TD2 - Plus courts chemins

Graphes et programmation dynamique

M1 Informatique - Semestre 1 - Année 2022-2023

Université Côte d'Azur

Christophe Crespelle
christophe.crespelle@univ-cotedazur.fr

Le sujet sera traite sur une seance d'1h30. Il est recommande que vous utilisiez les exercices que vous n'avez pas eu le temps de faire en seance comme entrainement en vue de l'examen final. Les exercices les plus importants sont les 1, 2 et 3 : commencez par ceux la.

Exercice 1. *Differentes facons de cheminer.*

a. Montrez qu'un sous-chemin d'un plus court chemin est aussi un plus court chemin.

b. Soit $G = (V, E)$ un graphe non oriente et non value. Soit $u, v \in V$ et soit $w \in V$ tel que wv est la derniere atrete d'un plus court chemin de u a v. On note $Nbc_u(v)$ le nombre de plus courts chemins de u a v. Quel est le nombre $Nbc_u(wv)$ de plus courts chemin de u a v qui terminent par l'arete wv ?

c. On note $Prec_u(v)$ l'ensemble des sommets $w \in V$ tels que wv est la derniere arete d'un plus court chemin de u a v. Exprimez $Nbc_u(v)$ en fonctions des $Nbc_u(w)$ pour $w \in Prec_u(v)$.

Dans la suite de l'exercice, le graphe G donne en entree est non oriente et non pondere.

d. Adaptez l'algorithme de parcours en largeur depuis u pour qu'en plus de la distance a u, il calcule pour chaque sommet v, le nombre de plus courts chemins de u a v. Quel est la complexite de l'algorithme adapte ?

e. Adaptez l'algorithme de parcours en largeur depuis u pour qu'il determine et stocke pour chaque sommet u la liste des sommets dans $Prec_u(v)$. Quelle est la taille de la representation ainsi fournie en sortie de l'algorithme ?

f. Etant donne la representation fournie en sortie de l'algorithme a la question precedante, comment tirer uniformement aleatoirement un plus court chemin de u a v. Quelle est la complexite d'un tel tirage aleatoire ?

Exercice 2. *La charue avant les boeufs.*

Quand vous vous levez le matin, pour vous habiller, vous devez mettre vos chaussettes avant vos chaussures, et vous pouvez mettre votre montre à n'importe quel instant. L'habillage d'un cadre moyen peut globalement se représenter par le graphe (oriente) de contraintes donne sur la figure 1, dans lequel un arc du vetement A vers le vetement B signifie que le vetement A doit obligatoirement etre mis avant de mettre le vetement B.

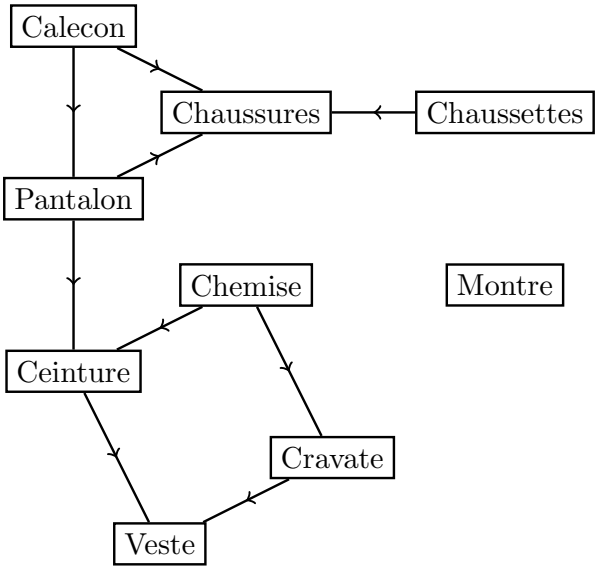

FIGURE 1 – Le graphe des contraintes sur l'ordre d'habillage. Un arc du vetement A vers le vetement B signifie que le vetement A doit obligatoirement etre mis avant de mettre le vetement B.

