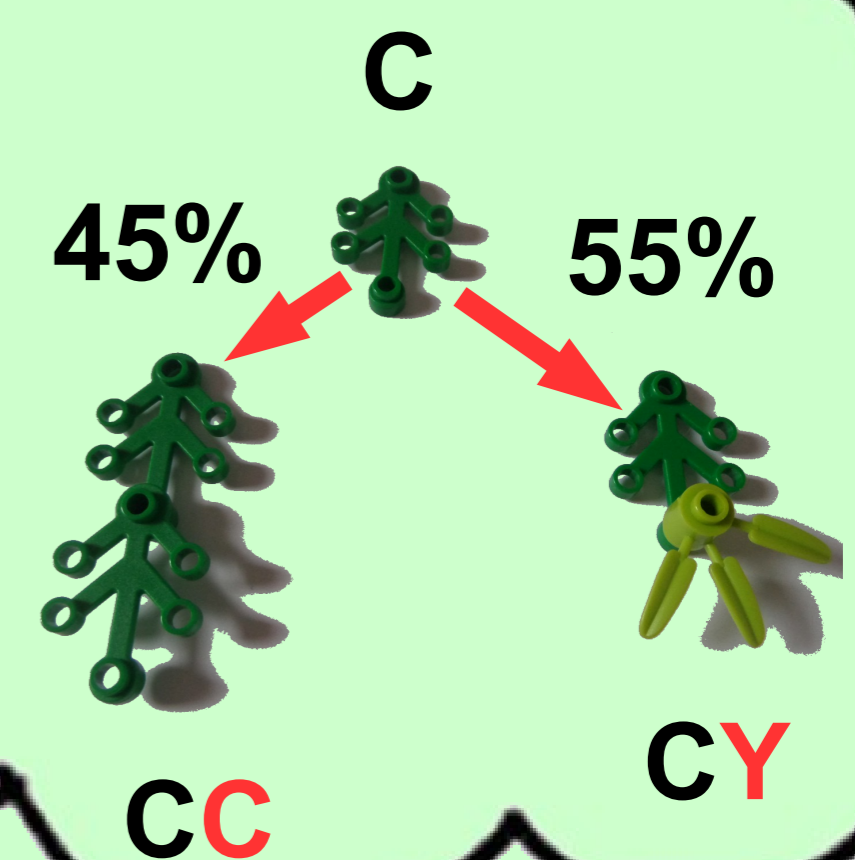
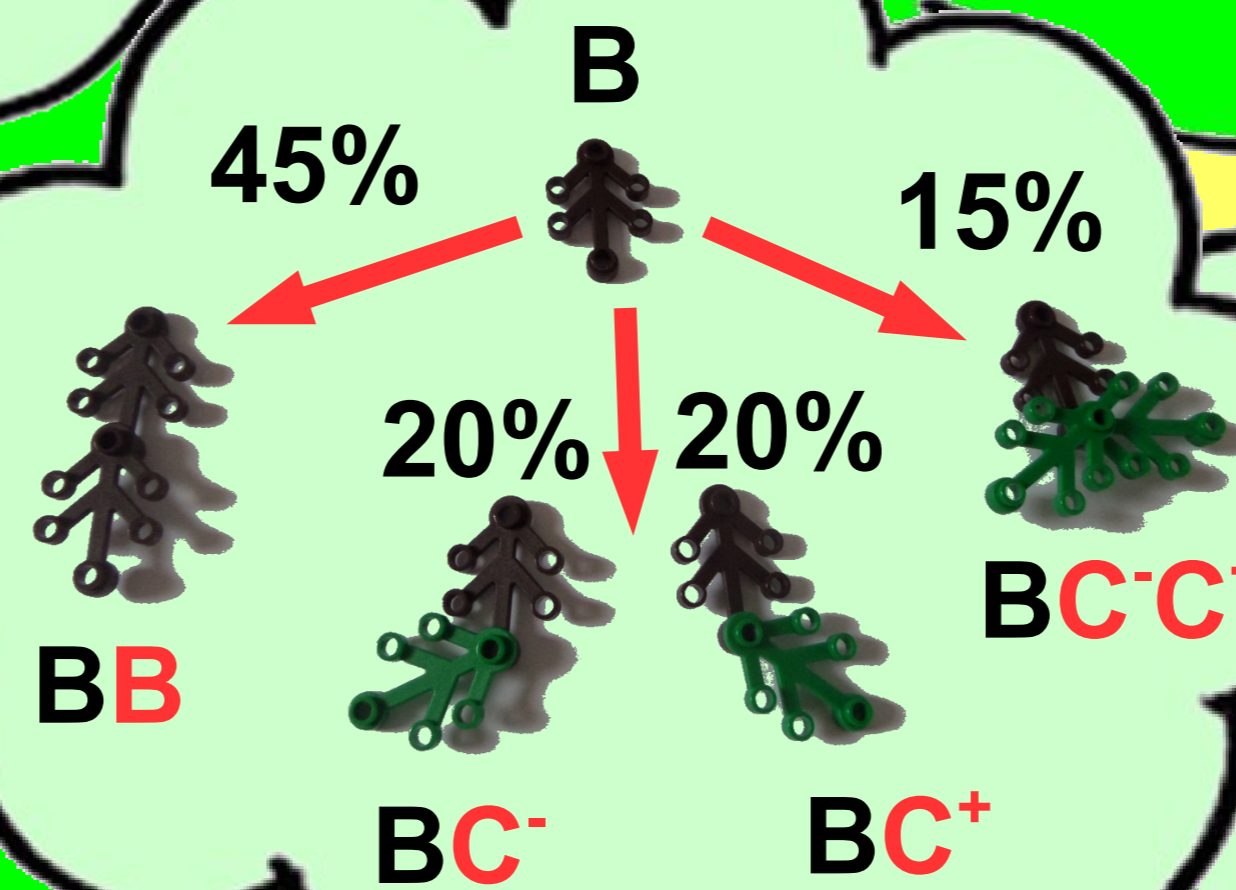
