Rekursion

I P5.JavaScript og Python

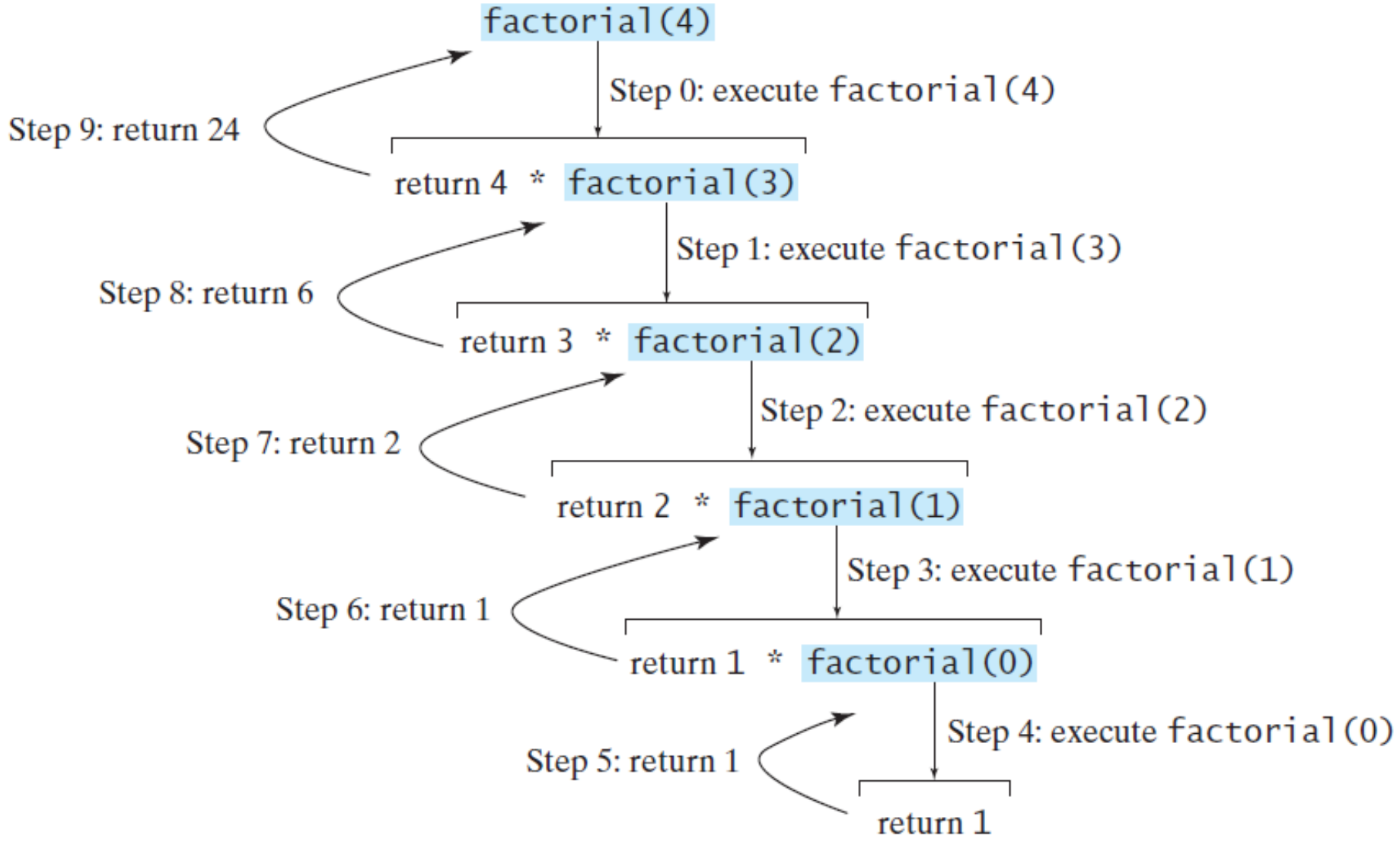
Kort fortalt

- Rekursion er en teknik, der leder til elegante løsninger til problemer, som ville være svære at programmere ved brug af løkker
- Rekursive funktioner kalder sig selv.
- Hvis du tænker rekursivt kan mange problemer løses rekursivt

$$0! = 1;$$
$$n! = n \times (n - 1)!;\ n > 0$$

Herover er faktultetsfunktionen. Prøv at programmere den.

