

PROGRAMMATION DYNAMIQUE

Consignes générales

- Les programmes doivent être systématiquement précédés d'une brève explication de leur fonctionnement (sauf éventuellement ceux de moins de 4 lignes). Inutile en revanche de surcharger vos codes à l'aide de commentaires inutiles.
- Vous êtes libres de définir autant de fonctions intermédiaires que vous le souhaitez pour répondre à une question. En particulier, on évitera de pondre des codes de plus de 8-9 lignes si on peut l'éviter.
- Le dernier exercice ne demande pas d'écrire de fonction python. Vous pouvez vous contenter de décrire vos algorithmes en pseudo-code, c'est-à-dire à mi-chemin entre du texte et une syntaxe python, le tout étant que la stratégie soit suffisamment claire.
- Si vous repérez ce qui semble être une coquille, n'hésitez pas à la modifier en expliquant les initiatives que vous êtes amené à prendre.

1. Entiers atteignables

Soit $n \in \mathbb{N}^*$. Un (petit) singe se déplace en sautant sur une règle graduée de n centimètres, en démarrant du bord gauche. A son k -ième saut, il peut sauter vers la gauche ou la droite d'exactly k centimètres, mais ne doit jamais sortir de la règle. L'entier n est dit atteignable s'il peut effectuer ses n bonds et terminer à l'extrémité droite de la règle. Par exemple,

- L'entier 1 est atteignable : il suffit de faire un saut de 1cm pour aboutir à droite de la règle ;
- L'entier 2 n'est pas atteignable : une fois le premier saut effectué, on se retrouve en position 1 et un saut de longueur 2 fait sortir de la règle ;
- L'entier 3 n'est pas atteignable : les deux premiers sauts doivent nécessairement être fait vers la droite, ce qui amène tout à droite de la règle. On peut alors faire un saut de 3cm vers la gauche, mais on ne termine pas à droite de la règle.
- L'entier 4 est atteignable par deux sauts à droite de 1 puis 2cm, suivi d'un retour à l'origine par un saut de 3cm vers la gauche, et enfin d'un saut de 4cm vers la droite.

L'objectif de l'exercice est d'écrire une fonction qui prend en argument l'entier n et renvoie le booléen **True** si et seulement si n est atteignable.

1. L'entier 6 est-il atteignable ? Justifier.
2. Montrer que l'entier 9 est atteignable en donnant une séquence de déplacements adéquate.

Dans toute la suite, on fixe un entier $n \in \mathbb{N}^*$. Pour tous entiers $i, k \in \llbracket 0; n \rrbracket$, on définit un booléen $b_{i,k}$ de la manière suivante :

$b_{i,k} = \text{True}$ si et seulement si on peut atteindre la position i à l'issue des k premiers sauts

puis on définit la liste L_k par

$$L_k = [b_{0,k}, b_{1,k}, \dots, b_{n,k}]$$

3. Préciser les listes L_0 , L_1 , L_2 et L_3 .
4. Soit $k \in \llbracket 1; n \rrbracket$ et $i \in \llbracket 0; n \rrbracket$. On suppose que $i - k \geq 0$ et $i + k \leq n$. Justifier que

$$b_{i,k} = (b_{i-k,k-1} \text{ or } b_{i+k,k-1})$$

Expliquer ce que devient la formule lorsque $i < k$ ou $i + k > n$.

5. En déduire une fonction qui prend en argument un entier $k \geq 1$ et une liste L représentant L_{k-1} et renvoie une nouvelle liste représentant L_k .
6. En déduire une fonction **atteignable** de complexité en $O(n^2)$ permettant de déterminer si un entier n donné en argument est atteignable ou pas.

2. Problème du carré maximal

On cherche à résoudre le problème suivant :

- **Entrée** : Une matrice A à n lignes et m colonnes où les coefficients $(a_{i,j})$ valent 0 ou 1. Les indices des lignes et des colonnes démarrent à 0.
- **Sortie** : La largeur maximale k d'un carré de 1 dans A , ainsi que les coordonnées (i, j) du coin en bas à droite d'un tel carré.