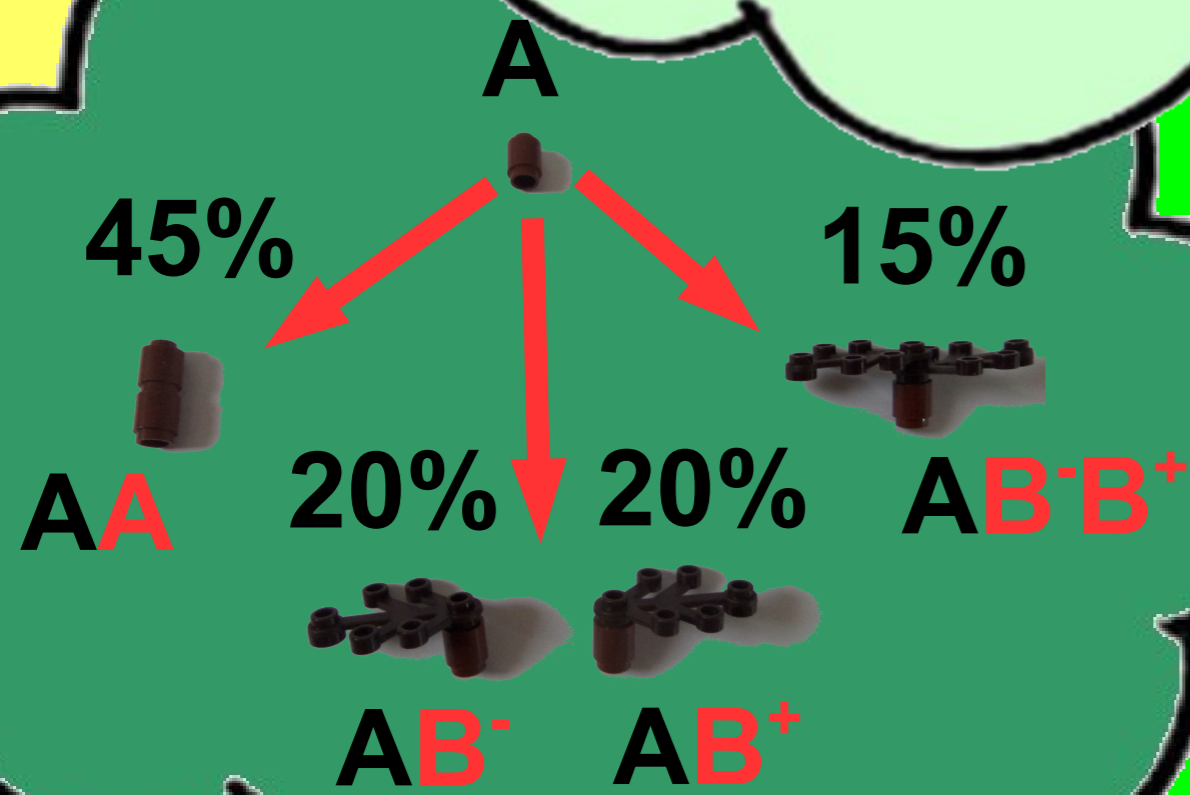


