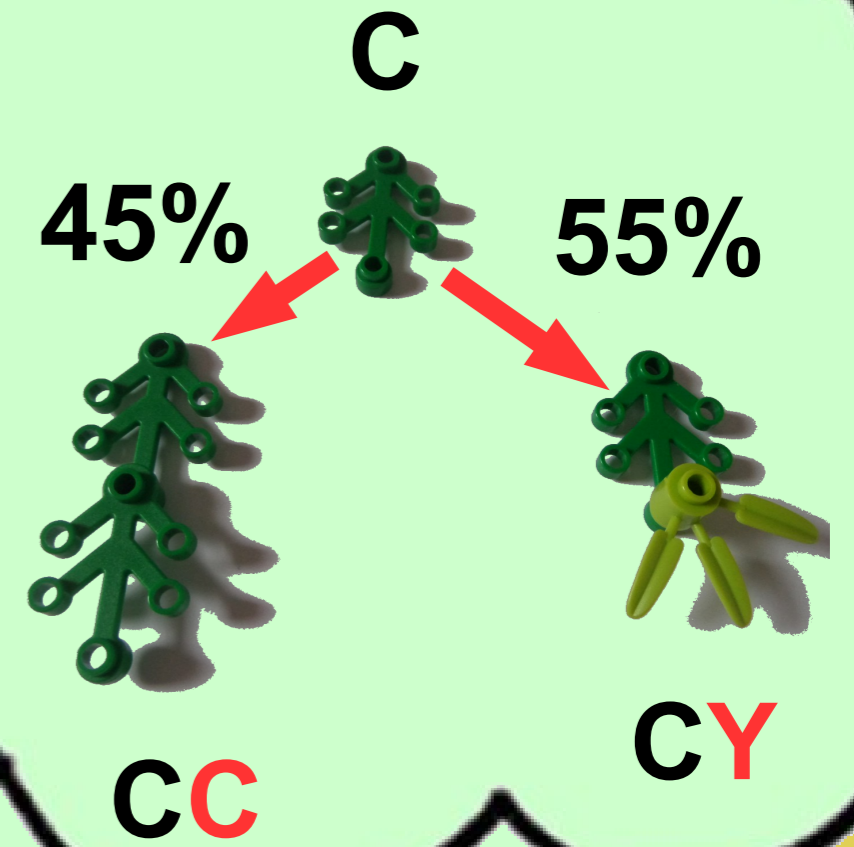
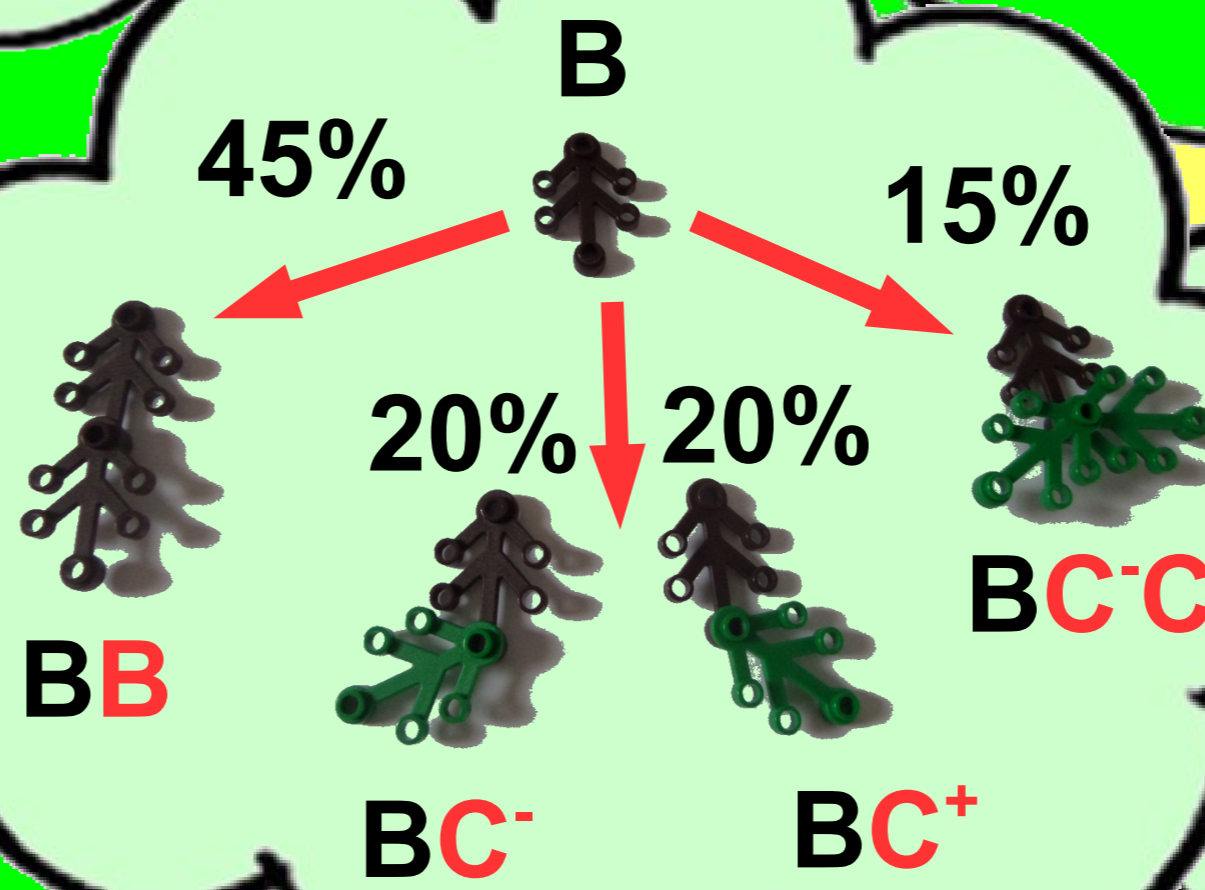
