
DS2 INFORMATIQUE : TRAFIC ROUTIER

Durée : 2 heures. Calculatrices interdites.

Toute fonction introduite au cours de l'énoncé ou dans votre copie peut être réutilisée dans les questions suivantes.

Lorsque vous rédigez un script Python, écrivez les caractères comme ceci.

Le sujet est composée de 4 parties. La partie 4 est indépendante et peut être traitée à partir de la question 7.

Rappels :

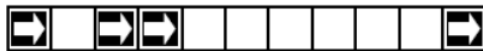
- Étant donné une liste L :
 - L'instruction `len(L)` retourne sa longueur.
 - Pour $0 \leq i < \text{len}(L)$, l'instruction `L[i]` retourne l'élément d'indice i de L .
 - Pour $0 \leq i < j \leq \text{len}(L)$, l'instruction `L[i:j]` retourne la liste
 $[L[i] , L[i+1] , \dots , L[j-1]]$
 - Si M est une autre liste, $L+M$ est la liste obtenue en concaténant L puis M . Par exemple, `[0,1]+[1,2]` retourne `[0,1,1,2]`.
- Le package `random` contient la fonction `randint` : étant donnés deux entiers $a \leq b$, `randint(a,b)` retourne un entier aléatoire dans $\llbracket a, b \rrbracket$.
- Étant donnés deux entiers a et b (b non nul), les instructions `a//b` et `a%b` retournent respectivement le quotient et le reste de la division euclidienne de a par b .

1 Préliminaires

On considère une portion de route, qu'on découpe en n « cases ». Une case peut contenir au plus une voiture. On représente une file de voitures par une liste L de longueur n :

- si une voiture est présente à la case d'indice i , alors `L[i]` retourne 1.
- sinon, `L[i]` retourne 0.

La liste L prend donc ses valeurs dans $\{0, 1\}$. Toutes les voitures circulent dans le même sens, vers la droite :



$L = [1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1]$

FIGURE 1 – Une file de longueur $n = 11$ représentant quatre voitures, situées sur les cases d'indices 0, 2, 3 et 10.

1. Donner la liste L correspondant à une file de longueur 8 avec des voitures situées sur les cases d'indice impair.
2. Écrire une fonction `occupe(L, i)` qui retourne `True` si la case d'indice i de la file (représentée par L) est occupée par une voiture, et `False` sinon.
3. Écrire une fonction `file_alea(n)` qui prend en argument un entier naturel n et retourne une file aléatoire de longueur n .
4. Combien y a-t-il de files de longueur n avec $n \in \mathbb{N}$? Justifier brièvement.
5. Écrire une fonction `egal(L1, L2)` qui retourne `True` si les listes $L1$ et $L2$ sont égales, `False` sinon. **Il est obligatoire de faire une boucle.**
6. Quelle est la complexité (dans le pire cas) de la fonction `egal` ?

2 Déplacement de voitures dans une file

Dans toute la suite du devoir, on considère une file représentée par une liste L . Les voitures se déplacent selon les règles suivantes :

- Si une voiture est à la fin de la file (la case la plus à droite), elle sort de la file.
- Toutes les autres voitures avancent d'une case vers la droite.
- La première case peut devenir occupée ou non, selon un paramètre booléen b .

Ces opérations constituent ce qu'on appelle une étape du déplacement des voitures.

On suppose avoir écrit en Python la fonction `avancer(L, b)` qui réalise une étape du déplacement des voitures dans la liste L et retourne la nouvelle liste correspondante. Le paramètre booléen b indique si la première case est occupée par une nouvelle voiture (si b vaut `True`), ou si elle est inoccupée (si b vaut `False`).

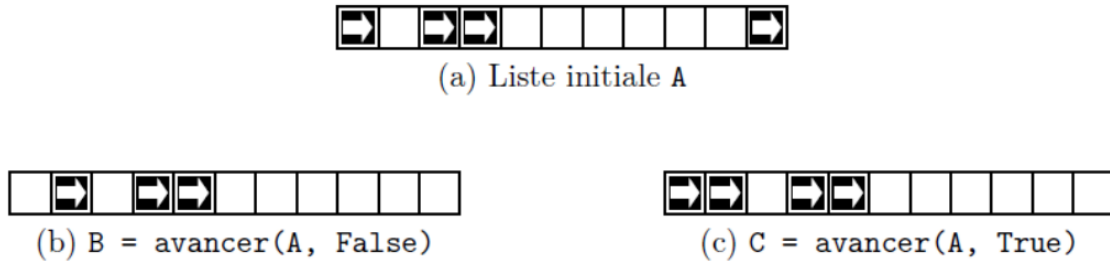


FIGURE 2 – Résultat de la méthode `avancer` appliquée à une liste A .

