année des mathématiques 2019 2020

Séminaire académique – 29 janvier 2020 :

« Mathématiques & Numérique »

Codage et mathématiques

L'algorithmique au service des mathématiques

Les mathématiques au service de l'algorithmique

Yann Retoré – Lycée Gabriel Fauré – Foix – Enseignant de mathématiques / NSI
Olivier GINESTE – Lycée Pierre Bourdieu – Fronton – Enseignant de mathématiques /NSI

Région académique
OCCITANIE

académie Toulouse

L’algorithmique au service des mathématiques

année des mathématiques 2019 2020

- Exemple en cycle 4
- Exemple en seconde
- Exemple en EDS

Région académique OCCITANIE

académie Toulouse

L'algorithmique dans la résolution de problème en Cycle 4

année des mathématiques 2019 2020

Problème

Jean-Claude a fabriqué un petit bassin contenant 500L d'eau. Au début de l'été, il vide ce bassin pour le nettoyer puis le remplit entièrement. Du fait de l'évaporation, 5% de l'eau du bassin s'évapore chaque jour. Dès que le bassin contient moins de 350L d'eau le soir au coucher du soleil, Jean-Claude ajoute 150L d'eau durant la nuit.

Jean-Claude, soucieux de sa dépense d'eau, souhaite déterminer le volume d'eau global qu'il a utilisé durant tout l'été.
Combien Jean-Claude aura-t-il versé d'eau dans son bassin au bout de 60 jours?

… une approche algorithmique permet de résoudre efficacement ce problème!

Région académique OCCITANIE
académie Toulouse

L'algorithmique dans la résolution de problème en Cycle 4

Au cycle 4 la modélisation par une suite numérique n'est pas disponible …

Seule une approche itérative au fil des jours d'été qui s'écoulent va permettre d'avancer dans le problème mais le nombre de jours (60) est suffisamment grand pour automatiser l'itération.

La notion de boucle pour l'itération et la notion d'instruction conditionnelle pour tester la quantité d'eau dans le bassin sont quant à elles disponibles et mobilisables au cycle 4.

Liberté • Égalité • Fraternité
RÉPUBLIQUE FRANÇAISE
académie Toulouse
Région académique
OCCITANIE

L'algorithmique dans la résolution de problème en Cycle 4

Identification des paramètres de l'algorithme:
Q la quantité d'eau dans le bassin, n le nombre de jours écoulés.

La contrainte d'évaporation quotidienne conduit à la boucle:

```
Tant que n<60 faire
    Q←Q-5%Q
Fin Tant que
```

La contrainte de remise à niveau du bassin sous le seuil de 350 L conduit au test:

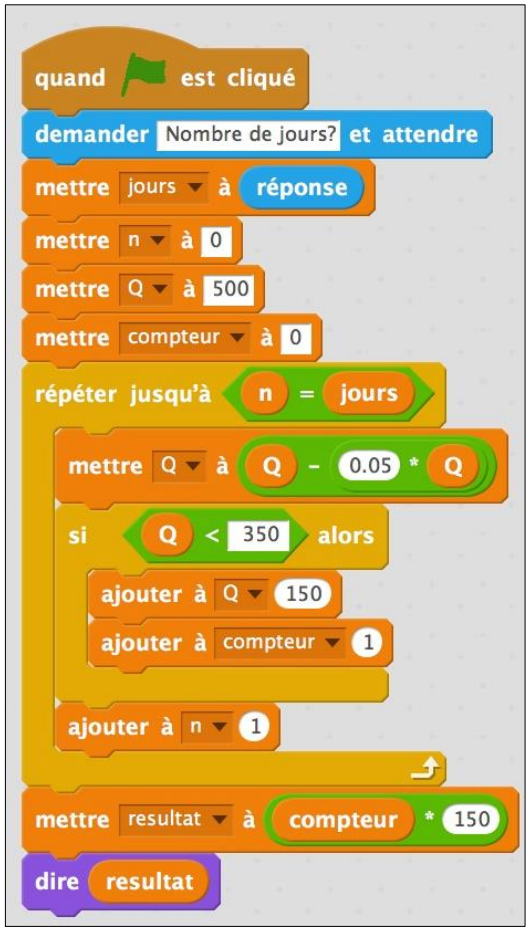
```
Si Q<350 faire
    Q←Q+150
Fin Si
```

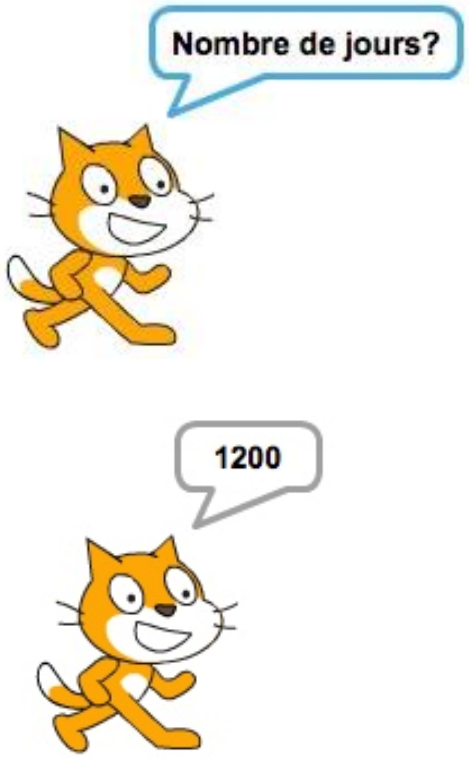
Région académique
OCCITANIE

L'algorithmique dans la résolution de problème en Cycle 4

année des mathématiques 2019 2020

Au collège ou en début de seconde avec Scratch





Au lycée avec une fonction python

```
def quantite_eau_ajoutee(jours):
    n=0
    Q=500
    compteur=0
    while n<jours:
        Q=Q-0.05*Q
        if Q<350:
            Q=Q+150
            compteur=compteur+1
        n=n+1
    resultat=compteur*150
    return resultat
```