Funktionskaldet



Fakultetsfunktionen i python – prøv at kalde den og forstå hvad der sker:

```
function factorial(n){
  if(n==1) {
    return(1);
  }
  else {
    return(n*factorial(n-1));
  }

}
```

Øvelser

1. Skriv en matematisk definition for at beregne 2^n for positiv n
2. Skriv en matematisk definition for at beregne x^n for reelt tal n
3. Skriv en matematisk definition for at bergene 1+2+3+....+n for positiv n
4. Skriv en matematisk definition for at bergene 1*2*3*....*n for positiv n
5. Skriv endelig nogle rekursive funktioner, der løser opgaverne ovenfor

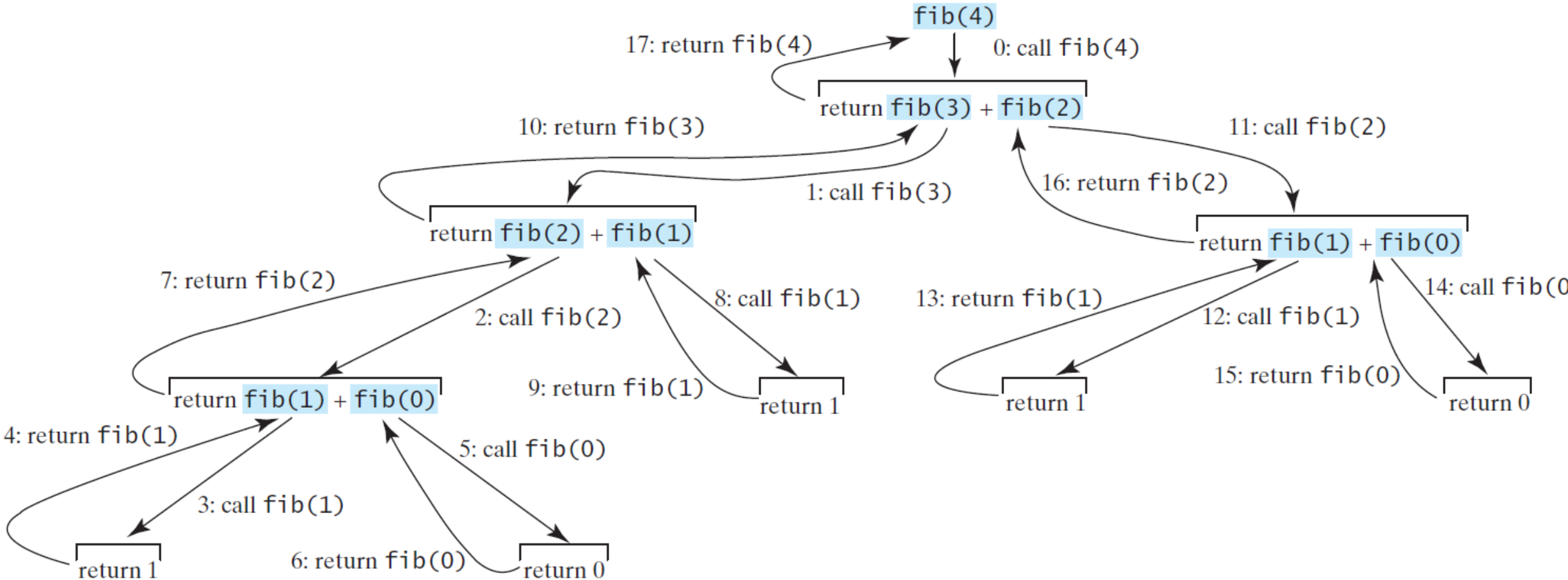
Fibonacci – Prøv at skrive den i javascript

