

Exemples d'algorithmes dans les programmes.

Table des matières

Le programme de seconde	2
Les nombres	2
Géométrie	2
Fonctions.....	2
Statistiques et probabilités	2
Le programme de SNT	3
Notions transversales de programmation	4
Le web.....	4
Les réseaux sociaux.....	4
Les données structurées et leur traitement.....	5
Localisation, cartographie et mobilité	6
Informatique embarquée et objets connectés	6
Le programme de première	7
Suites numériques, modèles discrets	7
Equations, fonctions polynômes du second degré.....	7
Analyse.....	7
Enseignement commun, première voie technologique	9
Le programme de Terminale, enseignement de spécialité	11
Algèbre et géométrie :	11
Analyse :	11
Probabilités :	11
Le programme de L'option Mathématiques complémentaires (Terminale)	12
Par thèmes d'études :	12
• Modèles définis par une fonction d'une variable :	12
• Modèles d'évolution :	12
• Calculs d'aires :	12
• Répétition d'expériences indépendantes, échantillonnage :	12
• Temps d'attente :	12
Par contenu :	13
• Analyse :	13
.....	13
• Probabilités et statistiques :	13
.....	13
Le programme de l'option Mathématiques expertes (Terminale)	13
Le programme de l'enseignement scientifique (Terminale)	14

Le programme de seconde.

Les nombres

Exemple d'algorithme

- Déterminer par balayage un encadrement de $\sqrt{2}$ d'amplitude inférieure ou égale à 10^{-n} .

Exemples d'algorithme

- Déterminer si un entier naturel a est multiple d'un entier naturel b .
- Pour des entiers a et b donnés, déterminer le plus grand multiple de a inférieur ou égal à b .
- Déterminer si un entier naturel est premier.

Exemple d'algorithme

- Déterminer la première puissance d'un nombre positif donné supérieure ou inférieure à une valeur donnée.

Géométrie

Exemples d'algorithme

- Étudier l'alignement de trois points dans le plan.
- Déterminer une équation de droite passant par deux points donnés.

Fonctions

- Résoudre une équation ou une inéquation du type $f(x) = k$, $f(x) < k$, en choisissant une méthode adaptée : graphique, algébrique, logicielle.

Exemples d'algorithme

- Pour une fonction dont le tableau de variations est donné, algorithmes d'approximation numérique d'un extremum (balayage, dichotomie).
- Algorithme de calcul approché de longueur d'une portion de courbe représentative de fonction.

Statistiques et probabilités

- Pour des données réelles ou issues d'une simulation, lire et comprendre une fonction écrite en Python renvoyant la moyenne m , l'écart type s , et la proportion d'éléments appartenant à $[m - 2s, m + 2s]$.

Expérimentations

- Lire et comprendre une fonction Python renvoyant le nombre ou la fréquence de succès dans un échantillon de taille n pour une expérience aléatoire à deux issues.
- Observer la loi des grands nombres à l'aide d'une simulation sur Python ou tableur.
- Simuler N échantillons de taille n d'une expérience aléatoire à deux issues. Si p est la probabilité d'une issue et f sa fréquence observée dans un échantillon, calculer la proportion des cas où l'écart entre p et f est inférieur ou égal à $1/\sqrt{n}$.

■ Algorithmique et programmation

La démarche algorithmique est, depuis les origines, une composante essentielle de l'activité mathématique. Au cycle 4, en mathématiques et en technologie, les élèves ont appris à écrire, mettre au point et exécuter un programme simple. Une consolidation des acquis du cycle 4 est proposée autour de deux idées essentielles :

- la notion de fonction ;
- la programmation comme production d'un texte dans un langage informatique.

Dans le cadre de cette activité, les élèves s'exercent à :

- décrire des algorithmes en langage naturel ou dans un langage de programmation ;
- en réaliser quelques-uns à l'aide d'un programme simple écrit dans un langage de programmation textuel ;
- interpréter, compléter ou modifier des algorithmes plus complexes.