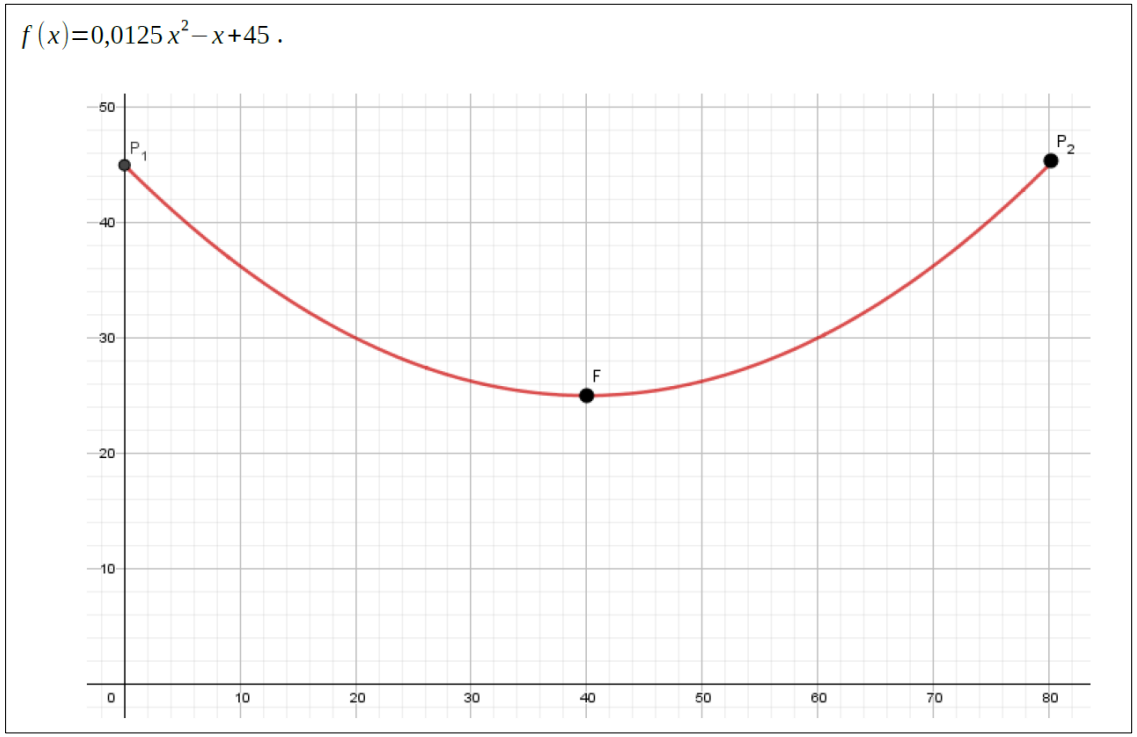
```
>>> quantite_eau_ajoutee(60)
1200
```

Région académique OCCITANIE

académie Toulouse

L'algorithmique dans la résolution de problème en Seconde

année des mathématiques 2019 2020



Quelle est la longueur du câble entre 2 pylônes ?