0	1	1	2	3	5	8	13	21	34	55	89	.	.
0	1	2	3	4	5	6	7	8	9	10	11		

```
fib(0) = 0
fib(1) = 1
fib(index) = fib(index - 2) + fib(index - 1); index >= 2
```

Skriv en python/java funktion der beregner Fibonnaci

```
function setup() {
  createCanvas(400, 400);
}

function draw() {
  background(220);

  for (var i = 0;  i <= 10; i++) {
    var s = "fib(" + i + ") = " + fib(i);
    text(s, 100, 50 + i * 20);
  }
}

function fib(n) {
  if (n <= 0) {
    return 0;
  } else if (n == 1) {
    return 1;
  } else {
    return fib(n - 1) + fib (n - 2);
  }
}
```



n	2	3	4	10	20	30	40	50
# of calls	3	5	9	177	21891	2692537	331160281	2075316483

Hvad sker der i følgende kode?

```
void drawCircle(int x, int y, float radius) {
  ellipse(x, y, radius, radius);
  if(radius > 2) {
    radius *= 0.75f;
    drawCircle(x, y, radius);
  }
}
```

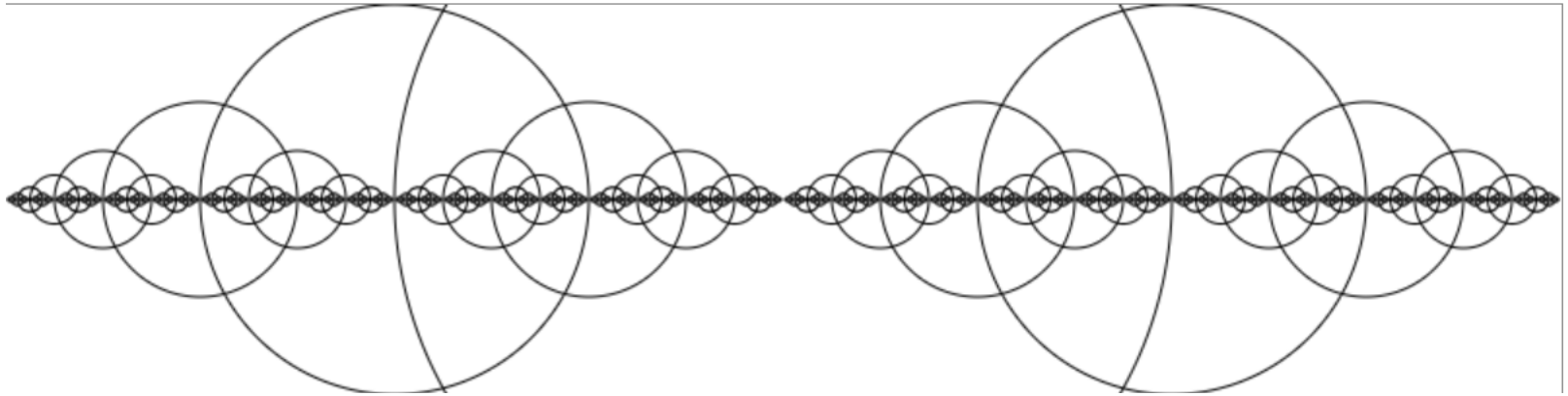
The drawCircle() function is calling itself recursively.