A titre d'exemple, pour la matrice ci-dessous, l'algorithme doit renvoyer $(i, j, k) = (4, 5, 4)$.

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}$$

A cet effet, pour tout $(i, j) \in \llbracket 0; n-1 \rrbracket \times \llbracket 0; m-1 \rrbracket$, on note $c_{i,j}$ la taille maximale d'un **carré** de 1 dont la case d'indice (i, j) est le **coin inférieur droit**.

1. Sur l'exemple de l'énoncé, déterminer les valeurs de $c_{i,j}$ pour $i, j \in \{0, 3\}$. On représentera le résultat dans un tableau.

On revient au cas général.

2. Déterminer les valeurs de $c_{0,j}$ et $c_{i,0}$ pour tous $(i, j) \in \llbracket 0; n-1 \rrbracket \times \llbracket 0; m-1 \rrbracket$.
3. Démontrer **soigneusement** la formule de récurrence suivante :

$$\forall i, j \geq 1, \quad c_{i,j} = \begin{cases} 0 & \text{si } a_{i,j} = 0 \\ 1 + \min \{c_{i,j-1}, c_{i-1,j}, c_{i-1,j-1}\} & \text{sinon} \end{cases}$$

4. En déduire un programme **carre_max** qui prend en argument la matrice **A** (sous la forme d'une liste de liste, chaque liste correspondant à une ligne de la matrice) et renvoie le triplet (i, j, k) . Préciser la complexité de votre algorithme.

3. Problème du téléspectateur

Pendant une période donnée, n émissions $E_1, \dots, E_n$ sont programmées à la télévision, toutes chaînes confondues. L'émission E_i commence à l'heure a_i et se termine à l'heure $b_i > a_i$. Un téléspectateur insatiable désire voir le plus possible d'émissions complètes. On suppose que la suite $(b_i)_{1 \leq i \leq n}$ est rangée par ordre croissant. Pour tout entier i , on note m_i le nombre maximal d'émissions (complètes) que l'on peut regarder jusqu'à l'heure b_i , avec $m_0 = 0$ par convention.

Note : Si une émission termine à l'heure a tandis qu'une autre émission commence au même horaire a , on considère que le téléspectateur peut enchaîner les deux émissions (autrement dit, le délai de zapping est négligeable).

1. Dans cette question uniquement, on suppose que $n = 5$ et les horaires des émissions sont les suivantes :

$$(a_0, a_1, a_2, a_3, a_4) = (0, 1, 3, 2, 5, 3) \quad \text{et} \quad (b_0, b_1, b_2, b_3, b_4) = (2, 4, 5, 6, 7, 8)$$

Donnez sans justification une sélection de cardinal maximal.

Indication : On pourra s'aider d'une représentation graphique des plages horaires des émissions.

2. Déterminer les valeurs de m_1 et m_2 . Pour la deuxième valeur, on distinguera deux cas suivant les valeurs de b_1 et a_2 .
3. Soit $i \in \llbracket 1; n \rrbracket$. On note $k(i)$ le plus grand entier tel que $b_{k(i)} \leq a_i$ (on prend $k(i) = 0$ si $b_1 > a_i$). Justifier l'égalité

$$m_i = \max(m_{i-1}, m_{k(i)} + 1)$$

4. Expliquer pourquoi pour tout $i \in \llbracket 1; n \rrbracket$, la valeur de $k(i)$ peut être trouvée en $O(\ln n)$ opérations élémentaires.
5. En déduire un algorithme de coût quasi-linéaire (ie en $O(n \ln n)$) qui permet de déterminer le nombre maximal d'émissions complètes que l'on peut regarder parmi $E_1, \dots, E_n$.
6. Préciser comment le modifier pour obtenir une liste d'émissions de cardinal maximal.