Un langage de programmation simple d'usage est nécessaire pour l'écriture des programmes informatiques. Le langage choisi est Python, langage interprété, concis, largement répandu et pouvant fonctionner dans une diversité d'environnements. Les élèves sont entraînés à passer du langage naturel à Python et inversement.

L'algorithmique a une place naturelle dans tous les champs des mathématiques et les problèmes ainsi traités doivent être en relation avec les autres parties du programme (fonctions, géométrie, statistiques et probabilité, logique) mais aussi avec les autres disciplines ou la vie courante.

À l'occasion de l'écriture d'algorithmes et de petits programmes, il convient de transmettre aux élèves l'exigence d'exactitude et de rigueur, et de les entraîner aux pratiques systématiques de vérification et de contrôle. En programmant, les élèves revisitent les notions de variables et de fonctions sous une forme différente.

- Variables informatiques de type entier, flottant, chaîne de caractère.
- Affectation (notée \leftarrow en langage naturel).
- Séquence d'instructions.
- Instruction conditionnelle.
- Boucle bornée (for), boucle non bornée (while).

Capacités associées

- Choisir ou déterminer le type d'une variable (entier, flottant ou chaîne de caractères).
- Concevoir et écrire une instruction d'affectation, une séquence d'instructions, une instruction conditionnelle.
- Écrire une formule permettant un calcul combinant des variables.
- Programmer, dans des cas simples, une boucle bornée, une boucle non bornée.
- Dans des cas plus complexes : lire, comprendre, modifier ou compléter un algorithme ou un programme.

■ Notion de fonction

Connaissances

Fonctions à un ou plusieurs arguments.

- Fonction renvoyant un nombre aléatoire. Série statistique obtenue par la répétition de l'appel d'une telle fonction.

Capacités associées

- Écrire des fonctions simples ; lire, comprendre, modifier, compléter des fonctions plus complexes. Appeler une fonction.
- Lire et comprendre une fonction renvoyant une moyenne, un écart type. Aucune connaissance sur les listes n'est exigée.
- Écrire des fonctions renvoyant le résultat numérique d'une expérience aléatoire, d'une répétition d'expériences aléatoires indépendantes.

Contenus	Capacités attendues
Affectations, variables Séquences Instructions conditionnelles Boucles bornées et non bornées Définitions et appels de fonctions	Écrire, exécuter et mettre au point un programme.
Exemples d'activités	
– Illustrer ces notions par des activités liées aux différents thèmes du programme.	

Au collège (cycle 4), les élèves ont découvert et pratiqué les éléments fondamentaux d'algorithmique et de programmation. Le programme de seconde de mathématiques approfondit l'apprentissage de la programmation. Une coordination avec le cours de mathématiques est donc nécessaire pour déterminer à quel moment des éléments de programmation peuvent être utilisés en sciences numériques et technologie.

Le web

- Calculer la popularité d'une page à l'aide d'un graphe simple puis programmer l'algorithme.

Les réseaux sociaux

Les algorithmes et les programmes

De très nombreux algorithmes sont mis en œuvre par les applications de réseautage social.

Toutes les applications s'appuient sur des services de mise en relation avec des internautes membres du réseau, relations ou amis communs : des algorithmes opérant sur les **graphes** et sur les bases de données sont au cœur de ces services.

À l'aide d'algorithmes de recommandation, les réseaux sociaux suggèrent aux utilisateurs des amis, des contenus, des annonces promotionnelles. Ils permettent aussi aux plateformes sociales d'étudier les comportements de leurs utilisateurs à des fins commerciales, politiques ou d'amélioration du service.

Rayon, diamètre et centre d'un graphe	Déterminer ces caractéristiques sur des graphes simples.
---------------------------------------	--

- Construire ou utiliser une représentation du graphe des relations d'un utilisateur. S'appuyer sur la densité des liens pour identifier des groupes, des communautés.
- Sur des exemples de graphes simples, en informatique débranchée, étudier les notions de rayon, diamètre et centre d'un graphe, de manière à illustrer la notion de « petit monde ».