- Prøv gerne at skrive den selv

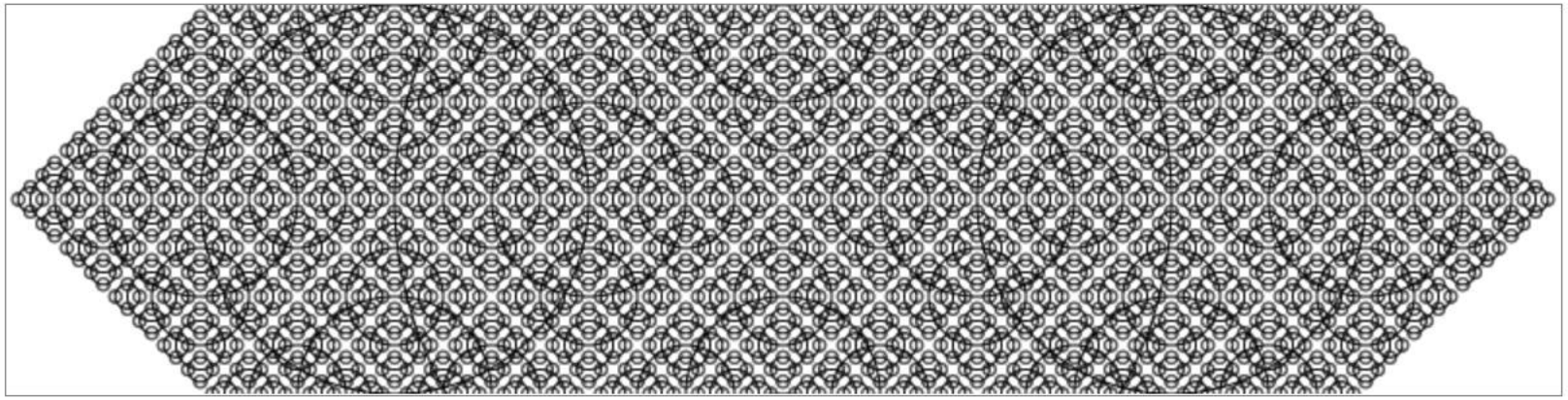
Hvad sker der mon i følgende kode? Kommenter!

```
function setup() {
  createCanvas(720, 560);
  noStroke();
  noLoop();
}

function draw() {
  drawCircle(width / 2, 280, 6);
}

function drawCircle(x, radius, level) {
  // 'level' is the variable that terminates the recursion once it rea
  // a certain value (here, 1). If a terminating condition is not
  // specified, a recursive function keeps calling itself again and ag
  // until it runs out of stack space - not a favourable outcome!
  const tt = (126 * level) / 4.0;
  fill(tt);
  ellipse(x, height / 2, radius * 2, radius * 2);
  if (level > 1) {
    // 'level' decreases by 1 at every step and thus makes the termina
    // attainable
    level = level - 1;
    drawCircle(x - radius / 2, radius / 2, level);
    drawCircle(x + radius / 2, radius / 2, level);
  }
}
```