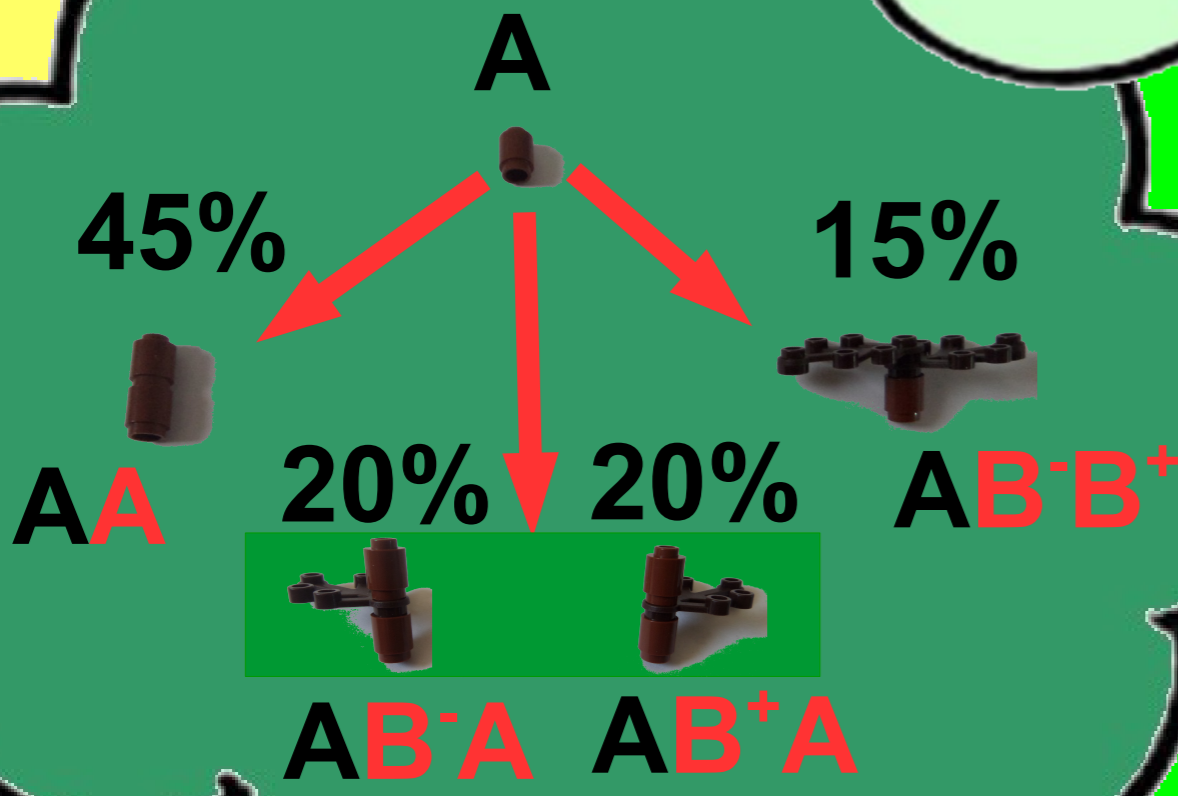