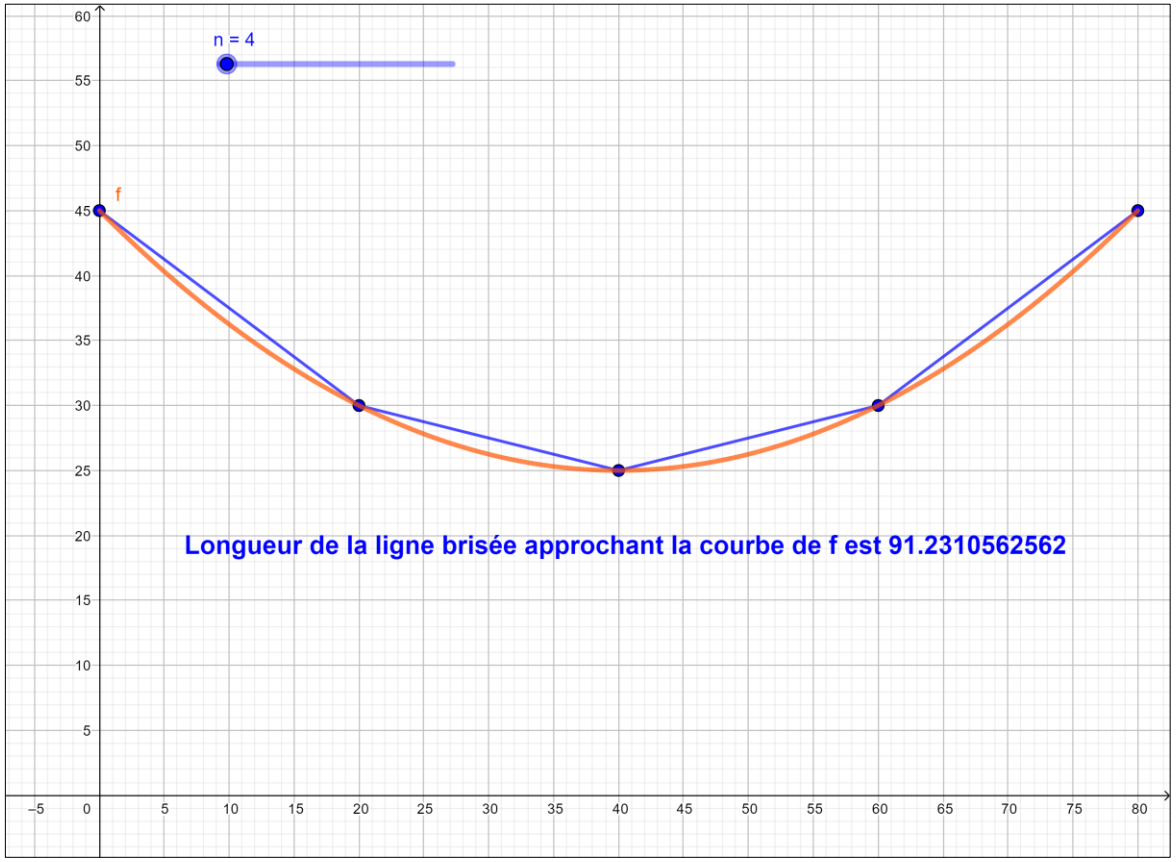
Région académique OCCITANIE

académie Toulouse

L'algorithmique dans la résolution de problème en Seconde

année des mathématiques 2019 2020

Ici encore l'outil mathématique pour réaliser ce calcul n'est pas au programme du lycée: $\int_0^{60} \sqrt{1 + f'^2(x)}\, dx$



Cependant en approchant la courbe par des segments, une démarche algorithmique va permettre de résoudre le problème et trouver une valeur approchée aussi précise qu'on veut du résultat.

Il va suffire de calculer de manière itérative les longueurs des segments et de les sommer au fur et à mesure.

Région académique OCCITANIE
académie Toulouse

L'algorithmique dans la résolution de problème en Seconde

année des mathématiques 2019 2020

En seconde en plus des notions d'algorithmique et programmation du cycle 4, les élèves vont disposer de la notion de fonction informatique.
C'est d'ailleurs un point commun aux deux disciplines à mettre en exergue et à comparer.

La décomposition du problème initial en sous problèmes dont chacun sera résolu à l'aide d'une fonction permet de répondre efficacement au problème posé.

Une fonction pour calculer les ordonnées des points de la courbe de f, une fonction pour calculer la longueur d'un segment et enfin une fonction utilisant les deux précédentes pour itérer sur le nombre de segments choisi et sommer les longueurs.

Liberté • Égalité • Fraternité
RÉPUBLIQUE FRANÇAISE
académie Toulouse
Région académique
OCCITANIE

L'algorithmique dans la résolution de problème en Seconde

année des mathématiques 2019 2020

```
from math import *

def f(x):
    return 0.0125*x**2-x+45

def distance(x,y,u,v):
    return sqrt((u-x)**2+(v-y)**2)

def longueur_courbe(a,b,n):
    long_ligne_brisee=0
    pas=(b-a)/n
    for i in range(n):
        long_segment=distance(a+i*pas,f(a+i*pas),a+(i+1)*pas,f(a+(i+1)*pas))
        long_ligne_brisee=long_ligne_brisee+long_segment
    return long_ligne_brisee
```

```
>>> longueur_courbe(0,80,4)
91.2310562561766
>>>
```

Région académique OCCITANIE

académie Toulouse

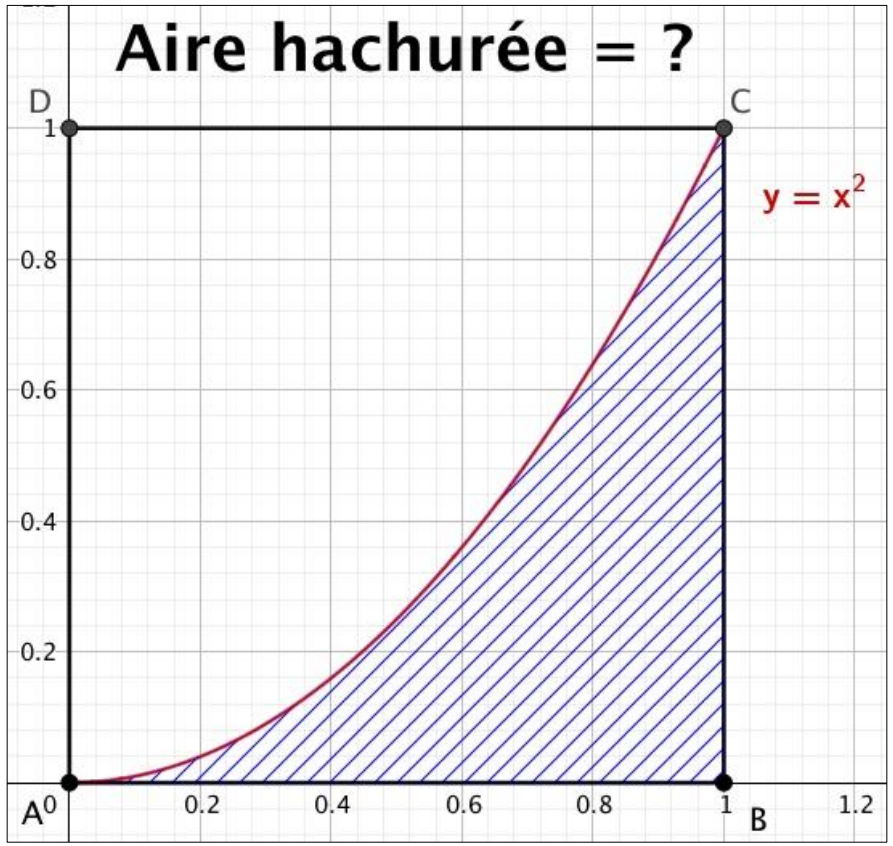
L'algorithmique dans la résolution de problème en EDS voie générale

