DM

Ce devoir sera à rendre sur Moodle ou en version papier pour l'automne 2020, lors des journées de présence à l'Université de Montpellier. Ces journées n'étant pas encore fixées, la date de rendu ne l'est pas non plus.

Exercice 1. *Programmation dynamique*

Au cours d'un jeu, il faut choisir un emplacement pour construire une ville. Le plateau de jeu est formé d'une grille carrée divisée en cellules et une ville occupe un ensemble de cellules libres formant un carré. Le but est de construire la plus grande ville possible et on souhaite mettre au point un algorithme pour décider de l'emplacement optimal. Le problème se modélise ainsi comme suit.

Entrée: une matrice A de taille $n \times n$ dont les entrées sont des `1` (occupée) et des `0` (libre).
Sortie 1: la taille de la plus grande sous-matrice carrée de A ne contenant que des `0`.
Sortie 2: l'emplacement de cette sous-matrice.

On note T_A la taille de la plus grande sous-matrice carrée ne contenant que des `0`. Pour $0 \leq i \leq n-1$ et $0 \leq j \leq n-1$, on note aussi $t_A(i,j)$ la taille k de la plus grande sous-matrice matrice carré $k \times k$ de A ne contenant que des `0` et dont le coin inférieur droit est la cellule $A_{[i,j]}$ de A. On dit qu'un tel carré est *basé en* $A_{[i,j]}$.

$$A = \begin{array}{|c|c|c|c|c|}\hline 1&0&1&1&0\\\hline 0&0&0&1&0\\\hline 0&0&0&0&0\\\hline 1&1&0&0&0\\\hline 0&0&0&0&0\\\hline\end{array}$$

1. Sur l'exemple proposé, que vaut $t_A(0,0)$, $t_A(0,4)$, $t_A(2,2)$? Et que vaut T_A?
2. On cherche à établir un premier algorithme pour le calcul de T_A, pas forcément optimal en complexité.
   - **i.** Écrire un algorithme EXTENSIBLE(A,i,j,k) qui prend en entrée i, j et k tels que $t_A(i,j) \geq k$ et teste si $t_A(i,j) \geq k+1$. *En entrée de l'algorithme, on suppose (sans le vérifier) que* $t_A(i,j) \geq k$. Évaluer sa complexité.
   - **ii.** En déduire un algorithme CARRÉ(A,i,j) qui étant donné i et j calcule la taille du plus grand carré de `0` basé en $A_{[i,j]}$. Évaluer sa complexité.
   - **iii.** En déduire un algorithme pour calculer T_A. Quel est la complexité de votre algorithme ?
3. On va fournir maintenant un algorithme plus rapide, basé sur la programmation dynamique. On donne la relation de récurrence suivante. Pour $i \geq 1$ et $j \geq 1$, $t_A(i,j) = 0$ si $A_{[i,j]} =$ `1` et $t_A(i,j) = 1 + \min\{t_A(i,j-1), t_A(i-1,j), t_A(i-1,j-1)\}$ sinon.
   - **i.** Donner les cas de base $t_A(0,j)$ et $t_A(i,0)$ et exprimer T_A en fonction des $t_A(i,j)$.
   - **ii.** Sur l'exemple proposé, calculer T_A à l'aide de la formule proposée.
   - **iii.** Prouver la relation de récurrence donnée.
   - **iv.** Écrire un algorithme implantant la formule ci-dessus (sans preuve) et établir sa complexité.
   - **v.** Comment peut-on modifier l'algorithme pour renvoyer également la position du plus grand carré libre ?

Exercice 2. *Graphes*

L'algorithme de Dijkstra fournit les distances du sommet s passé en paramètre à tous les sommets du graphe.

1. Modifier l'algorithme de Dijkstra pour qu'il calcule également les chemins les plus courts entre le sommet s et tous les autres sommets.
2. Quelle est la complexité en mémoire de l'algorithme obtenu ?

Exercice 3. *Calculabilité*

1. Donner la table de transition d'une machine de Turing qui teste si son entrée est de la forme $0 \cdots 01 \cdots 10 \cdots 0$, où le nombre de 0 dans la première partie, de 1 au centre et de 0 à droite sont quelconques, éventuellement nuls. *La machine doit répondre* VRAI *sur les entrées* 001000*,* 1*,* 000*,* 0111000 *et* FAUX *sur les entrées* 0101 *ou* 101.
2. Montrer qu'il n'existe pas d'algorithme qui, étant donné le code d'un algorithme A et une entrée x, détermine si sur l'entrée x, A renvoie 1. *Montrer qu'avec un tel algorithme, on pourrait résoudre le problème de l'arrêt.*