CS 101: Computer Programming and Utilization

Shivaram Kalyanakrishnan
(Abhiram Ranade's slides, borrowed and edited)
Lecture 18

Today's Lecture

- Recursion
- Examples

Euclid's theorem on GCD

THEOREM: If m % n == 0, then GCD(m, n) = n, else GCD(m,n) = GCD(n, m % n).

The theorem looks like a program!

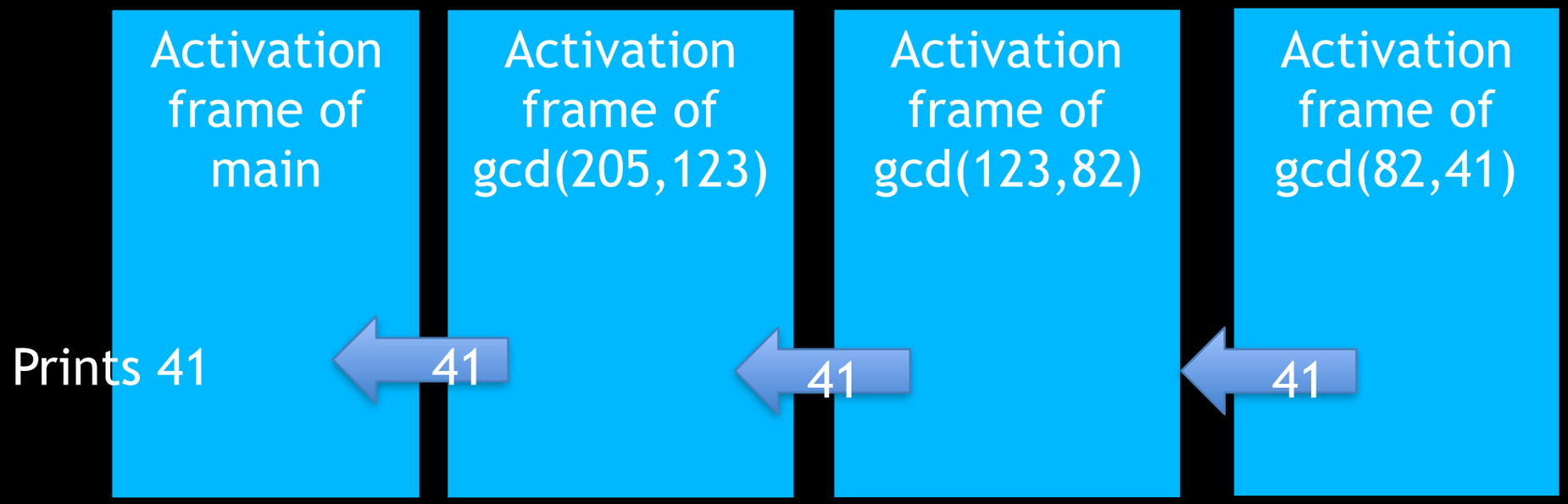
```
int gcd(int m, int n){
    if (m % n == 0) return n;
    else return gcd(n, m % n);
}
```

Will this work?

Execution

```
int gcd(int m, int n){
  if(m % n == 0) return n;
  else return gcd(n, m%n);
}
```

```
main_program{
    cout <<
 gcd(205,123)
       << endl;
        }
```



Recursion

- Recursion = The phenomenon of a function calling itself
  - Seems like we are defining the function in terms of itself
  - But no circularity if the arguments to the new call are different from the arguments in the original call.

- Each call executes in its own activation frame.

- Some call must return without another recursive call
  - Otherwise infinite recursion (error!)

- In the body of gcd there was just one recursive call. We can have several calls if we wish. Examples soon.

Comparison of recursive and non-recursive gcd

```
int gcd(int m, int n){
    if (m % n == 0) return n;
    else return gcd(n, m % n);
}
```

```
int gcd(int m, int n){
  while(m % n != 0){
   int r = m%n;
   m = n;
   n = r;
  }
  return n;
}
```

Recursive calls in gcd(205,123):

- gcd(123,82)
- gcd(82,41)

- Values of m,n in consecutive iterations of gcd(205,123):
- 205, 123
- 123, 82,
- 82, 41

- The two programs are ”really” doing the same calculations!
- But on the surface they look very different.

Remarks

- Recursion often produces compact, elegant programs.
  - Recursive programs might be slightly slower because they need to create activation frames etc.
- Recursion is also a way to discover algorithms.

Euclid quite possibly thought to himself:

- “Instead of doing laborious computation to find the gcd of 205 and 123, can I find two smaller numbers whose gcd is the same as that of 205 and 123?”
- This is recursive thinking! It is common in mathematics.
- We will see more examples soon.

Today's Lecture

- Recursion

- Examples