Her kommer resultatet af koden

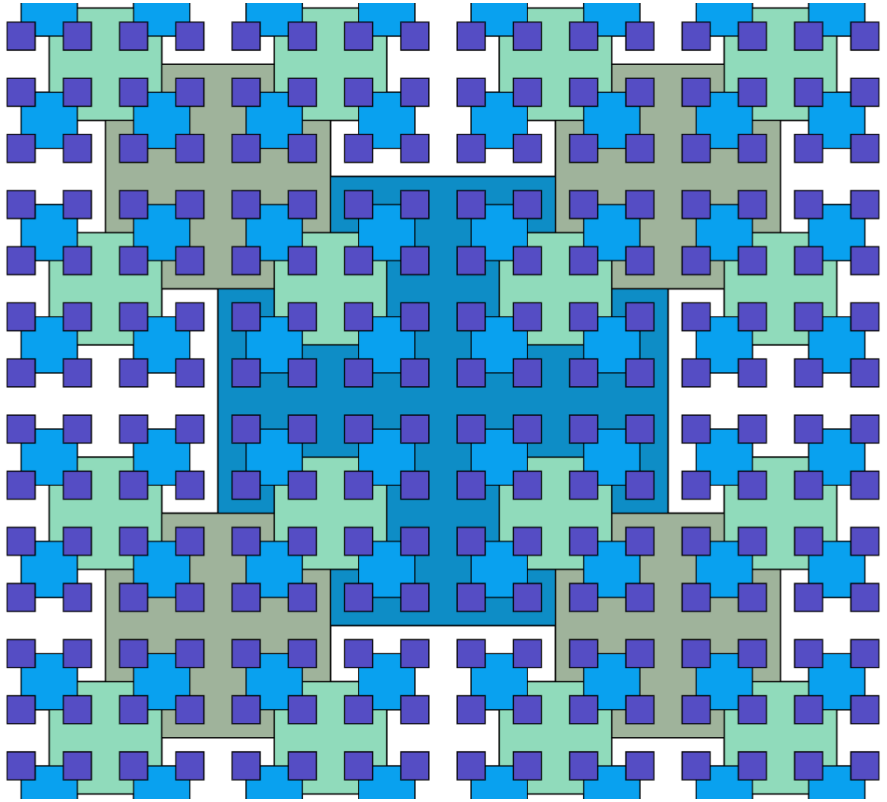


Prøv at udvide den så I får noget ala følgende:



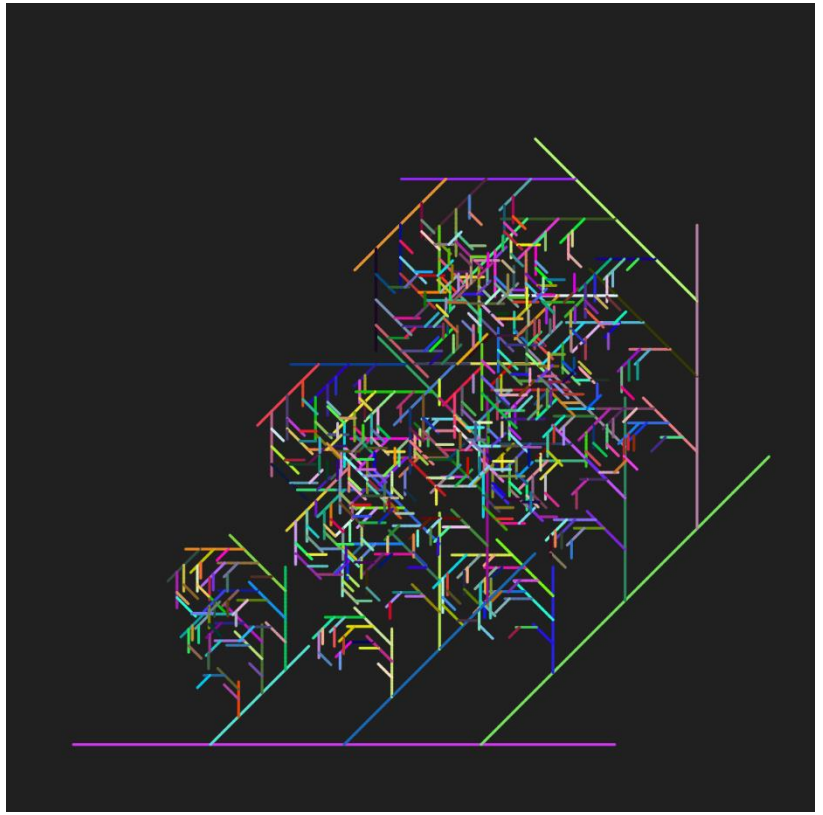
Hvad gør følgende kode og hvor I består det rekursive kald?

- https://editor.p5js.org/jeyv/sketches/w8e0rS9-U



Hvor er det rekursive kald og hvad gør koden?

