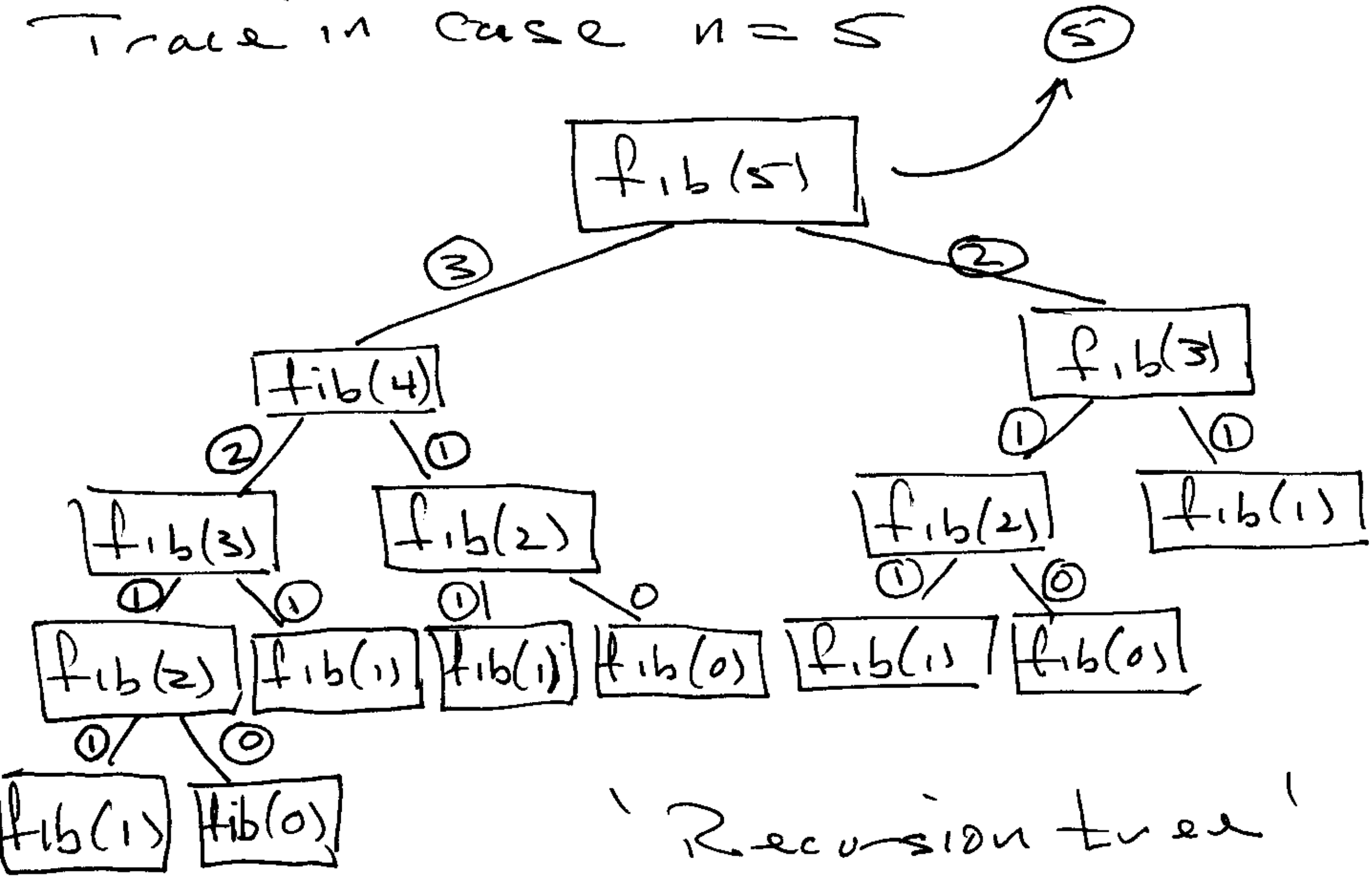
CSE 20   11-4-20

Fibonacci2.py: uses recursion

function that calls itself

Trace in case n = 5



'Recursion tree'

Euclidean Algorithm

Gcd(a, b) = largest integer that divides both a and b.

Defn d divides a (notation: $d|a$) iff there exist $k \in \mathbb{Z}$ such that

$$a = d \cdot k$$

Algorithm:

$$a = b \cdot q_1 + r_1$$

$$b = r_1 \cdot q_2 + r_2$$

$$r_1 = r_2 \cdot q_3 + r_3$$

$$\vdots$$

$$r_{n-3} = r_{n-2} \cdot q_{n-1} + \boxed{r_{n-1}} \leftarrow \text{Gcd}(a, b)$$

$$r_{n-2} = \boxed{r_{n-1}} \cdot q_n + \underset{0}{\overset{r_n}{\shortparallel}} \leftarrow \text{stop}$$

Ex. $\text{GCD}(45, 33) = 3$

$$\begin{cases} 43 // 33 \to 1 \\ 43 \% 33 \to 12 \end{cases}$$

$$45 = 33 \cdot 1 + 12$$

$$33 = 12 \cdot 2 + 9$$

$$12 = 9 \cdot 1 + \boxed{3} \leftarrow \text{gcd}$$

$$9 = \boxed{3} \cdot 3 + 0$$

(div. of 45) $= \{1, 3, 5, 15, 45\}$

(div. of 33) $= \{1, 3, 11, 33\}$

(common div. of 45 & 33) $= \{1, 3\}$

gcd

still to come

- break stmt
- continue stmt

templates:

```
while cond: # continue transfers here
    stmt
      :
    break
      :
    continue
      :
    stmt
# end while
# break transfers here
```

can also do this in a for loop

```
for var in list:
    |       |
    |-break
    |       |
    |  continue
    |       |
#end for
```

<u>Ex.</u> Break.py

Break2.py

Continue.py next time