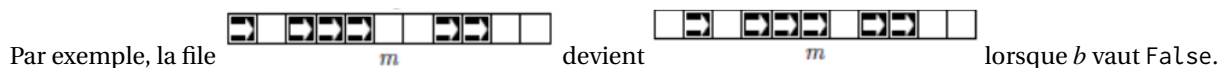
7. Que renvoie l'instruction : `avancer(avancer([0, 1, 1, 0], False), True)` ?

8. On considère un indice m , avec $0 \leq m < \text{len}(L)$. On suppose que les seules voitures qui se déplacent sont celles situées sur la case d'indice m ou à droite de celle-ci. Les autres voitures sont immobiles.



Écrire la fonction `avancer_fin(L, m)` qui retourne la liste L obtenue après cette simulation.

9. On considère $m > 0$ un indice de la liste L qui correspond à une case inoccupée. On suppose que les seules voitures qui se déplacent sont celles situées à gauche de la case d'indice m . Les autres voitures sont immobiles. Enfin, un booléen b indique si une nouvelle voiture est introduite sur la case la plus à gauche.



Écrire la fonction `avancer_debut(L, b, m)` qui retourne la liste L obtenue après cette simulation.

10. On considère $m > 0$ un indice de la liste L qui correspond à une case inaccessible qui bloque l'avancée des voitures. Si une voiture se trouve à la case $m - 1$, elle ne pourra pas avancer et la case $m - 1$ sera elle aussi inaccessible : une voiture à la case $m - 2$ ne pourra alors pas avancer non plus, etc. Les voitures situées à gauche de la case m peuvent avancer à condition que la case qui suit soit rendue libre. Les voitures situées à droite de la case m ne bougent pas. Enfin, un booléen b indique si une nouvelle voiture est introduite sur la case la plus à gauche.



Écrire la fonction `avancer_blocage(L, b, m)` qui retourne la liste L obtenue après cette simulation.

3 Croisement de deux files

On considère deux files $L1$ et $L2$ de même longueur impaire $n \geq 3$ qui se croisent en leur milieu (Figure 3.(a)). On note m l'indice de la case du milieu. La file $L1$ est horizontale et les voitures s'y déplacent vers la droite ; la file $L2$ est verticale et les voitures s'y déplacent vers le bas. Les voitures avancent au même rythme d'une case à la fois, mais la file $L1$ est prioritaire sur la file $L2$. Dans ce cadre, une étape de déplacement des voitures s'effectue ainsi :

- Tout d'abord, les voitures situées sur la case m et les suivantes de L1 et L2 avancent normalement. Si une voiture se trouve sur la dernière case d'une file, elle sort de la file comme précédemment.
- Ensuite, les voitures de la file L1 sur les case 0 à $m - 1$ avancent d'une case.
- Enfin, les voitures de la file L2 sur les case 0 à $m - 1$ avancent d'une case. Cependant, si une voiture se trouve déjà sur la case m , cette case est inaccessible et bloque l'avancée des voitures de la file L2 pour cette étape.

Les Figures 3.(b) à 3.(f) représentent la configuration des voitures après chaque étape de déplacement.