Les données structurées et leur traitement.

Les algorithmes et les programmes

La recherche dans des **données structurées** a d'abord été effectuée selon une indexation préalable faite par l'homme. Des algorithmes ont ensuite permis d'automatiser l'indexation à partir de textes, d'images ou de sons.

Une table de données peut faire l'objet de différentes opérations : rechercher une information précise dans la collection, trier la collection sur une ou plusieurs propriétés, filtrer la collection selon un ou plusieurs tests sur les valeurs des descripteurs, effectuer des calculs, mettre en forme les informations produites pour une visualisation par les utilisateurs.

Pour assurer la persistance des données, ces dernières sont stockées dans des fichiers. Le format CSV (*Comma Separated Values*, les données avec des séparateurs) est un format de fichier simple permettant d'enregistrer une table. À tout fichier sont associées des **métadonnées** qui permettent d'en décrire le contenu. Ces métadonnées varient selon le type de fichier (date et coordonnées de géolocalisation d'une photographie, auteur et titre d'un fichier texte, etc.).

- Consulter les métadonnées de fichiers correspondant à des informations différentes et repérer celles collectées par un dispositif et celles renseignées par l'utilisateur.
- Télécharger des données ouvertes (sous forme d'un fichier au format CSV avec les métadonnées associées), observer les différences de traitements possibles selon le logiciel choisi pour lire le fichier : programme Python, tableur, éditeur de textes ou encore outils spécialisés en ligne.
- Explorer les données d'un fichier CSV à l'aide d'opérations de tri et de filtre, effectuer des calculs sur ces données, réaliser une visualisation graphique des données.

Les algorithmes et les programmes

Les algorithmes cartographiques concernent principalement l'affichage sélectif d'informations variées et le **calcul d'itinéraires**. L'affichage est paramétré par les informations à montrer, que l'on peut choisir par simples clics. Une difficulté est liée au mélange d'informations de types différents lors des changements d'échelle : les graphismes peuvent être très différents et beaucoup d'informations doivent être supprimées pour les grandes échelles, mais une route doit être représentée avec à peu près la même largeur, quelle que soit l'échelle.

Les récepteurs GPS fournissent la localisation sous une forme normalisée facilement décodable, par exemple selon le **protocole NMEA 0183** (*National Marine Electronics Association*), ou directement dans les métadonnées EXIF d'une photo. La localisation et les cartes se couplent dans le suivi permanent de la position sur la carte ou sur un itinéraire précalculé.

Calculs d'itinéraires	Utiliser un logiciel pour calculer un itinéraire. Représenter un calcul d'itinéraire comme un problème sur un graphe.
-----------------------	--

Informatique embarquée et objets connectés

Contenus	Capacités attendues
Systèmes informatiques embarqués	Identifier des algorithmes de contrôle des comportements physiques à travers les données des capteurs, l'IHM et les actions des actionneurs dans des systèmes courants.
Interface homme-machine (IHM)	Réaliser une IHM simple d'un objet connecté.
Commande d'un actionneur, acquisition des données d'un capteur	Écrire des programmes simples d'acquisition de données ou de commande d'un actionneur.
Exemples d'activités	
<ul style="list-style-type: none">— Identifier les évolutions apportées par les algorithmes au contrôle des freins et du moteur d'une automobile, ou à celui de l'assistance au pédalage d'un vélo électrique.— Réaliser une IHM pouvant piloter deux ou trois actionneurs et acquérir les données d'un ou deux capteurs.— Gérer des entrées/sorties à travers les ports utilisés par le système.— Utiliser un tableau de correspondance entre caractères envoyés ou reçus et commandes physiques (exemple : le moteur A est piloté à 50 % de sa vitesse maximale lorsque le robot reçoit la chaîne de caractères « A50 »).	

