Algorithmique
TD n° 4 : Programmation dynamique 1

Exercice 1 : Sous-séquence croissante

Etant donné une suite d'entiers dans un tableau de taille n, par exemple $T = [3, 10, 4, 5, 6, 9, 7, 8]$, on veut déterminer la longueur maximale d'une sous-séquence de valeurs croissantes ; ainsi dans l'exemple, $(3, 4, 5, 6, 7, 8)$ est une sous-séquence de longueur maximum et sa longueur est 6. Dans cet exercice, nous cherchons une solution par programmation dynamique.

1. Y a-t-il une récurrence pour calculer x_i, la longueur maximum d'une sous-séquence croissante dans le sous-tableau restreint aux i premières cases du tableau T ?

2. Donner une récurrence pour calculer y_i, la longueur maximum d'une sous-séquence croissante qui se termine par l'élément $T[i]$.

3. Proposer un algorithme de programmation de dynamique.

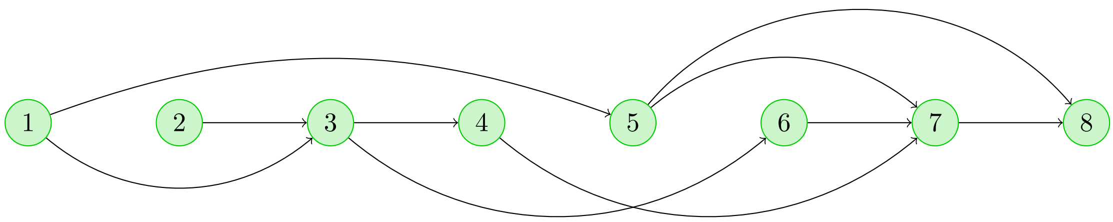
4. Quelle est sa complexité ?

5. Comment récupérer une sous-suite croissante de longueur maximale ?

Exercice 2 : Chemins dans les graphes - 1

On considère un graphe orienté $G = (S, A)$ où l'ensemble des sommets est l'ensemble $\{1, \ldots, n\}$ avec n un entier fixé, où les arcs $(i, j) \in A$ vérifient $i < j$ (le graphe est acyclique).
L'objectif est de calculer la longueur (en nombre d'arcs) maximale d'un chemin partant du sommet 1 (peu importe son point d'arrivée).

1. Considérons le graphe de la figure ci-dessous. Donner un chemin de longueur maximale partant de 1 et arrivant en 7.



2.a On souhaite calculer une table $D[i]$ avec $1 \leqslant i \leqslant n$ telle que $D[i]$ corresponde à la longueur maximale d'un chemin arrivant en i et partant de 1. On prendra $D[i] = -\infty$ lorsqu'il n'y a pas de chemin de 1 à i. Remplir le tableau ci-dessous à partir du graphe de l'exemple précédent.

i	1	2	3	4	5	6	7	8
D								

2.b Définir $D[1]$ et donner la récurrence pour $D[i+1]$ en fonction de $D[j]$ avec $j < i + 1$.

3 Proposer un algorithme qui étant donné $G = (S, A)$ défini comme précédemment, calcule la longueur maximale d'un chemin partant du sommet 1. Quelle est la complexité de votre algorithme ? Comment doit-on représenter G pour avoir une bonne complexité.

4 Comment retrouver le chemin à partir du tableau $D[-]$?

Exercice 3 : Chemins dans les graphes valués - 2

On considère un graphe orienté valué $G = (S, A, w)$ avec $w : A \mapsto \mathbf{R}$. Le graphe est acyclique. Proposer un algorithme efficace pour calculer toutes les distances des plus courts chemins depuis le sommet q.

Exercice 4 :

Super Mario doit traverser un champ de taille $n \times m$, allant de la cellule $[0, 0]$ dans le coin supérieur gauche du champ à la cellule $[n-1, m-1]$ dans le coin inférieur droit. Chaque cellule (i, j) contient un bonus $B[i, j]$ gagné si Super Mario passe par cette cellule. La matrice $B[i, j]$, de taille $n \times m$, est une donnée du problème. Les déplacements ne sont autorisés qu'à droite (est) et vers le bas (sud).