année des mathématiques 2019 2020

Évaluer une aire par la méthode de Monte-Carlo

L'évaluation de l'aire d'une surface plane entre l'axe des abscisse et la courbe d'une fonction continue sur un un intervalle borné [a;b] ne sera accessible mathématiquement via l'intégrale qu'en terminale.

Si l'on admet le résultat de la méthode de Monte-Carlo cela devient possible en valeur approchée à l'aide d'un algorithme de simulation.



Région académique OCCITANIE

académie Toulouse

L'algorithmique dans la résolution de problème en EDS voie générale

année des mathématiques 2019 2020

Méthode de Monte-Carlo

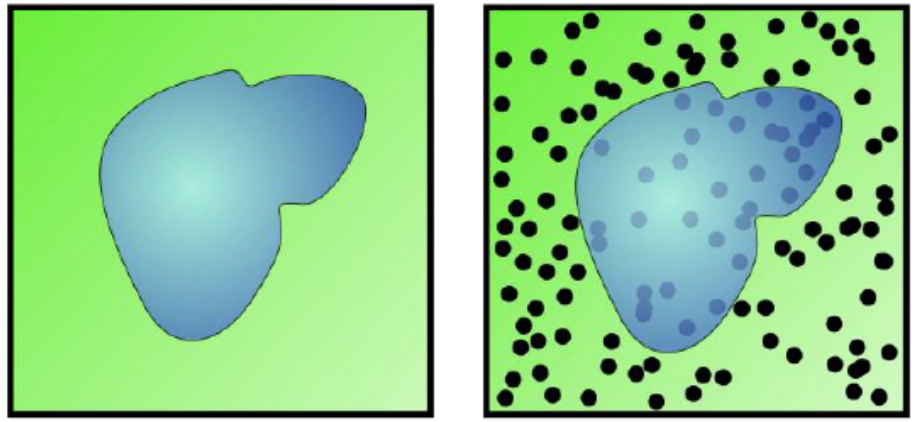
On connait l'aire du rectangle et l'on souhaite estimer la superficie de la zone bleue.

On tire aléatoirement un très grand nombre de points N dans le rectangle. On compte le nombre de points n qui appartiennent à la zone bleue.

Pour un très grand nombre de lancers, on a

$$\frac{n}{N} \cong \frac{aire\ zone\ bleue}{aire\ du\ rectangle}.$$

Ainsi $aire\ zone\ bleue \cong \frac{n}{N} \times aire\ rectangle$



Région académique OCCITANIE

académie Toulouse

L'algorithmique dans la résolution de problème en EDS voie générale

année des mathématiques 2019 2020

Réponse au problème avec la méthode de Monte-Carlo

On admet qu'un point « tiré » au hasard dans le carré ABCD va se trouver dans la zone hachurée avec une probabilité égale au quotient de l'aire hachurée par l'aire du carré ABCD. Dans notre cas particulier puisque l'aire du carré ABCD vaut 1, la probabilité que le point tiré au hasard se trouve dans l'aire hachurée est égale à l'aire hachurée.

Un algorithme de simulation permet de tirer au hasard n points dans le carré ABCD et de déterminer la fréquence du nombre de points « tombés » dans la zone hachurée.
La méthode de Monte-Carlo et la loi des grands nombres permettent d'affirmer que plus n sera grand plus cette fréquence sera proche de la probabilité et donc de l'aire hachurée.