Le programme de première

Suites numériques, modèles discrets

Exemples d'algorithme

- Calcul de termes d'une suite, de sommes de termes, de seuil.
- Calcul de factorielle.
- Liste des premiers termes d'une suite : suites de Syracuse, suite de Fibonacci.

Equations, fonctions polynômes du second degré

Exemple d'algorithme

- Méthode de Héron pour l'approximation numérique des racines en utilisant la forme canonique.

Analyse

Exemple d'algorithme

- Écrire la liste des coefficients directeurs des sécantes pour un pas donné.

Exemple d'algorithme

- Méthode de Newton, en se limitant à des cas favorables.

Exemple d'algorithme

- Construction de l'exponentielle par la méthode d'Euler. Détermination d'une valeur approchée de e à l'aide de la suite $\left(1 + \frac{1}{n}\right)^n$.

Exemple d'algorithme

- Approximation de π par la méthode d'Archimède.

Exemple d'algorithme

- Méthode de Monte Carlo : estimation de l'aire sous la parabole, estimation du nombre π .

Exemples d'algorithmes

- Algorithme renvoyant l'espérance, la variance ou l'écart type d'une variable aléatoire.
- Fréquence d'apparition des lettres d'un texte donné, en français, en anglais.

Expérimentations

Le travail expérimental de simulation d'échantillons prolonge celui entrepris en seconde. L'objectif est de faire percevoir le principe de l'estimation de l'espérance d'une variable aléatoire, ou de la moyenne d'une variable statistique dans une population, par une moyenne observée sur un échantillon.

- Simuler une variable aléatoire avec Python.
- Lire, comprendre et écrire une fonction Python renvoyant la moyenne d'un échantillon de taille n d'une variable aléatoire.
- Étudier sur des exemples la distance entre la moyenne d'un échantillon simulé de taille n d'une variable aléatoire et l'espérance de cette variable aléatoire.
- Simuler, avec Python ou un tableur, N échantillons de taille n d'une variable aléatoire, d'espérance μ et d'écart type σ . Si m désigne la moyenne d'un échantillon, calculer la proportion des cas où l'écart entre m et μ est inférieur ou égal à $2\sigma/\sqrt{n}$.

L'accent est mis sur la programmation modulaire qui permet de découper une tâche complexe en tâches plus simples.

■ **Notion de liste**

La génération des listes en compréhension et en extension est mise en lien avec la notion d'ensemble. Les conditions apparaissant dans les listes définies en compréhension permettent de travailler la logique. Afin d'éviter des confusions, on se limite aux listes sans présenter d'autres types de collections.

Capacités associées

- Générer une liste (en extension, par ajouts successifs ou en compréhension).
- Manipuler des éléments d'une liste (ajouter, supprimer ...) et leurs indices.
- Parcourir une liste.
- Itérer sur les éléments d'une liste.

Enseignement commun, première voie technologique

Tout au long du cycle terminal, les élèves sont amenés à :

- écrire une fonction simple en langage Python ;
- interpréter un algorithme donné ;
- compléter, améliorer ou corriger un programme informatique ;
- traduire un algorithme en langage naturel ou en langage Python ;
- décomposer un programme en fonctions ;
- organiser une feuille de calcul.

Capacités attendues

■ Variables :

- utiliser un générateur de nombres aléatoires entre 0 et 1 pour simuler une loi de Bernoulli de paramètre p ;
- utiliser la notion de compteur ;
- utiliser le principe d'accumulateur pour calculer une somme, un produit.

■ Fonctions :

- identifier les entrées et les sorties d'une fonction ;
- structurer un programme en ayant recours aux fonctions.

■ Listes :

- générer une liste (en extension, par ajouts successifs, en compréhension) ;
- manipuler des éléments d'une liste (ajouter, supprimer ...) et leurs indices ;
- itérer sur les éléments d'une liste.