Nous notons $V[i, j]$ le bonus total maximum possible qui peut être accumulé pour aller de la cellule $[0, 0]$ à la cellule $[i, j]$, c'est-à-dire la somme des bonus des cellules le long du meilleur chemin possible de la cellule $[0, 0]$ à la cellule $[i, j]$. Nous visons à calculer la valeur $V[n-1, m-1]$.

1. Donnez une expression récursive pour $V[i, j]$ en fonction de $V[-, -]$ pour deux paramètres ayant des valeurs plus petites que i et j, respectivement.
   Astuce : Considérez les cellules *à partir desquelles* on peut arriver à la cellule $[i-1, j-1]$.
2. Donnez le pseudocode de l'algorithme de calcul de la table $V[i, j]$. Donner sa complexité.

Exercice 5 : Road trip

Pour l'été prochain, vous et vos amis planifiez de faire un petit tour en voiture électrique de location dans le pays imaginaire de Simnation. Le trajet est déjà planifié et vous cherchez maintenant à planifier vos arrêts aux bornes de recharge. Vous avez deux objectifs : Ne jamais tomber en panne à cause d'une charge nulle de vos batteries et minimiser le coût de rechargement total. Initialement, la voiture n'est pas chargée et vous pouvez la rendre avec n'importe quel niveau de charge. La première station se trouve là où vous prenez la voiture et la dernière, là où vous la rendez. Votre voiture consomme 0.1 kWh par km et possède une batterie de 10kWh qu'on peut charger avec des paliers de 0.1 kWh.
Vous connaissez par avance la liste des n stations où vous pouvez vous arrêter, la distance qui les sépares ainsi que le prix du kWh en simflouzs :
prix[i] : le prix du kWh pour la station i.
dist[i] : distance en km entre la ville i et la ville i+1.

1. Quel est le coût minimal pour prix=[12,14,21,14] et dist=[31,42,31] ?

2. Si on suppose qu'on connaît le coût minimum pour rejoindre la station $i-1$ avec une certaine charge restante, quel est le coût minimum pour joindre la i-ème avec une charge restante de c kWh ?
Plus formellement, soit $cout(i, c)$ le cout minimum pour aller à la station i et avoir une charge de c (après recharge). Donnez une relation de récurrence pour $cout(i, c)$. On suppose que c est donnée en multiple de 0.1 kWh. Quel est le cas de base ?

3. Écrire un algorithme de programmation dynamique pour calculer le coût du voyage.

4. Programmez votre algorithme pour les valeurs suivantes :
Prix (simflouzs/kWh) [12,14,21,14,17,22,11,16,17,12,30,25,27,24,22,15,24,23,15,21]
Distances (kms) [31,42,31,33,12,34,55,25,34,64,24,13,52,33,23,64,43,25,15]
Quel sera le coût du road trip ?

5. Comment peut-on récupérer les charges qu'on doit effectuer à chaque station ?

Exercice 6 :

Nous avons une boîte de Lego avec des pièces ayant toutes la même largeur et la même profondeur, mais ayant des hauteurs variables de toutes les valeurs entières possibles, en théorie une quantité illimitée de pièces de chaque type. Nous voulons calculer le nombre $T(n, k)$ de façons possibles de construire une tour de hauteur n en utilisant k pièces [1]. Par exemple :
Entrée : `n=5; k=3;`
Sortie : `6` (car il existe 6 tours possibles, comme listé dans le tableau ci-dessous) :

[3, 1, 1]	[1, 3, 1]	[1, 1, 3]
[1, 2, 2]	[2, 1, 2]	[2, 2, 1]

1. Il s'agit du nombre de compositions de l'entier n ayant k parties.

1. Donnez une expression pour $T(n, k)$ en fonction de $T(-, -)$ pour deux paramètres ayant des valeurs plus petites que n et/ou k.
   Astuce : Considérez la dernière pièce ajoutée aux tours qui sont une solution.
2. Proposer un algorithme pour calculer $T(n, k)$.