Liberté • Égalité • Fraternité
RÉPUBLIQUE FRANÇAISE
académie Toulouse
Région académique
OCCITANIE

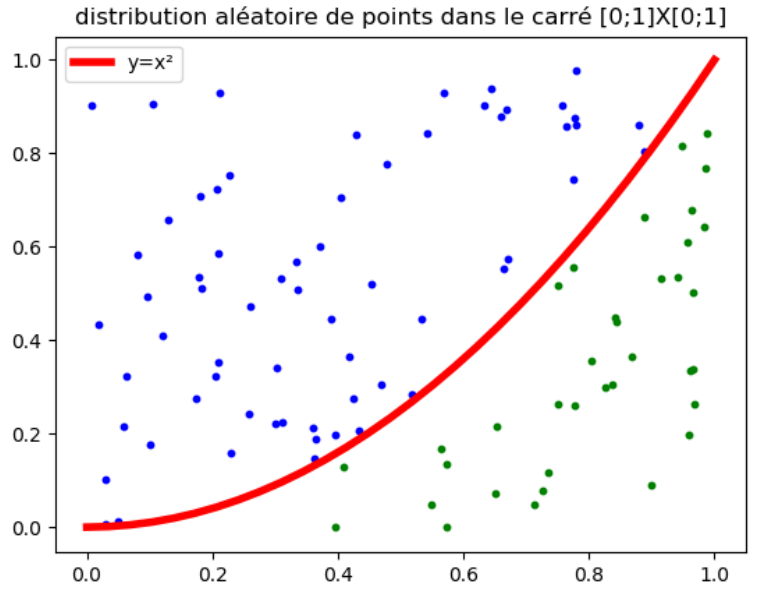
L'algorithmique dans la résolution de problème en EDS voie générale

année des mathématiques 2019 2020

On génère de manière aléatoire n abscisses et ordonnées de points dans l'intervalle [0;1] à l'aide d'une boucle.

On teste pour chaque point si son ordonnée est inférieure ou égale au carré de son abscisse et on compte le nombre de points pour lesquels c'est le cas.

On détermine alors la fréquence de points qui sont dans la zone hachurée … proche de la valeur de l'aire recherchée.



35 points sur 100 sont dans la zone sous la courbe dont on cherche l'aire

Région académique OCCITANIE
académie Toulouse

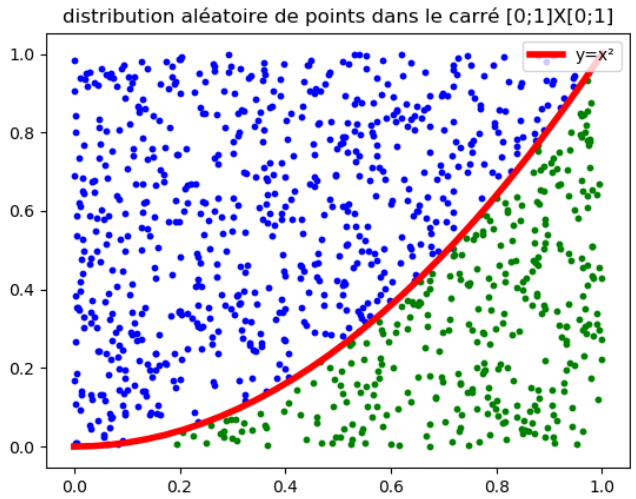
L'algorithmique dans la résolution de problème en EDS voie générale

année des mathématiques 2019 2020

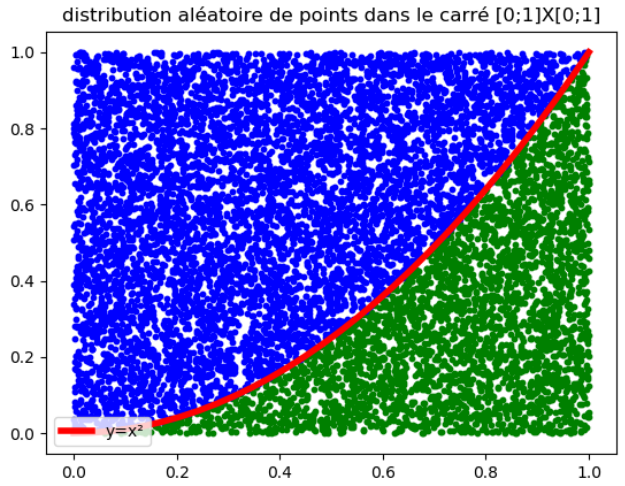
```python
import numpy as np
import matplotlib.pyplot as plt
import random

def MonteCarloParabole(N):
    compteur=0
    XV=[]
    YV=[]
    XB=[]
    YB=[]
    for i in range(N):
        x=random.random()
        y=random.random()
        if y<=x**2:
            XV.append(x)
            YV.append(y)
            compteur+=1
        else:
            XB.append(x)
            YB.append(y)
    FreqSousCourbe=compteur/N
    resultat=FreqSousCourbe
    plt.scatter(XV,YV,s=10,c='g')
    plt.scatter(XB,YB,s=10,c='b')
    x = np.linspace(0, 1, 30)
    y = x**2
    plt.plot(x, y,c='r',label="y=x²",linewidth=4)
    plt.title("distribution aléatoire de points dans le carré [0;1]X[0;1]")
    plt.legend()
    return resultat
```

```
>>> MonteCarloParabole(1000)
0.335
>>> plt.show()
```




```
>>> MonteCarloParabole(10000)
0.3324
>>> plt.show()
```



Région académique OCCITANIE

académie Toulouse

Les mathématiques au service de l'algorithmique

année des mathématiques 2019 2020

- Algorithmique dans le programme de NSI
- Algorithme de recherche d'un élément dans une liste
- Complexité
- Correction
- Terminaison

Région académique
OCCITANIE
académie Toulouse

Les mathématiques au service de l'algorithmique

année des mathématiques 2019 2020

Algorithmique

Le concept de méthode algorithmique est introduit ; de nouveaux exemples seront vus en terminale. Quelques algorithmes classiques sont étudiés. L'étude de leurs coûts respectifs prend tout son sens dans le cas de données nombreuses, qui peuvent être préférentiellement des données ouvertes.