■ Sélection de données :

- traiter un fichier contenant des données réelles pour en extraire de l'information et l'analyser ;
- réaliser un tableau croisé de données sur deux critères à partir de données brutes.

Commentaires

- Les notions relatives aux types de variables et à l'affectation sont consolidées. Comme en classe de seconde, on utilise le symbole « \leftarrow » pour désigner l'affectation dans un algorithme écrit en langage naturel.
- L'accent est mis sur la programmation modulaire qui permet de découper une tâche complexe en tâches plus simples.
- La génération des listes en compréhension et en extension est mise en lien avec la notion d'ensemble. Les conditions apparaissant dans les listes définies en compréhension permettent de travailler la logique.
- Afin d'éviter des confusions, il est recommandé de se limiter aux listes sans présenter d'autres types de collections.

Situations algorithmiques (sauf série STD2A)

- Calculer un terme de rang donné d'une suite, une somme finie de termes.
- Déterminer une liste de termes d'une suite et les représenter.
- Déterminer le rang à partir duquel les termes d'une suite sont supérieurs ou inférieurs à un seuil donné, ou aux termes de même rang d'une autre suite.

Situations algorithmiques (sauf série STD2A)

- Calculer une valeur approchée d'une solution d'une équation par balayage.
- Les élèves travaillent avec des données réelles dans des domaines variés (sécurité routière, démographie, économie, agronomie ...).
- Au moins un traitement statistique de fichier de données individuelles anonymes est proposé, issu par exemple du web (OpenData ...).

Situations algorithmiques (sauf série STD2A)

- À partir de deux listes représentant deux caractères d'individus, déterminer un sous-ensemble d'individus répondant à un critère (filtre, utilisation des ET, OU, NON).
- Dresser le tableau croisé de deux variables catégorielles à partir du fichier des individus et calculer des fréquences conditionnelles ou marginales.

Situations algorithmiques (sauf série STD2A)

- Simuler des échantillons de taille n d'une loi de Bernoulli à partir d'un générateur de nombres aléatoires entre 0 et 1.
- Représenter par un histogramme ou par un nuage de points les fréquences observées des 1 dans N échantillons de taille n d'une loi de Bernoulli.
- Compter le nombre de valeurs situées dans un intervalle de la forme $[p - ks; p + ks]$ pour $k \in \{1; 2; 3\}$.

Le programme de Terminale, enseignement de spécialité

Algèbre et géométrie :

- Pour un entier n donné, génération de la liste des coefficients $\binom{n}{k}$ à l'aide de la relation de Pascal.
- Génération des permutations d'un ensemble fini, ou tirage aléatoire d'une permutation.
- Génération des parties à 2, 3 éléments d'un ensemble fini.

Analyse :

- Recherche de seuils.
- Recherche de valeurs approchées de π , e , $\sqrt{2}$, $\frac{1+\sqrt{5}}{2}$, $\ln(2)$, etc.
- Méthode de dichotomie.
- Méthode de Newton, méthode de la sécante.
- Algorithme de Briggs pour le calcul du logarithme.
- Résolution par la méthode d'Euler de $y' = f$, de $y' = ay + b$.
- Méthodes des rectangles, des milieux, des trapèzes.
- Méthode de Monte-Carlo.
- Algorithme de Brouncker pour le calcul de $\ln(2)$.

Probabilités :

- Simulation de la planche de Galton.
- Problème de la surréservation. Étant donné une variable aléatoire binomiale X et un réel strictement positif α , détermination du plus petit entier k tel que $P(X > k) \leq \alpha$.
- Simulation d'un échantillon d'une variable aléatoire.
- Calculer la probabilité de $(|S_n - pn| > \sqrt{n})$, où S_n est une variable aléatoire qui suit une loi binomiale $\mathcal{B}(n,p)$. Comparer avec l'inégalité de Bienaymé-Tchebychev.
- Simulation d'une marche aléatoire.
- Simuler N échantillons de taille n d'une variable aléatoire d'espérance μ et d'écart type σ . Calculer l'écart type s de la série des moyennes des échantillons observés, à comparer à σ/\sqrt{n} . Calculer la proportion des échantillons pour lesquels l'écart entre la moyenne et μ est inférieur ou égal à ks , ou à $k\sigma/\sqrt{n}$, pour $k = 1, 2, 3$.