Modélisation de la croissance d'un arbre

PAR
GANDOUIN Gaël,
PALAU Gabriel,
SIMON Thomas,
MAGNIER Evan,
FOUCHER-REMY Auxence,
& STACHOULIS Georgios
 Élèves du Lycée Val de Durance de Pertuis

Un arbre est modélisé par 4 parties, qui évoluent chacune d'une certaine manière :
 A-le tronc
 B-les branches
 C-les feuilles
 Y-les bourgeons (fin de l'évolution)



Encadrement du nombre de A, B, C et Y

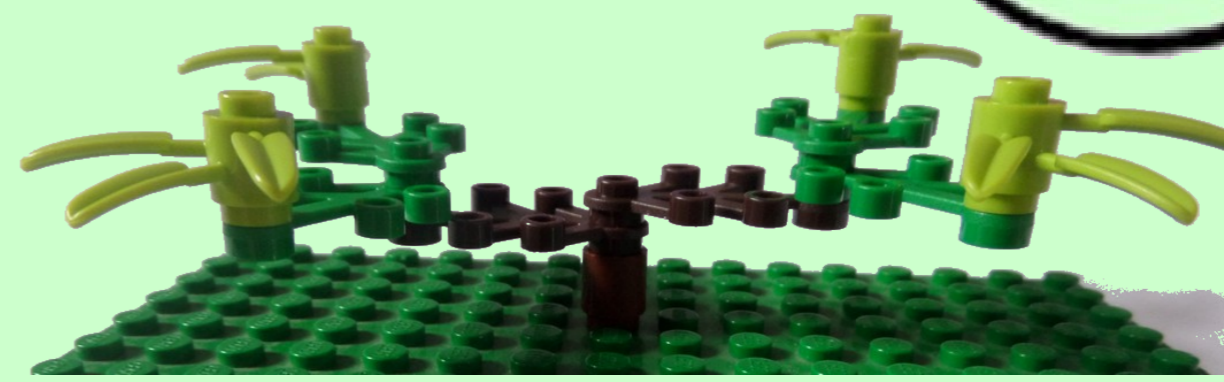


Vidéo-conférence avec Cluj



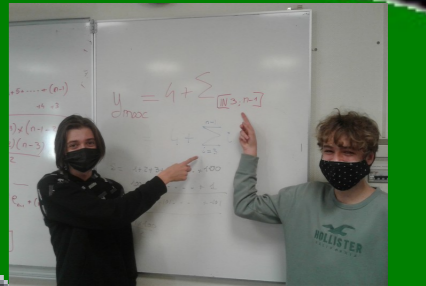
Présentation au congrès d'Avignon

A: A _{min} =1 A _{max} =n-2	n	3	4	5	6
B: B _{min} =2 B _{max} =n-3n+4 2	B _{max}	4	7	11	16
C: C _{min} =4 C _{max} =n-3n+8n 6	C _{max}	4	8	15	26
Y: Y _{min} =4 Y _{max} =n-n+2 2	B _{max}	1	4	7	11
	A _{max}	1	2	3	4

