EPU GB 4ème année BIMB — *Algorithmique pour la biologie*
Examen du 22 avril 2025 — Documents autorisés.

Inscrivez **lisiblement** vos Nom et Prénom en tête de vos copies.

Exercice 1 : (Programmation dynamique – Optimisation de Cultures Cellulaires [15.5 points])

Une entreprise de biotechnologie cherche à maximiser la production de cellules dans un laboratoire. Ils ont plusieurs types de cultures cellulaires, chacune nécessitant un **temps de culture** et une quantité de **nutriments** pour atteindre une certaine production. Cependant, les ressources (temps, espace, nutriments) sont limitées et doivent être allouées de manière optimale pour maximiser la production de cellules.

Les données :

- Il y a I types de cultures cellulaires.
- On veut se limiter à T heures de culture et on a à sa disposition un maximum de N unités de nutriments à allouer entre les différentes types de cultures cellulaires.
- Chaque type de culture cellulaire i (pour $1 \leq i \leq I$) a un rendement $R_i(t, n)$, qui représente la production de cellules obtenue lorsqu'on alloue t heures de culture et n unités de nutriments à ce type de culture.

Le tableau ci-contre montre les rendements pour 3 types de cultures cellulaires (dénotés par α, β et γ), en fonction du temps de culture (en heures) et de la quantité de nutriments (en unités) alloués. Exemple : *pour le type de culture cellulaire γ, après 3h de culture avec 4 unités de nutriments, on obtient une production estimé à 120.*

Quantité nutriments	Type cult.	0h	1h	2h	3h	4h	5h
0	α	0	0	0	0	0	0
	β	0	0	0	0	0	0
	γ	0	0	0	0	0	0
1	α	0	50	100	145	180	190
	β	0	40	70	100	130	160
	γ	0	30	90	120	150	170
2	α	0	50	110	155	200	220
	β	0	40	105	140	175	210
	γ	0	35	100	130	160	205
3	α	0	50	120	160	220	250
	β	0	45	110	150	190	230
	γ	0	35	105	140	170	220
4	α	0	50	120	160	220	260
	β	0	50	120	160	200	180
	γ	0	40	80	120	160	200
5	α	0	40	100	130	180	200
	β	0	50	100	140	180	160
	γ	0	45	90	130	180	195

Le rendement est spécifique à chaque type de culture cellulaire i et dépend du nombre d'heures de culture t et de la quantité de nutriments n alloués à la culture cellulaire.

L'Objectif est de maximiser la **production totale de cellules** tout en respectant les contraintes suivantes :

- le temps total de culture ne doit pas dépasser T heures (par exemple, $T = 5$ heures),
- le nombre total d'unités de nutriments ne doit pas dépasser N unités (par exemple, $N = 5$ unités).

Indications. On s'inspirera du problème du sac à dos : le poids de l'objet i est remplacé par un couple (t_i, n_i) où t_i est le nombre d'heures alloué à la culture du type cellulaire i, et n_i est la quantité de nutriments allouée. Il faudra faire en sorte que la somme des heures de culture ne dépasse pas T et la somme des quantités des nutriments ne dépasse pas N.

Questions.

1. **[1 point]** Définissez la famille de problèmes dans laquelle se place cette recherche de production optimale. *Indication : il faudra prendre en compte les différents types de cultures cellulaires, la durée totale de culture allouée, et le nombre total d'unité de nutriments alloué.*
2. **[1.5 points]** Donnez les cas de base (cas où on peut donner le résultat en temps constant). *Indications : condirérez à la fois l'absence de temps disponible pour les cultures, l'absence de nutriments, mais aussi l'absence de types de culture cellulaire.*
3. **[2 points]** Donnez la récurrence entre les différentes solutions associées à des problèmes de la famille. *Indications : pour chaque type de culture, on a le choix entre sélectionner ce type de culture cellulaire (et dans ce cas-là, il faudra envisager toutes les possibilités pour les durées et quantités de nutriments) ou alors ne pas sélectionner ce type de culture.*
4. **[2.5 points]** En déduire un algorithme naïf récursif.
5. **[2.5 points]** En déduire un algorithme de programmation dynamique, version avec mémoïsation (appel récursif, mais en passant en argument un tableau qui permet de ne pas recalculer une valeur déjà calculée).

6. **[3 points]** Implémentez une version itérative de l'algorithme de programmation dynamique qui optimise la production tout en respectant les contraintes de durée totale et de quantité totale de nutriments.
7. **[1 point]** Analysez la complexité temporelle du dernier algorithme (itératif) en fonction du nombre de types de cultures cellulaires, de la durée totale et de la quantité totale de nutriments allouées.
8. **[1 point]** Analysez la complexité temporelle de l'algorithme par programmation dynamique (avec mémoïsation) en fonction du nombre de types de cultures cellulaires, de la durée totale et de la quantité totale de nutriments allouées.
9. **[1 point]** On souhaite, en plus de déterminer la production optimale, savoir quelles sont les durées de cultures et les quantités de nutriments allouées à chacun des types de cultures cellulaires pour obtenir l'optimum. Expliquer comment faire, autrement dit, décrire l'algorithme sans pour autant donner le pseudo-code.

Exercice 2 : (**Les mots de longueurs** k **– [7 points]**)

1. **[1 point]** Ecrivez une fonction Python qui prend en argument une liste de mots `lm` et une lettre `l` et qui renvoie une *nouvelle* liste de mots obtenus à partir de `lm` en ajoutant à la fin de chaque mot la lettre `l`.
2. **[3 points]** Etant donné un alphabet Σ représenté par une liste `L` de taille n et un entier k, écrivez un programme Python qui calcule la liste de tous les mots de longueur k possibles sur l'alphabet Σ. Par exemple : avec l'alphabet $\Sigma = \{a, b, c\}$ et $k = 2$, l'algorithme devra calculer la liste : `[aa, ab, ac, ba, bb, bc, ca, cb, cc]`.
   Indication : *la version récursive s'explique facilement. En appelant* `GenererMot(L,k)` *la fonction à écrire,* `GenererMot(L,0)` *renvoie la liste qui ne contient que le mot vide :* `['']`*, et* `GenererMot(L,k)` *fera appel à* `GenererMot(L,k-1)`*. On pourra aussi utiliser la fonction de la question 1.*
3. **[1 point]** Appliquez votre algorithme à l'alphabet $\Sigma = \{a, b\}$ (donc `L=['a','b']`) et $k = 3$. On décrira avec précision le déroulement de l'algorithme.
4. **[2 points]** Donnez la complexité de votre algorithme. Peut-on faire mieux si on veut la liste exhaustive des mots de longueur k sur Σ. Justifiez votre réponse.