Il est nécessaire de montrer l'intérêt de prouver la correction d'un algorithme pour lequel on dispose d'une spécification précise, notamment en mobilisant la notion d'invariant sur des exemples simples. La nécessité de prouver la terminaison d'un programme est mise en évidence dès qu'on utilise une boucle non bornée (ou, en terminale, des fonctions récursives) grâce à la mobilisation de la notion de variant sur des exemples simples.

Région académique OCCITANIE
académie Toulouse

Les mathématiques au service de l'algorithmique

année des mathématiques 2019 2020

On considère le jeu suivant.

L'ordinateur choisit un nombre entier au hasard compris entre 1 et 100.
L'utilisateur a alors 10 essais possibles au maximum pour retrouver le nombre choisi par l'ordinateur.
A chaque essai, l'ordinateur indique au joueur si la proposition est supérieure ou inférieure au nombre à deviner.

Quelle est la stratégie gagnante pour l'utilisateur ?

Région académique OCCITANIE
académie Toulouse

Les mathématiques au service de l'algorithmique

année des mathématiques 2019 2020

Une Stratégie gagnante :

Nombre à deviner : 63

ESSAI	Proposition	Plage
1	50	[50..100]
2	75	[50..75]
3	57	[57..75]
4	61	[61..75]
5	68	[61..68]
6	64	[61..64]
7	62	[62..64]
8	63	GAGNE

A chaque étape, l'amplitude de la zone de recherche est divisé par 2 et $\frac{100}{2^{10}} < 1$!!!

Région académique
OCCITANIE

Les mathématiques au service de l'algorithmique

année des mathématiques 2019 2020

Recherche « naïve » dans une liste ordonnée.

```
def recherche(liste,elt):
    indice=-1
    for i in range(len(liste)):
        if liste[i]==elt :
            indice=i
    return indice
```

Recherche à l'aide de l'algorithme de dichotomie dans une liste ordonnée

```
def recherche_dico(liste,elt):
    indice_debut=0
    indice_fin=len(liste)-1
    trouve=False
    indice=-1
    while (not(trouve) and indice_fin-indice_debut>=0):
        indice_milieu=(indice_debut+indice_fin)//2
        if (liste[indice_milieu]==elt):
            trouve=True
            indice=indice_milieu
        if (liste[indice_milieu]>elt):
            indice_fin=indice_milieu-1
        else :
            indice_debut=indice_milieu+1
    return indice
```

Les mathématiques au service de l'algorithmique

La notion de **complexité algorithmique** est une problématique de l'informatique fortement liée aux mathématiques.

Définition : La complexité algorithmique consiste à « mesurer » la performance d'un algorithme afin d'obtenir un résultat.

Un programme sera plus performant par rapport à un autre qui réalise la même tâche, si sa complexité algorithmique est plus faible.

Liberté • Égalité • Fraternité
RÉPUBLIQUE FRANÇAISE
académie Toulouse
Région académique
OCCITANIE

Les mathématiques au service de l'algorithmique

année des mathématiques 2019 2020

Complexité de la recherche « naïve » pour une liste à n éléments

```
def recherche(liste,elt):
    indice=-1
    for i in range(len(liste)):
        if liste[i]==elt :
            indice=i
    return indice
```

Pour calculer la complexité, on considère qu'une affectation, qu'un calcul, qu'une comparaison, qu'un affichage, qu'une affectation a un coût de 1.

Quel est la complexité de cet algorithme ?

Région académique OCCITANIE
académie Toulouse

Les mathématiques au service de l'algorithmique

année des mathématiques 2019 2020

Complexité de la recherche « naïve » pour une liste à n éléments

```
def recherche(liste,elt):
    indice=-1
    for i in range(len(liste)):
        if liste[i]==elt :
            indice=i
    return indice
```

On simplifie le calcul :

- Affectation d'indice : 1
- Nbre de comparaison dans la boucle pour : n
- Une affectation si elt est dans la liste : 1

- Bilan : Coût n+1 ou n+2
- **On dit que c'est un coût linéaire, c'est-à-dire majoré par K.n**

Région académique OCCITANIE
académie Toulouse

Les mathématiques au service de l'algorithmique

année des mathématiques 2020 2019

Complexité de la recherche dichotomique pour une liste à n éléments

```python
def recherche_dico(liste,elt):
    indice_debut=0
    indice_fin=len(liste)-1
    trouve=False
    indice=-1
    while (not(trouve) and indice_fin-indice_debut>=0):
        indice_milieu=(indice_debut+indice_fin)//2
        if (liste[indice_milieu]==elt):
            trouve=True
            indice=indice_milieu
        if (liste[indice_milieu]>elt):
            indice_fin=indice_milieu-1
        else :
            indice_debut=indice_milieu+1
    return indice
```

Quel est la complexité de cet algorithme ?

Combien de fois **dans le pire des cas** va-t-on répéter les instructions présentes dans la boucle « while » ?

