ALGORITHME DE FLOYD-WARSHALL

Leçons 925, 931

Références Cormen p. 609

Types de données et algorithmes, Froidevaux p. 477

Théorème

Soit $G = (S, A)$ un graphe et $w : A^2 \to \mathbb{R}$ une fonction de poids.
Si G n'a pas de cycle de poids négatif, on peut trouver toutes les plus courtes distances entre deux sommets en $O(|S|^3)$, ~~et~~, s'il a un tel cycle, on le trouve aussi.

Résumé

I – Structure d'un plus court chemin : sommets intermédiaires plus petits que k.

II – Une relation de récurrence sur la distance n'utilisant que les k premiers sommets

III – Calcul des distances par programmation dynamique.

Remarque si $u \xrightarrow{p_1} v \xrightarrow{p_2} w$ est un plus court chemin, alors p_1 et p_2 sont aussi des plus courts chemins : c'est ce qui va nous permettre de construire l'algorithme.

I • Pour $p = \langle v_1, v_2, \dots, v_p \rangle$, les sommets v_2 à v_{p-1} sont intermédiaires.

• Supposons que $S = [\![1 \dots n]\!]$, on pose $S_k = [\![1 \dots k]\!]$.

→ on s'intéresse aux chemins de i à j ne passant que par des sommets de S_k.

Pour un chemin $u \xrightarrow{p} v$ dans S_k, il y a deux possibilités :

* soit k n'est pas un sommet intermédiaire de p, ces sommets sont donc tous dans S_{k-1} et le chemin est le même dans S_{k-1} et S_k.
* soit k est un sommet intermédiaire de p et on peut diviser p en $u \xrightarrow{p_1} k \xrightarrow{p_2} v$. p_1 et p_2 n'ont alors que des sommets intermédiaires dans S_{k-1} et sont les plus courts chemins $u \to k$ et $k \to v$.

II] - Posons alors $d_{ij}^{(k)}$ le poids d'un plus court chemin de i à j dont tous les sommets intermédiaires sont dans S_k.

- La remarque de I] donne la relation de récurrence :

$$d_{ij}^{(k)} = \begin{cases} w_{ij} & \text{si } k = 0 \\ \min(d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)}) & \text{si } k \geq 1. \end{cases}$$

il n'y a pas d'intermédiaire, c'est donc une arête de G !

on ne passe pas par k

on passe par k.

- En itérant, on trouve $d_{ij}^{(n)}$, la distance entre i et j qui s'autorise de passer par tous les sommets de G.

III] - On peut ainsi définir l'algorithme de Floyd-Warshall ainsi, prenant en entrée la matrice $W = (w_{ij})_{i,j}$, coïncidant avec $(d_{ij}^{(0)})_{i,j}$:

```
FLOYD-WARSHALL(W)
| n = nombre de lignes de W.
| (d_ij^(0))_ij = (w_ij)_ij
| Pour k = 1 à n
| | Pour i = 1 à n
| | | Pour j = 1 à n
| | | L d_ij^(k) = min(d_ij^(k-1), d_ik^(k-1) + d_kj^(k-1))
| Renvoyer D = (d_ij^(n))_i,j
```

Le temps d'exécution de cet algorithme est $O(1)$ par passage dans la boucle, c'est-à-dire $O(|S|^3)$.
Et il est correct par définition de la relation de récurrence sur $d^{(k)}$.

Remarque on n'utilise qu'un tableau de taille fixe, sans opération ni structure de données ~~é~~élaborée : la constante du O est donc petite, ce qui rend l'algorithme intéressant même sur des graphes de taille modérée.