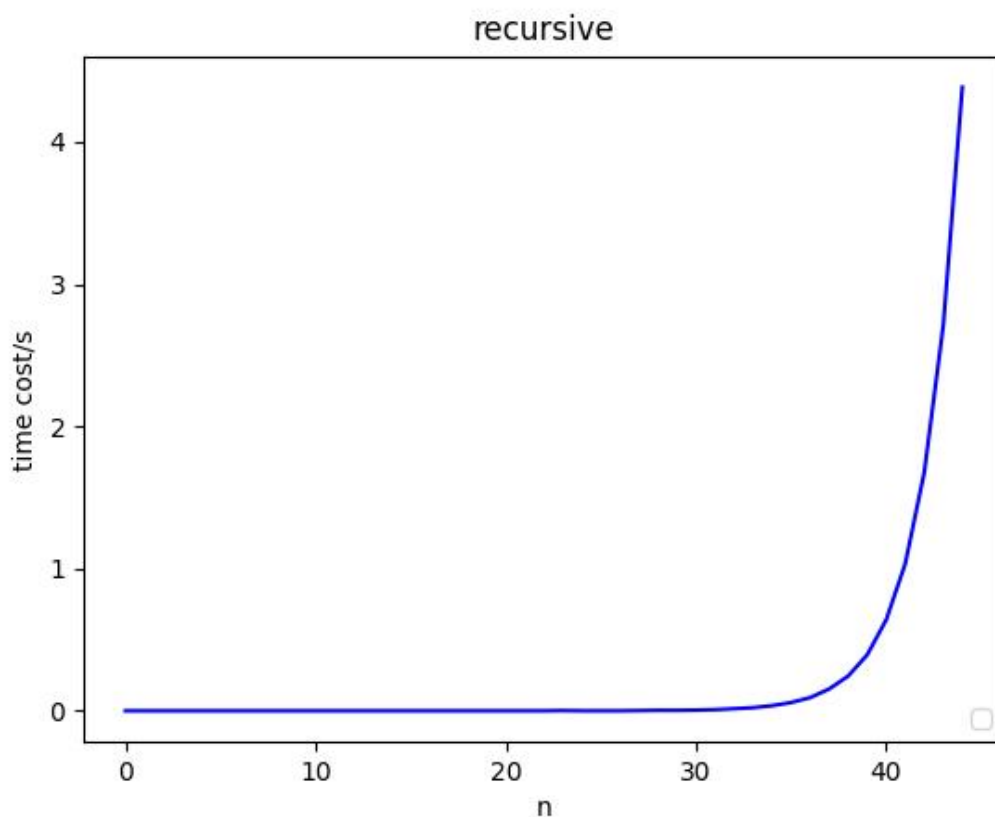
17.2

No. Because in a recursive function, there may be more than 1 positions where the recursive function calls itself. The return address is corresponding to the position where the function is called, so there are different return addresses if the function is called recursively at different position.

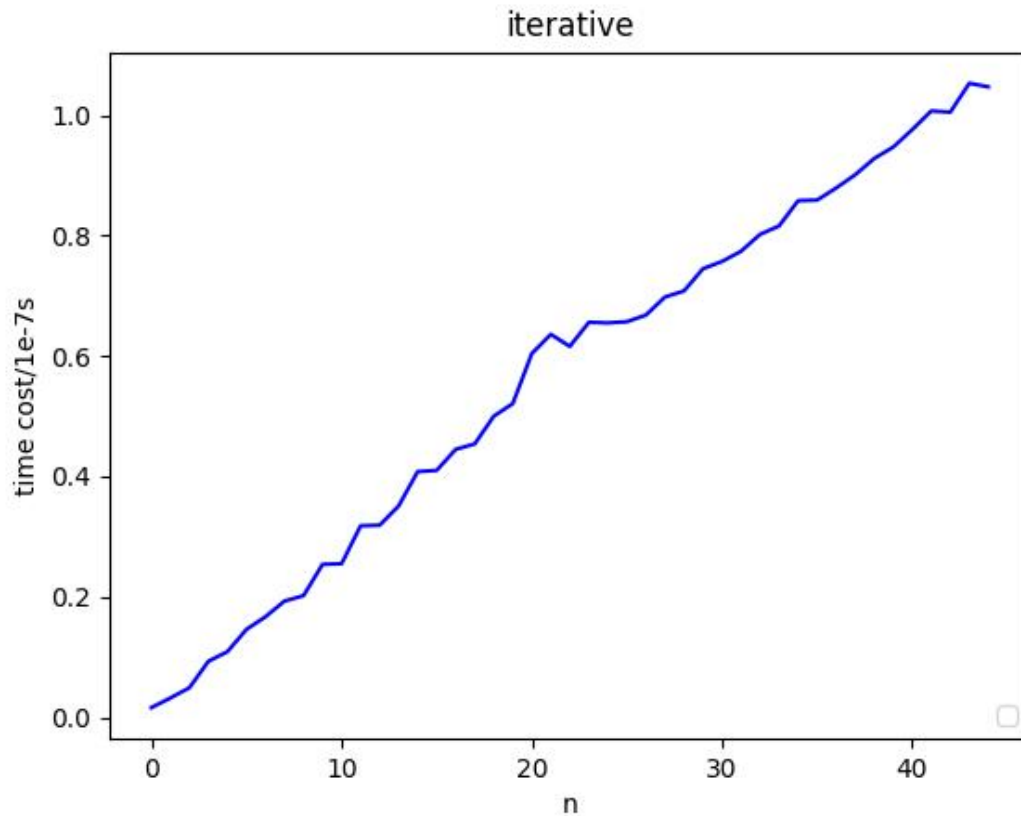
17.8

```
1  int Fibonacci_iterative(int n){
2      if(n == 0 || n == 1) return 1;
3      int i, a = 0, b = 1, c;
4      for(i = 0; i < n; ++i){
5          c = a + b;
6          a = b;
7          b = c;
8      }
9      return c;
10 }
```

The running time of recursive version:



The running time of iterative version:



The recursive version significantly slower. When the n th fibonacci number is found iteratively, the 0th, 1st, 2nd ... and the $(n-1)$ th fibonacci number are found just once. But when the n th fibonacci number is found recursively, the fibonacci numbers that have been found are not saved so they should be found for a lot of times. That brings a huge time cost.

17.12

a. No.

b. 6

17.14

9
7
5
3
1
2
4
6
8
10