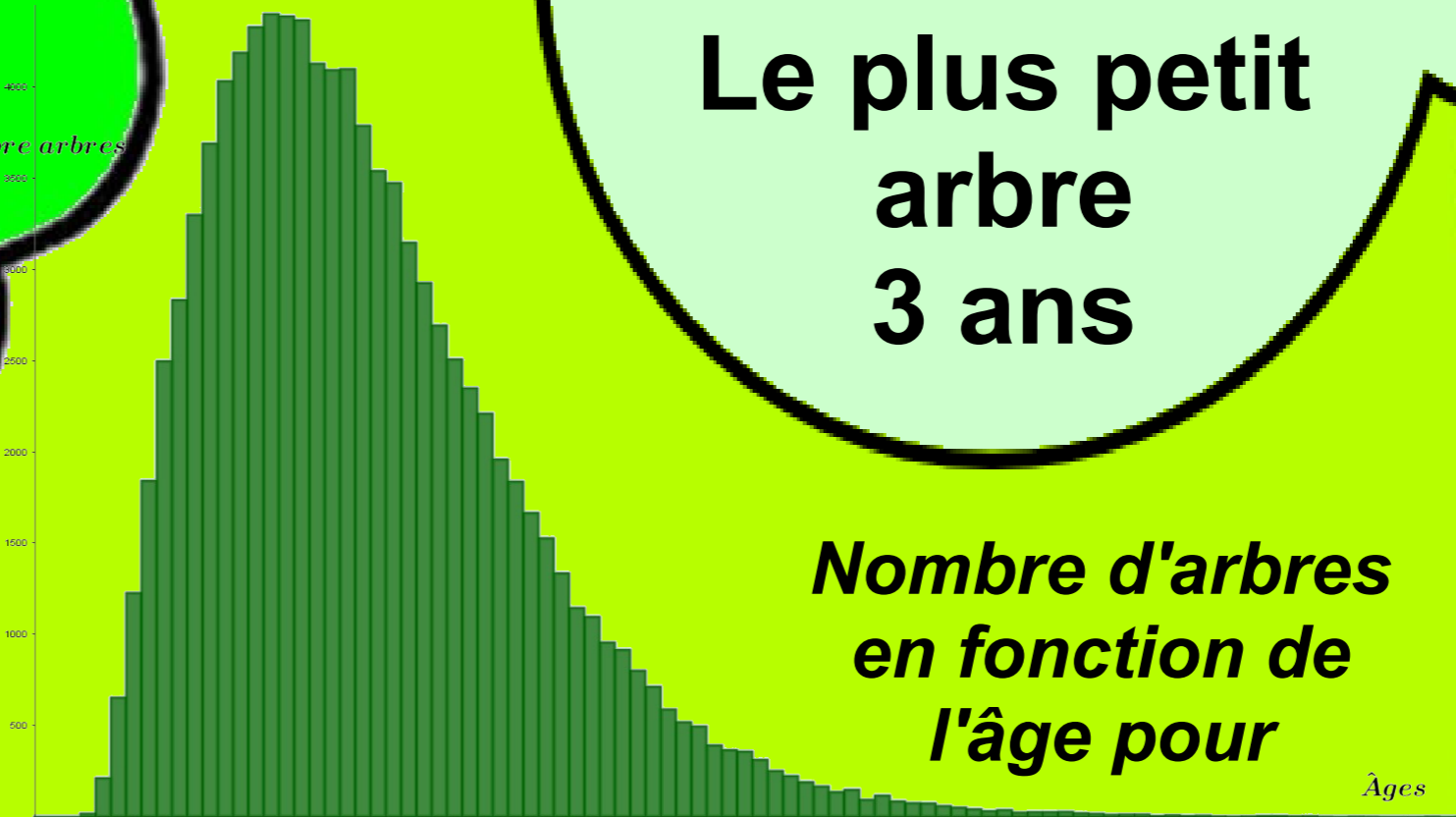


A → BB → CCCC → YYYY

Le plus petit arbre 3 ans



Travail de recherche



Nombre d'arbres en fonction de l'âge pour

100 000 simulations

Âge moyen des arbres
21,23 ans

(sur une simulation de 100 000 arbres)

Âge = nombre d'évolutions jusqu'à avoir que des Y

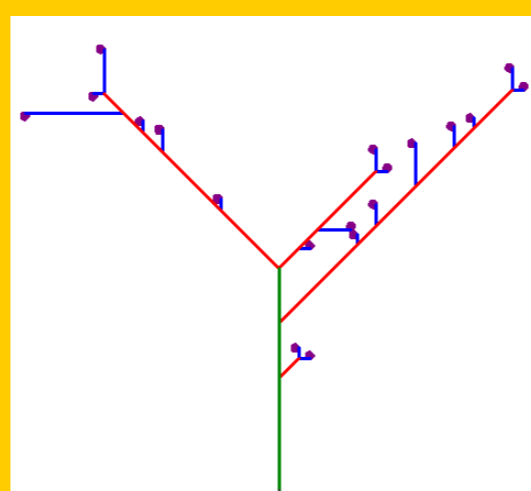


Présentations dans les collèges



Rencontre avec le chercheur de l'INRIA

Présentations à Cluj, lors de la mobilité ERASMUS



Simulation d'un arbre avec le programme de Thomas

```
# 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
```

```
# 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
    R.append(0)
    R[r]+=1
print(z/echantillon)
print("max:"+str(len(R)))

print(R)
```