Si l'on note N le nombre de répétition, N est le plus petit entier qui vérifie :

$$\frac{n}{2^N} < 1 \Leftrightarrow n < 2^N \Leftrightarrow \log_2 n < N$$

Donc $N = \lfloor \log_2(n) \rfloor + 1$

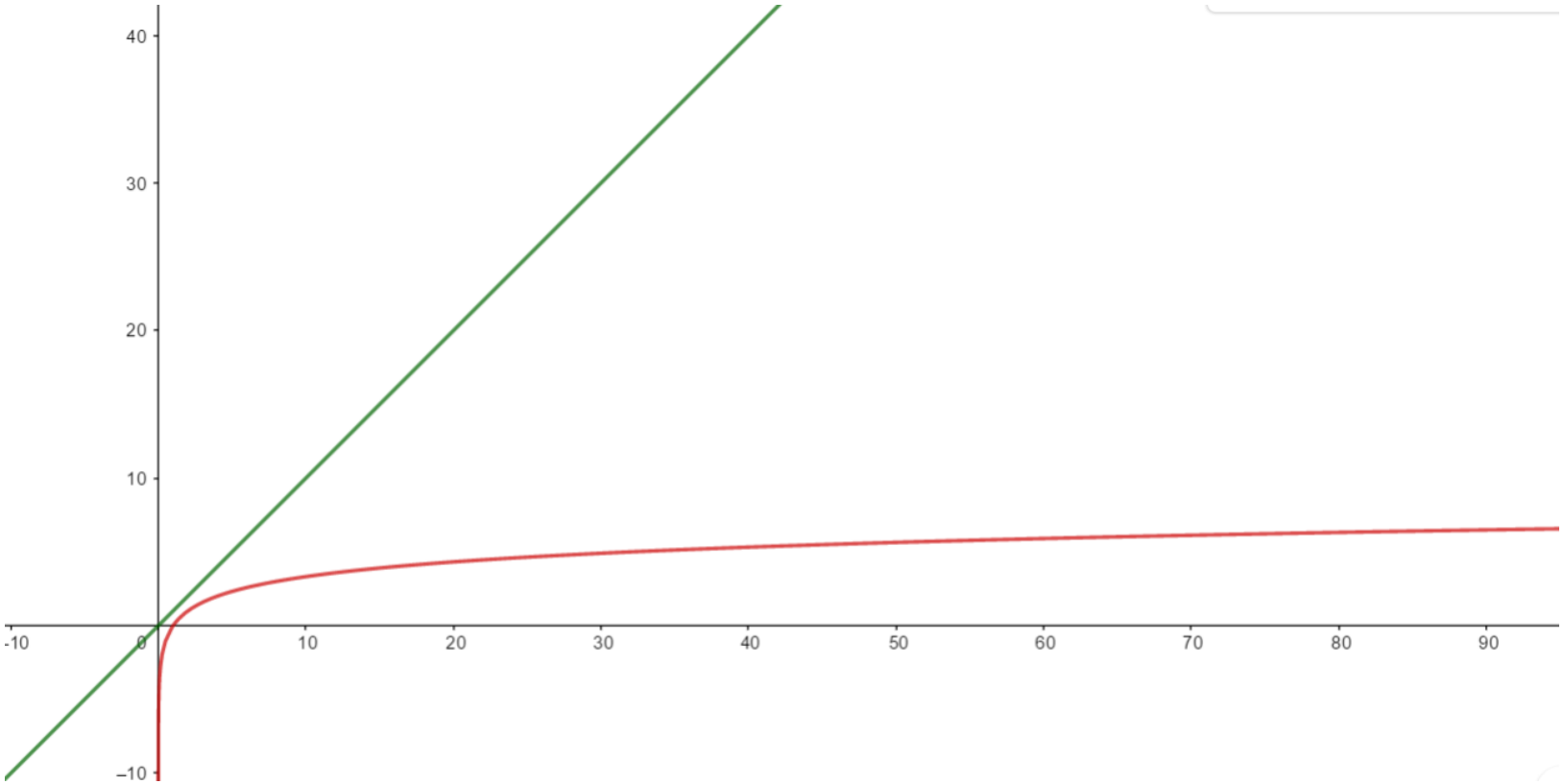
Région académique OCCITANIE

académie Toulouse

Les mathématiques au service de l'algorithmique

année des mathématiques 2019 2020

Comparaison des complexités des deux algorithmes



Région académique OCCITANIE

académie Toulouse

Les mathématiques au service de l'algorithmique

année des mathématiques 2019 2020

Autres enjeux : Correction d'un algorithme / Terminaison d'un algorithme

- **Est-ce que l'algorithme que je viens d'écrire répond bien au problème posé dans TOUS les cas de figure , autrement dit, est-il CORRECT ?**

- **Est-ce qu'il se termine (boucle infini éventuelle) dans TOUS les cas?**

Région académique
OCCITANIE
académie Toulouse

Les mathématiques au service de l'algorithmique

année des mathématiques 2019 2020

Autres enjeux : Correction d'un algorithme / Terminaison d'un algorithme

- **Une première réponse : METTRE EN ŒUVRE une banque de tests !**

  Exemple : Algo de recherche d'un max dans une liste

```
n=nbre d'éléments de la liste
max=liste[0]
Pour i allant de 1 à (n-1)
        Si liste[i]>liste[i-1] alors max=liste[i]
```

Région académique OCCITANIE
académie Toulouse

Les mathématiques au service de l'algorithmique

Autres enjeux : Correction d'un algorithme / Terminaison d'un algorithme

- **Une première réponse : METTRE EN ŒUVRE une banque de tests !**

Mais est ce suffisant ???