On se propose de trouver une méthode systematique pour habiller tout le monde, quels que soient ses vetements. C'est a dire, qu'etant donne un graphe de contrainte comme celui de la figure 1, on veut trouver un ordre total sur les vetements à enfiler qui respecte toutes les contraintes du graphe de contrainte.

a. Pour quels graphes de contraintes cela est-il possible ? Demontrez le.

Indication. Determinez d'abord pour quels graphes ce n'est pas possible.

Indication. Prouvez que pour les autres c'est toujours possible, par induction sur le nombre de sommets, en choisissant bien le sommet a enlever dans l'induction.

b. Effectuer un parcours en profondeur du graphe de la figure 1 depuis un sommet de votre choix et notez le temps t_{deb} de debut de traitement et le temps t_{fin} de fin de traitement pour chaque sommet.

c. Classez les sommets par ordre de fin decroissant. Que remarquez vous ?

d. Montrez que cela est une propriete generale : dans un graphe G qui satisfait les conditions de la question 1 de l'exo 2, quelque soit le sommet $u \in V$, l'ordre inverse de fin de traitement des sommets dans un parcours en profondeur de source u est un tri topologique.
N.B. : on utilise la version du parcours en profondeur qui ne s'arete que lorsque tout les sommets du graphe ont ete visites.

On va concevoir un deuxieme algorithme pour le meme probleme, plus evident mais plus delicat a implementer que DFS.

Definition [Source et puits]
Une **source** dans un graphe oriente est un sommet qui n'a aucun voisin entrant. Un sommet qui n'a aucun arc sortant est appele un **puits**.

e. Montrez qu'un graphe oriente sans circuit (DAG) possede au moins une source.

f. Montrez que si u est une source d'un DAG G, alors il existe un ordre topologique de G qui commence par u.

g. Deduisez de la question precedante un algorithme pour determiner un tri topologique d'un DAG G.

h. Comment determinez les sources d'un DAG sur ses listes d'adjacences ? Quel est l'etape la plus couteuse en temps dans l'algorithme que vous avez propose a la question precedante ? Quel est la complexite totale de cet algorithme ?

i. Comment ameliorez l'implementation de cet algorithme pour obtenir une complexite lineaire.

Exercice 3. *A star is born.*

L'algorithme A*, derit dans l'algorithme 1, est un algorithme de plus court chemin dans un graphe (eventuellement oriente) pondere positivement qui trouve un chemin de la source s a la destination t. C'est une variante de l'algorithme de Dijkstra dont le but n'est pas de determiner un chemin de s vers toutes les destinations, mais seulement vers la destination t donnee. L'enjeu est de minimiser la partie du graphe qui est explore par l'algorithme afin de decouvrir un tel chemin. Pour cela, on utilise une fonction h qui est donnee avec le graphe et qui donne une estimation (a priori sans garantie) pour chaque noeud u de la distance de u a t.

Algorithme 1 : L'algorithme A*(G,s,t,h)

```
1  Un tableau dist de taille n ; Un tableau coul de taille n;
2  pour u de 0 a n − 1 faire
3      dist[u] ← +∞ ; f[u] ← +∞ ; coul[u] ← blanc;
4  dist[s] ← 0 ; f[s] ← dist[s] + h[s] ; coul[s] = gris ; Q ← {s};
5  tant que Q ≠ ∅ et non(trouve) faire
6      u ← min_f(Q) ; Q ← Q \ {u} ; coul[u] ← noir;
7      si u = t alors
8          trouve ← vrai
9      sinon
10         pour v ∈ N(u) faire
11             si dist[u] + w(u, v) < dist[v] alors
12                 dist[v] ← dist[u] + w(u, v) ; f[v] ← dist[v] + h(v);
13                 si coul[v]=blanc alors
14                     Q ← Q ∪ {v} ; coul[v] ← gris;
```

