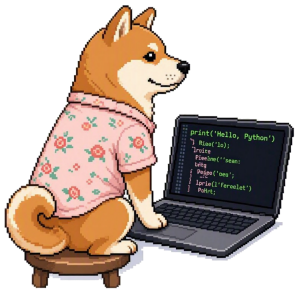
Informatique PT. 2026 - 2027

Mathématiques & Informatique - F. Gaunard
`http://frederic.gaunard.com`
Lycée Voltaire, Paris 11e.

1 Travaux Pratiques : Révisions d'algorithmique



Pour ce premier TP de rentrée, on propose des exercices, principalement issus de l'oral **Math II** qui invitent à manipuler des listes ou des tableaux de manière assez élémentaire.
On rappelle qu'un exercice de ce type doit être, le jour de l'oral, traité en moins de 30 minutes.
On attend donc que soient traités tous les exercices pendant la séance.

Exercice 1. *Nombres en binaire concaténés*, Oral Math II 2026

On note (U_n) la suite de chaînes de caractères qui est construite de sorte à ce que U_{n+1} soit obtenu en concaténant U_n avec le $n-$ième nombre binaire (et vérifiant $U_0 =$ `'0'`).

1. Écrire une fonction d'argument n qui retourne en chaîne de caractères le nombre n en binaire.
2. Écrire une fonction qui donne le $n-$ième terme de la suite.
3. Donner la taille (en nombre de caractères) du terme U_n en fonction de n.
4. Écrire une fonction qui renvoie le nombre d'apparitions du motif `'11'` dans le terme U_n.
5. Déterminer la complexité en temps et en mémoire du calcul de U_n avec l'algorithme proposé à la Question **2.** Proposer une optimisation si nécessaire.

Exercice 2. *Tri*, Oral Math II 2024

1. Écrire une fonction `parcours` d'argument une liste $L = [a_0, a_1, \ldots, a_{n-1}]$ qui modifie cette liste en permutant a_k et a_{k+1} si $a_{k+1} < a_k$, pour k variant de 0 à $n-2$, et qui renvoie le nombre de permutations effectuées.
   Par exemple, si `L = [5, 4, 1, 6]`, alors `parcours(L)` renvoie le nombre 2 et la liste `L` devient `[4, 1, 5, 6]`.

2. Que peut-on dire du dernier élément d'une liste à laquelle on a déjà appliqué une fois la fonction `parcours` ?

3. On veut utiliser cette fonction pour trier par ordre croissant une liste selon le principe suivant : pour une liste L donnée en entrée on répète l'application de la fonction parcours jusqu'à ce que `L` devienne invariante par cette fonction.
   Écrire une fonction `tri` qui ordonne une liste donnée en entrée selon ce principe et qui renvoie le nombre d'itérations effectuées.

4. Expliquer pourquoi la fonction `tri` donne bien le résultat voulu en un temps fini.
   Évaluer son coût de calcul dans le meilleur des cas et dans le pire des cas.

5. Écrire une fonction `tri1` comme une amélioration de la fonction `tri` en évitant les comparaisons inutiles.

Exercice 3. *Trains*, Oral Math II 2025, 2026

1. Écrire une fonction `voeux` qui prend en argument une liste voyageurs (telle que `voyageurs[i]` donne le train j dans lequel il aimerait voyager), et un nombre de trains `rt`.
   La fonction doit renvoyer une liste train telle que `train[i]` donne la liste des voyageurs qui veulent voyager dans ce train, ordonnés de façon croissante.
2. Écrire une fonction `possible1` d'argument une liste voyageurs et une liste `capacite` telle qu'elle renvoie un booléen indiquant s'il est possible ou non de remplir les trains.

Désormais, si le `train[i]` ne peut pas accueillir tous les passagers, les passagers en surplus se dirigent vers le train suivant.

3. Écrire une fonction `possible2` d'argument une liste voyageurs et une liste `capacite` telle que `possible2` renvoie un booléen selon la méthode décrite ci-dessus.
4. Écrire une fonction `affecte` d'argument les listes `voyageurs` et `capacite` qui renvoie une liste `affectation` telle que `affectation[i]` donne la composition du train i.

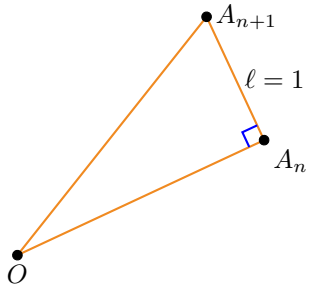
Exercice 4. *Suite de points et nombre de tours*, Oral Math II 2019

Dans le plan affine, muni d'un repère orthonormé direct d'origine O, on considère la suite de points $(A_n)_{n\in\mathbb{N}}$ tels que :

- ✗ A_0 a pour coordonnées $(1, 0)$.
- ✗ pour tout entier naturel non nul n, le triangle OA_nA_{n+1} est rectangle en A_n, la distance A_nA_{n+1} vaut 1 et l'angle $\left(\overrightarrow{OA_n}, \overrightarrow{OA_{n+1}}\right)$ est direct.

Si x_n et y_n sont les coordonnées de A_n, on a :

$$\begin{cases} x_{n+1} = x_n - \dfrac{y_n}{\sqrt{x_n^2 + y_n^2}} \\ y_{n+1} = y_n + \dfrac{x_n}{\sqrt{x_n^2 + y_n^2}} \end{cases}$$



1. Écrire une fonction `A` d'argument N renvoyant la liste des coordonnées des points A_n pour $0 \leq n \leq N$.

2. Écrire une fonction `afficher` d'argument N affichant la figure représentant les $(N + 1)$ premiers points A_n. La tester pour $N = 60$.

Les points tournent autour de l'origine O du repère. On note $T(k)$ la valeur de n telle que le point A_n commence le k-ième tour.

3. Écrire une fonction `tours` d'argument m qui renvoie la liste des $T(k)$ pour k variant de 1 à m.
   Afficher `tours(5)`.

4. Faire tracer chacun des cinq premiers tours avec une couleur différente.

5. Déterminer les abscisses des dix premiers points d'intersection de la ligne reliant les A_n avec la partie positive de l'axe des abscisses.
   Vérifier qu'à chaque tour, on s'est éloigné du centre d'une distance environ égale à π.