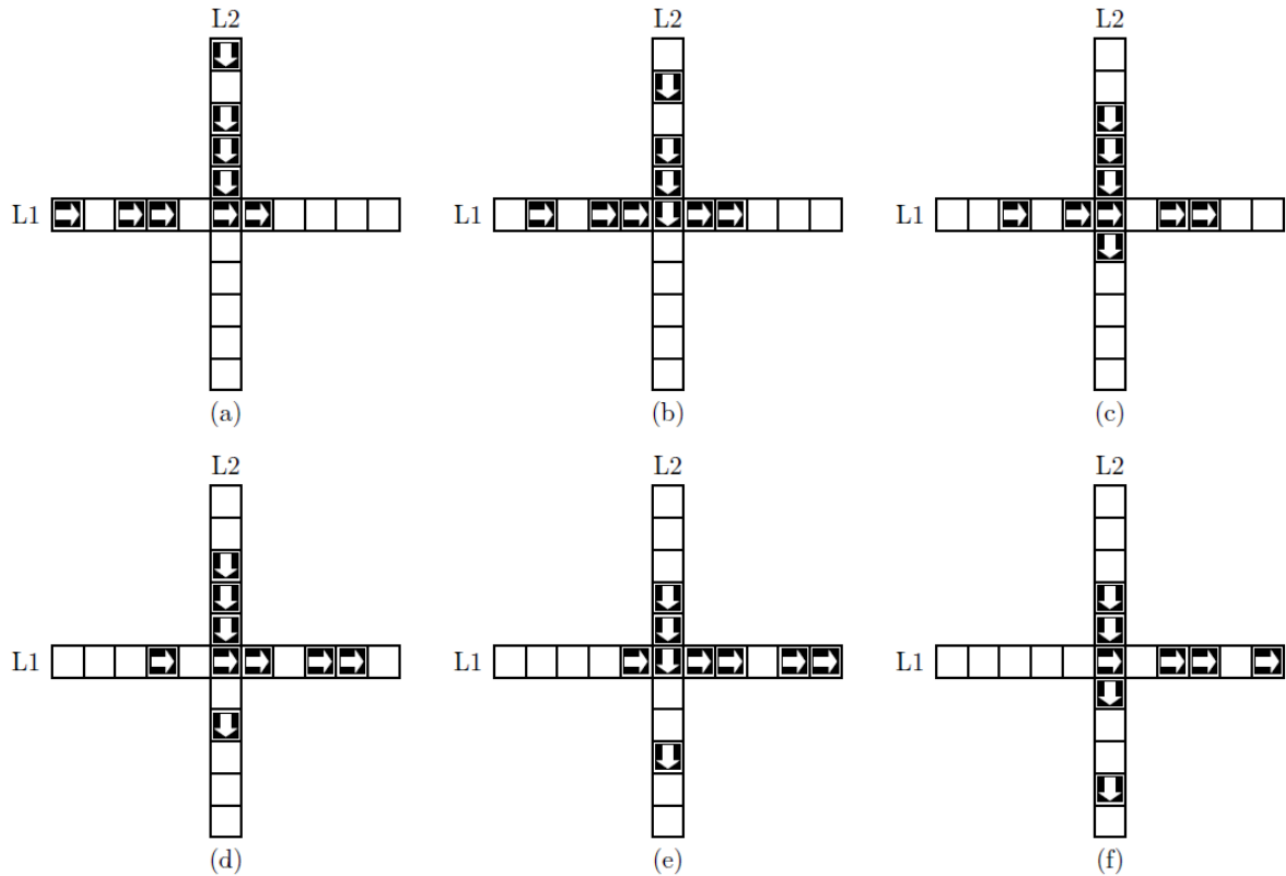


FIGURE 3 – Déplacement des voitures en présence d'un croisement. (a) : configuration initiale. De (b) à (f) : configuration obtenue étape après étape.

11. Écrire une fonction `avancer_files(L1, b1, L2, b2)` qui retourne le résultat d'une étape de simulation sous la forme d'un couple de liste $[M1, M2]$. Les booléens `b1` et `b2` indiquent respectivement si une nouvelle voiture est introduite au début des files L1 et L2. Les listes `M1` et `M2` correspondent aux listes après déplacement.
12. On considère les listes :
 - $D=[0, 1, 0, 1, 0]$
 - $E=[0, 1, 1, 0, 0]$
 Que retourne l'instruction `avancer_files(D, False, E, True)` ?

4 Représentation alternative d'une file

Dans cette partie, on considère une file représentée par une liste L de longueur $n \in \mathbb{N}^*$. On note $L_i \in \{0, 1\}$ la valeur de la case d'indice i , et on pose :

$$r(L) := \sum_{i=0}^{n-1} 2^{n-1-i} L_i = 2^{n-1} L_0 + 2^{n-2} L_1 + \dots + 2 L_{n-2} + L_{n-1}$$

On appellera l'entier $r(L)$ la représentation de la liste L .

-
13. On pose $M = [1, 0, 0, 1, 0, 0]$ et $N = [1, 0, 0, 1, 0, 1]$. Calculer $r(M)$ et $r(N)$.
 14. Écrire une fonction `rep(L)` qui retourne la représentation de la liste L .
 15. Déterminer la division euclidienne de $r(L)$ par 2, en précisant le quotient et le reste obtenus en fonction des valeurs L_i .
 16. On note L' la liste obtenue par l'instruction `avancer(L, False)`. Écrire L' en fonction des valeurs L_i , puis calculer $r(L')$. Si on note r et rp les entiers $r(L)$ et $r(L')$ respectivement, écrire une instruction simple en Python qui calcule rp à partir de r .
 17. Reprendre la question précédente lorsque L' est obtenue par l'instruction `avancer(L, True)`.
 18. Écrire une fonction `inv_rep(r, n)` qui prend en argument deux entiers r et n , et qui retourne la liste L de longueur n telle que $r(L) = r$.
 19. En utilisant les fonctions des questions précédentes, écrire une fonction `avance(L, b)` qui réalise l'opération décrite au début de la partie 2.