Algorithmique et programmation

La démarche algorithmique est, depuis les origines, une composante essentielle de l'activité mathématique. Au collège, en mathématiques et en technologie, les élèves ont appris à écrire, mettre au point et exécuter un programme simple. Les classes de seconde et de première ont permis de consolider les acquis du collège (notion de variable, type, de variables, affectation, instruction conditionnelle, boucle notamment), d'introduire et d'utiliser la notion de fonction informatique et de liste. En algorithmique et programmation, le programme reprend les programmes de seconde et de première sans introduire de notion nouvelle, afin de consolider le travail des classes précédentes.

Les algorithmes peuvent être écrits en langage naturel ou utiliser le langage Python. On utilise le symbole « ← » pour désigner l'affectation dans un algorithme écrit en langage naturel. L'accent est mis sur la programmation modulaire qui permet de découper une tâche complexe en tâches plus simples.

L'algorithmique trouve naturellement sa place dans toutes les parties du programme et aide à la compréhension et à la construction des notions mathématiques.

Le programme de L'option Mathématiques complémentaires (Terminale)

Par thèmes d'études :

- Modèles définis par une fonction d'une variable :
 - Résolution d'équations par balayage, par dichotomie.
- Modèles d'évolution :
 - Calcul des termes d'une suite.
 - Recherche de seuils.
 - Méthode d'Euler.
- Calculs d'aires :
 - Calcul d'un terme de rang donné d'une suite.
 - Recherche d'une valeur approchée de précision donnée.
- Répétition d'expériences indépendantes, échantillonnage :
 - Dans le cadre de la loi binomiale : calcul de coefficients binomiaux (triangle de Pascal), de probabilités ; détermination d'un intervalle I pour lequel la probabilité $P(X \in I)$ est inférieure à une valeur donnée α , ou supérieure à $1 - \alpha$.
 - Simulation avec Python d'une variable aléatoire (de la loi de Bernoulli, d'une loi uniforme discrète, etc.) d'un échantillon de taille n d'une variable aléatoire. Fonction Python renvoyant une moyenne pour un échantillon. Série des moyennes pour N échantillons de taille n d'une variable aléatoire d'espérance μ et d'écart type σ . Calcul de l'écart type s de la série des moyennes des échantillons observés, à comparer à σ/\sqrt{n} . Calcul de la proportion des cas où l'écart entre la moyenne m et μ est inférieur ou égal à $k\sigma/\sqrt{n}$ ou à ks , pour $k = 2$ ou $k = 3$.
- Temps d'attente :
 - Simulation d'une variable aléatoire de loi géométrique à partir du schéma de Bernoulli.
 - Simulation d'une loi exponentielle à partir d'une loi uniforme.
 - Demi-vie d'un échantillon de grande taille d'atomes radioactifs.

Par contenu :

- Analyse :

- Recherche de seuils.
- Pour une suite récurrente $u_{n+1} = f(u_n)$, calcul des termes successifs.
- Recherche de valeurs approchées de constantes mathématiques, par exemple π , $\ln 2$, $\sqrt{2}$.
- Méthodes de recherche de valeurs approchées d'une solution d'équation du type $f(x) = k$: balayage, dichotomie, méthode de Newton.
- Algorithme de Briggs pour le calcul de logarithmes.

- Sur des exemples, résolution approchée d'une équation différentielle par la méthode d'Euler.
- Méthode des rectangles, des trapèzes.
- Méthode de Monte-Carlo pour un calcul d'aire.