- **Apport des mathématiques pour « prouver » que l'algorithme**
  - **Répond au problème (correction)**
  - **Se termine (terminaison)**

Région académique
OCCITANIE

Les mathématiques au service de l'algorithmique

année des mathématiques 2020 2019

CORRECTION D'UN ALGORITHME – PRINCIPE DE BASE

En général, on identifie une propriété (invariant) telle que

- **Initialisation : Elle est vraie au moment d'entrer dans la boucle**
- **Hérédité : Si elle est vraie au moment de commencer une itération, alors les opérations effectuées pendant l'itération, font qu'elle est encore vraie à la fin de l'itération.**

Liberté • Égalité • Fraternité
RÉPUBLIQUE FRANÇAISE
académie Toulouse
Région académique OCCITANIE

Les mathématiques au service de l'algorithmique

année des mathématiques 2019 2020

CORRECTION D'UN ALGORITHME – Exemple avec la recherche « naïve »

```
def recherche(liste,elt):
    indice=-1
    for i in range(len(liste)):
        if liste[i]==elt :
            indice=i
    return indice
```

Invariant : (indice $\neq -1 \Leftrightarrow$ elt appartient à liste[0..i[)

Remarque : si i=0, liste[0..i[est vide.

Région académique OCCITANIE

académie Toulouse

Les mathématiques au service de l'algorithmique

année des mathématiques 2019 2020

CORRECTION D'UN ALGORITHME – Exemple avec la recherche « naïve »

```
def recherche(liste,elt):
    indice=-1
    for i in range(len(liste)):
        if liste[i]==elt :
            indice=i
    return indice
```

Invariant : (indice $\neq -1 \Leftrightarrow$ elt appartient à liste[0..i[)

A la sortie de la boucle for, i=nbre élément de la liste.
Comme l'invariant est toujours vérifié,
indice $\neq -1 \Leftrightarrow$ elt appartient à liste[0..n[, ce qui prouve formellement le résultat de l'algorithme.

Région académique OCCITANIE
académie Toulouse

Les mathématiques au service de l'algorithmique

année des mathématiques 2019 2020

```python
def recherche_dico(liste,elt):
    indice_debut=0
    indice_fin=len(liste)-1
    trouve=False
    indice=-1
    while (not(trouve) and indice_fin-indice_debut>=0):
        indice_milieu=(indice_debut+indice_fin)//2
        if (liste[indice_milieu]==elt):
            trouve=True
            indice=indice_milieu
        if (liste[indice_milieu]>elt):
            indice_fin=indice_milieu-1
        else :
            indice_debut=indice_milieu+1
    return indice
```

CORRECTION D'UN ALGORITHME – Exemple avec la recherche dichotomique

Invariant de la boucle while
Si elt est présent dans le tableau c'est nécessairement à un indice compris entre indice_début et indice_fin.

Région académique OCCITANIE
académie Toulouse

Les mathématiques au service de l'algorithmique

année des mathématiques 2019 2020

Invariant de la boucle while

Si elt est présent dans le tableau c'est nécessairement à un indice compris entre indice_début et indice_fin.

Cet invariant de boucle montre que si la fonction renvoie (-1) alors elt n'est pas dans la liste.
En effet, la sortie du WHILE assure que Indice_fin – indice_début <0 et l'invariant indique que « si elt est dans le tableau, son indice est entre indice_debut et indice_fin. »

Nécessairement elt n'est pas dans le tableau !

Région académique OCCITANIE
académie Toulouse

Les mathématiques au service de l'algorithmique

année des mathématiques 2019 2020

Terminaison d'une boucle non bornée – PRINCIPE DE BASE

En général, on identifie un VARIANT de boucle. Il s'agit d'une quantité entière qui :

- **Doit être positive ou nulle pour rester dans la boucle**
- **Doit décroitre strictement à chaque itération**

Ce procédé est basé sur la méthode de la descente infinie (mathématiques) : « Toute suite d'entiers naturels strictement décroissante est nécessairement finie. »

Région académique OCCITANIE
académie Toulouse

Les mathématiques au service de l'algorithmique

année des mathématiques 2019 2020

```python
def recherche_dico(liste,elt):
    indice_debut=0
    indice_fin=len(liste)-1
    trouve=False
    indice=-1
    while (not(trouve) and indice_fin-indice_debut>=0):
        indice_milieu=(indice_debut+indice_fin)//2
        if (liste[indice_milieu]==elt):
            trouve=True
            indice=indice_milieu
        if (liste[indice_milieu]>elt):
            indice_fin=indice_milieu-1
        else :
            indice_debut=indice_milieu+1
    return indice
```

Exemple de terminaison

La quantité
indice_fin – indice_début est un variant de la boucle while.

C'est bien une quantité entière positive ou nulle qui décroit strictement.

Région académique
OCCITANIE
académie Toulouse

année des mathématiques 2019 2020

Merci pour votre attention.

Région académique
OCCITANIE

académie Toulouse