Algorithme de Floyd

Soit $R = (E, \Gamma, l)$ un réseau. On veut chercher, pour tous les points x et y de E, la longueur d'un plus court chemin de x à y. On pose $E = \{1, 2, ..., n\}$.
On considère la matrice $L^0 = (l^0_{i,j})$ telle que $l^0_{i,i} = 0$ et, si $i \neq j$:

$$l^0_{i,j} = \begin{cases} l(i,j) & \text{si } (i,j) \in \overrightarrow{\Gamma} \\ +\infty & \text{sinon} \end{cases}$$

1) On considère la matrice $L^k = (l^k_{i,j})$ où $l^k_{i,j}$ représente la longueur minimum des chemins de i à j dont les sommets intermédiaires sont dans $E_k = \{1, 2, ..., k\}$. Pour $k = 0$, on pose $E_k = \emptyset$. On voit alors que L^0 correspond bien à L^k avec $k = 0$. On voit aussi que L^n est la réponse au problème que l'on veut résoudre.
Proposer une méthode pour calculer L^k. Pour cela on cherchera à établir une formule de récurrence permettant de calculer $l^k_{i,j}$ en fonction de termes de la forme $l^{k-1}_{i',j'}$. On supposera qu'il n'y a pas de circuit absorbant.

2) Proposer un schéma de programme permettant de calculer L^n dans le cas où il n'y a pas de circuit absorbant.

3) Quelle est la complexité de calcul de votre schéma de programme?

4) Proposer une variante de votre schéma de programme permettant de calculer L^n ou bien de détecter la présence d'un circuit absorbant.