- Probabilités et statistiques :
 - Simulation d'une variable de Bernoulli ou d'un lancer de dé (ou d'une variable uniforme sur un ensemble fini) à partir d'une variable aléatoire de loi uniforme sur $[0, 1]$.
 - Simulation du comportement de la somme de n variables aléatoires indépendantes et de même loi.

Algorithmique et programmation

La démarche algorithmique est, depuis les origines, une composante essentielle de l'activité mathématique. Au collège, en mathématiques et en technologie, les élèves ont appris à écrire, mettre au point et exécuter un programme simple. Les classes de seconde et de première ont permis de consolider les acquis du collège (notion de variable, type, de variables, affectation, instruction conditionnelle, boucle notamment), d'introduire et d'utiliser la notion de fonction informatique et de liste. En algorithmique et programmation, le programme de mathématiques complémentaires reprend les programmes des classes de

seconde et de première sans introduire de notion nouvelle, afin de consolider le travail des classes précédentes.

Les algorithmes peuvent être écrits en langage naturel ou utiliser le langage Python. On utilise le symbole « \leftarrow » pour désigner l'affectation dans un algorithme écrit en langage naturel. L'accent est mis sur la programmation modulaire qui permet de découper une tâche complexe en tâches plus simples.

L'algorithmique trouve naturellement sa place dans toutes les parties du programme et aide à la compréhension et à la construction des notions mathématiques.

Le programme de l'option Mathématiques expertes (Terminale)

- Algorithme d'Euclide de calcul du PGCD de deux nombres et calcul d'un couple de Bézout.
- Crible d'Ératosthène.
- Décomposition en facteurs premiers.

Le programme de l'enseignement scientifique (Terminale)

• Un usage explicité des outils numériques

Des outils numériques variés trouvent des applications dans le cadre de l'enseignement scientifique : logiciels de calcul ou de simulation, environnements de programmation, logiciels tableurs, etc. Il convient d'associer leur utilisation par les élèves à la compréhension au moins élémentaire de leur nature et de leur fonctionnement.

3.1 La biodiversité et son évolution

<p>Au cours de l'évolution biologique, la composition génétique des populations d'une espèce change de génération en génération.</p> <p>Le modèle mathématique de Hardy-Weinberg utilise la théorie des probabilités pour décrire le phénomène aléatoire de transmission des allèles dans une population. En assimilant les probabilités à des fréquences pour des effectifs de grande taille (loi des grands nombres), le modèle prédit que la structure génétique d'une population de grand effectif est stable d'une génération à l'autre sous certaines conditions (absence de migration, de mutation et de sélection). Cette stabilité théorique est connue sous le nom d'équilibre de Hardy-Weinberg.</p> <p>Les écarts entre les fréquences observées sur une population naturelle et les résultats du modèle s'expliquent notamment par les effets de forces évolutives (mutation, sélection, dérive, etc.).</p>	<p>Pour la transmission de deux allèles dans le cadre du modèle de Hardy-Weinberg, établir les relations entre les probabilités des génotypes d'une génération et celles de la génération précédente.</p> <p>Produire une démonstration mathématique ou un calcul sur tableur ou un programme en Python pour prouver ou constater que les probabilités des génotypes sont constantes à partir de la seconde génération (modèle de Hardy-Weinberg).</p> <p>Utiliser des logiciels de simulation basés sur ce modèle mathématique.</p> <p>Analyser une situation d'évolution biologique expliquant un écart par rapport au modèle de Hardy-Weinberg.</p>
--	--

3.5 L'intelligence artificielle

<p>Un programme peut comporter jusqu'à plusieurs centaines de millions de lignes de code, ce qui rend très probable la présence d'erreurs appelées bogues (ou <i>bugs</i>). Ces erreurs peuvent conduire un programme à avoir un comportement inattendu et entraîner des conséquences graves.</p>	<p>Étant donné un programme très simple, proposer des jeux de données d'entrée permettant d'en tester toutes les lignes.</p> <p>Corriger un algorithme ou un programme bogué simple.</p>
---	--