# Modelisation the growth of the tree

BY  
GANDOUIN Gaël,  
PALAU Gabriel,  
SIMON Thomas,  
MAGNIER Evan,  
FOUCHER-REMY Auxence,  
& STACHOULIS Georgios  
Students of Lycée  
Val de Durance de Pertuis

A tree is modeled  
by 4 parts, which evolve  
each in a certain way:  
Has the trunk  
B-the branches  
C-the leaves  
Y-les buds (end of evolution)



Framing the number of  
A, B, C and Y

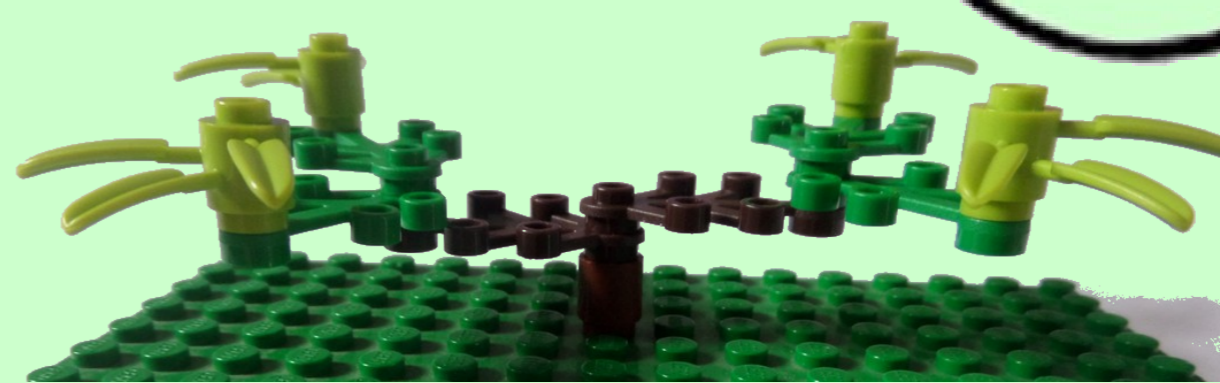


Vidéo-  
conférence  
with Cluj



Presentation  
at the  
Avignon  
Congress

A: $A_{min}=1$ $A_{max}=n-2$	$n$	3	4	5	6
B: $B_{min}=2$ $B_{max}=\frac{n^2-3n+4}{2}$	$y_{max}$	4	7	11	16
C: $C_{min}=4$ $C_{max}=\frac{n^2-3n^2+8n}{6}$	$c_{max}$	4	8	15	26
$y_{min}=4$ $y_{max}=\frac{n^2-n+2}{2}$	$B_{max}$	1	4	7	11
	$A_{max}$	1	2	3	4