Se koden her:
https://editor.p5js.org/KevinWorkman/sketches/09uOivsQN



Generelt for rekursive funktioner

- De implementeres typisk ved brug af if-else eller switch
- De indeholder et basistilfælde
- De indeholder et rekursivt kald

Øvelse – brug rekursion:

Skriv et program – gerne i pseudokode, der printer en besked n-gange

Skriv et program – gerne i pseudokode, der drikker en kop kaffe

Rekursive løsninger

```
def nPrintln(message, n):
    if n >= 1:
        print(message)
        nPrintln(message, n - 1)
```

```
def drinkCoffee(cup):
    if cup is not empty:
        cup.takeOneSip()
        drinkCoffee(cup)
```

Flere øvelser om rekursion

1. Skriv en rekursiv funktion, der beregner eksponenten af et tal
2. Skriv en rekursiv funktion, der beregner den harmoniske serie: $1+\frac{1}{2}+\frac{1}{3}+\frac{1}{4}+\frac{1}{5}+\cdots$
3. Skriv en rekursiv funktion der undersøger om et tal er lige eller ulige
4. Skriv en rekursiv funktion, der foretager binær søgning på en liste af
5. Skriv en rekursiv funktioner, der undersøger om et ord er et palindrome
6. Skriv en rekursiv funktion der returnerer et reversibelt ord
7. Skriv en rekursiv funktion, der finder største fælles divisor af to tal
8. Skriv en rekursiv funktion, der returnerer alle tænkelige permutationer af et ord (eksempel: ”abc” så er ”acb” en permutation.
9. Skriv et rekursivt program der tager et heltal input N og beregner en approksimation af det det gyldne snit (prøv gerne at google det gyldne snit, det forekommer mange steder i naturen:

$$f(N) = 1 \quad \text{hvis } N = 0$$
$$= 1 + 1 / f(N-1) \quad \text{hvis } N > 0$$

Prøv også at lave disse opgaver uden rekursion

Endnu flere øvelser

10. Skriv en rekursiv funktion, der beregner mindste fælles divisor
11. Skriv en rekursiv funktion, der konstruerer Pascals Trekant
12. Skriv en rekursiv funktion, der finder det største tal i en liste
13. Skriv en rekursiv funktion, der sorterer en liste af tal (Bubble sort eller mergesort)
14. Skriv en rekursiv funktion, der beregner Ackermans Funktion:

$$A(m,n) = \begin{cases} n+1 & \text{if } m = 0 \\ A(m-1,1) & \text{if } m > 0 \text{ and } n = 0 \\ A(m-1, A(m,n-1)) & \text{if } m > 0 \text{ and } n > 0 \end{cases}$$

15. Skriv en rekursiv funktion, der beregner Tribonacci rækkefølgen (starter med 0,0,1)

Læs Kapitel 8 i Nature of Code: https://natureofcode.com/book/chapter-8-fractals/ og forstå eksemplerne.

Løs så mange af øvelserne 8.1 til 8.5 i mindre grupper

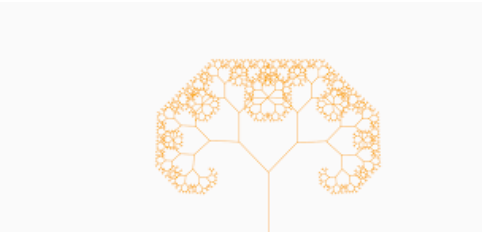
Rekursion og fraktaler

I skal udvælge og arbejde med og præsentere for resten af klassen i mindre grupper en af følgende:

1) Sjov med rekursion i p5: https://www.youtube.com/watch?v=jPsZwrV9ld0. Se videoen og prøv at modificere koden og lave jeres egen mønster ved brug af rekursion

2) IFS fraktal. Se under afsnit 5.rekursion i mat a bogen for vejledning. Målet er at I skal have visualiseret et bregne blad. I kan også læse om Feigenbaum fraktalen og prøve at lave den (se samme kapitel).

3) Fraktal træer. Se https://www.youtube.com/watch?v=0jjeOYMjmDU og prøv jeres eget fraktaltræ.



4) Se https://www.youtube.com/watch?v=YxOc8AvwQtM og prøv at justere på koden