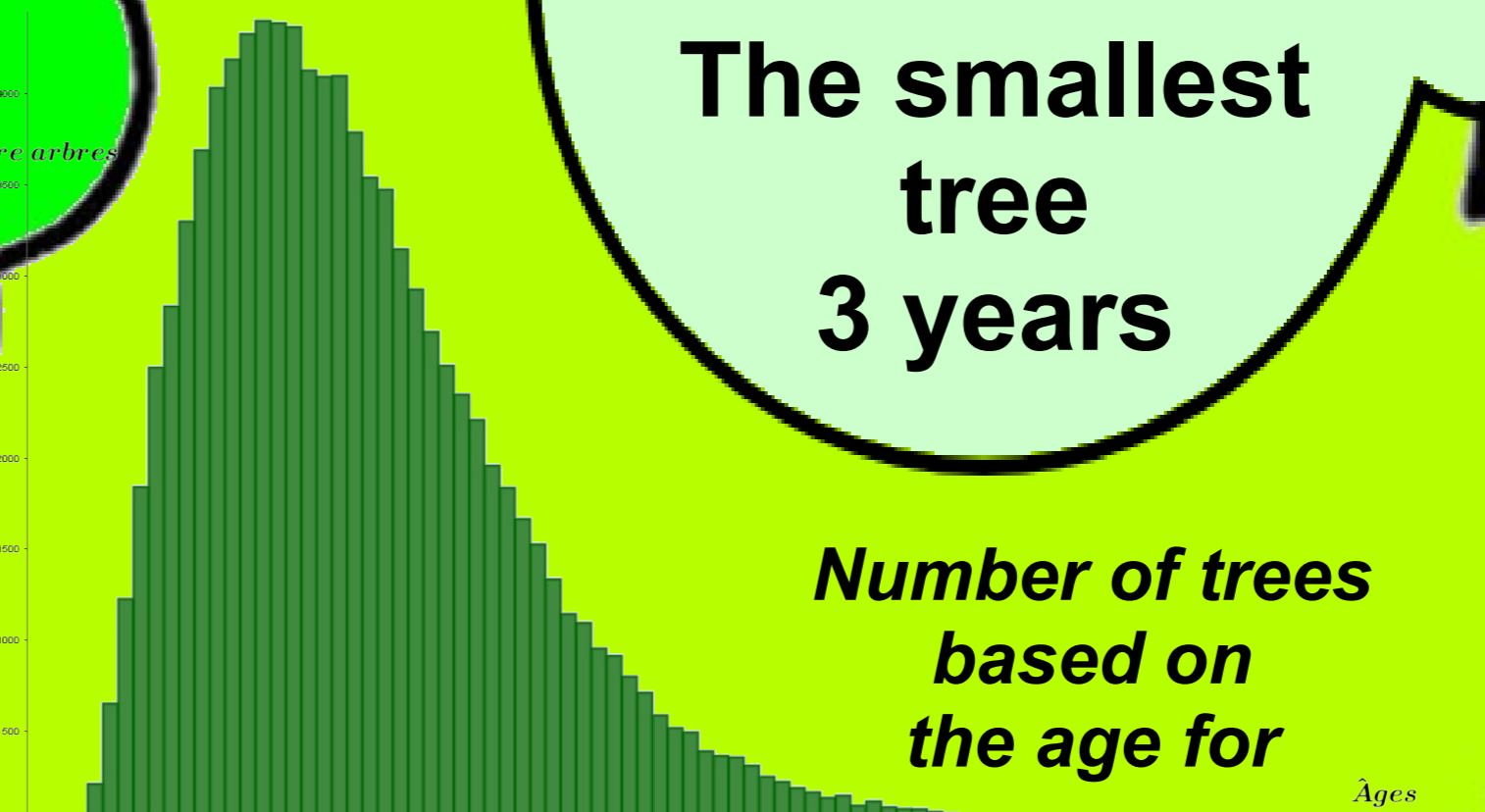


A → BB → CCCC → YYYY

The smallest  
tree  
3 years



Research  
work



Number of trees  
based on  
the age for  
100,000 simulations

Average tree age  
**21,23 ans**  
(on a simulation of 100,000 trees)

Age = number of changes  
until you only have Y's

Meeting with the INRIA  
Sophia Antipolis researcher

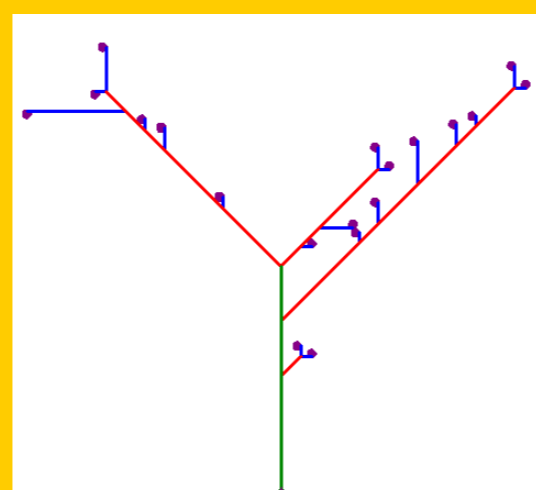


```
# Values of evolutionary probabilities
AA=0.45
AAB=0.40
ABB=0.15
BB=0.45
BBC=0.4
BCC=0.15
CC=0.45
Cy=0.55
```

```
# After "A"
from random import random
def imageA(AA,AAB):
    L=[]
    x=random()
    if x<AA:
        L.append("A")
    elif x<=AA+AAB:
        L.append("A")
        L.append("B")
    else:
        L.append("B")
        L.append("B")
    return L
```

```
# After "B"
def imageB(BB,BBC):
    L=[]
    x=random()
    if x<=BB:
        L.append("B")
    elif x<=BB+BBC:
        L.append("B")
        L.append("C")
    else:
        L.append("C")
        L.append("C")
    return L
```

```
# After "C"
def imageC(CC):
    L=[]
    x=random()
    if x<=CC:
        L.append("C")
    else:
        L.append("Y")
    return L
```



Simulation of a tree with  
Thomas' program

Presentations in colleges



```
# Number of "Y" s in the tree
def nbY(L):
    c=0
    for i in range(len(L)):
        if L[i]=="Y":
            c=c+1
    return c
```

# ----- start -----

```
def evolution():
    L=["A"]
    n=len(L)
    evo=0
    while n>=1:
        M=[]
        evo=evo+1
        for i in range(n):
            Lire=L[i]
            if Lire=="A":
                M=M+imageA(AA,AAB)
            if Lire=="B":
                M=M+imageB(BB,BBC)
            if Lire=="C":
                M=M+imageC(CC)
            if Lire=="Y":
                M.append("Y")
        L=M
        n=len(M)
        if n==nbY(L):
            n=0
    return (evo)
```

```
R=[]
z=0
echantillon=100000
for i in range (echantillon):
    r=evolution()
    z+=r
    while len(R)<r+1:
        R.append(0)
    R[r]=1
    print(z/echantillon)
    print("max:"+str(len(R)))
print(R)
```