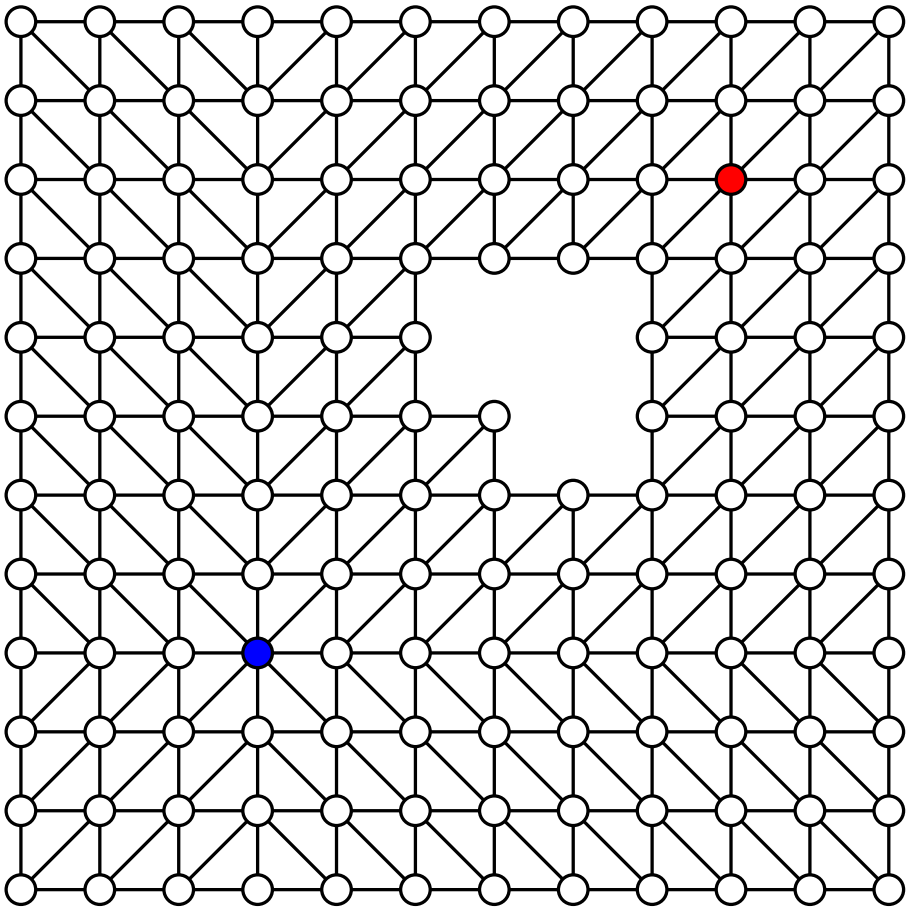

FIGURE 2 – Un graphe exemple pour l'execution de $A*(G, s, t, h)$ avec un poids unitaire sur toutes les aretes. Prendre pour h la distance euclidienne sur le dessin fourni, le noeud bleu pour s et le noeud rouge pour t.

a. Executez l'agorithme $A*(G, s, t, h)$ sur le graphe de la figure 2 en prenant pour h la distance euclidienne sur le dessin fourni, le noeud bleu pour s et le noeud rouge pour t. Les aretes de G ont toutes poids 1.

b. Analysez le pseudo-code de l'algorithme 1. Quelle est la difference entre A* et l'algorithme de Dijkstra ?

On considere maintenant des fonctions d'estimation h ayant la propriete de ne jamais surestimer la distance de u a t, c'est a dire telles que $\forall u \in V, h(u) \leq dist(u, t)$.

c. Montrez que pour de telles h, le chemin de s a t qui est decouvert par A* est bien un plus court chemin de s a t.

Indication. Considerez le moment ou t est choisi comme le noeud u a la ligne 6.

d. A-t-on $dist[u] = dist(s, u)$ pour tous les noeuds noirs u ?

e. Dans le cas (tres) particulier ou $h(u) = dist(u, t)$, quels sont les noeuds colores en noirs a la fin de l'algorithme ?

Exercice 4. *Bandit de grand chemin*

La guerilla fait rage dans la region de Paradisio. L'armee rebelle prevoit de s'emparer de la ville de Dolcevita. Pour maximiser ses chances de reussite, elle veut eviter d'une reaction rapide du pouvoir en place, qui concentre ses forces a Paradisio. La carte de la region, avec les differents temps de trajet, est donnee sur la figure 3. Les rebelles possede l'arsenal necessaire pour couper une voie de communication, mais pas plus, et ils souhaitent rallonger au maximum le temps de trajet de Paradisio a Dolcevita.

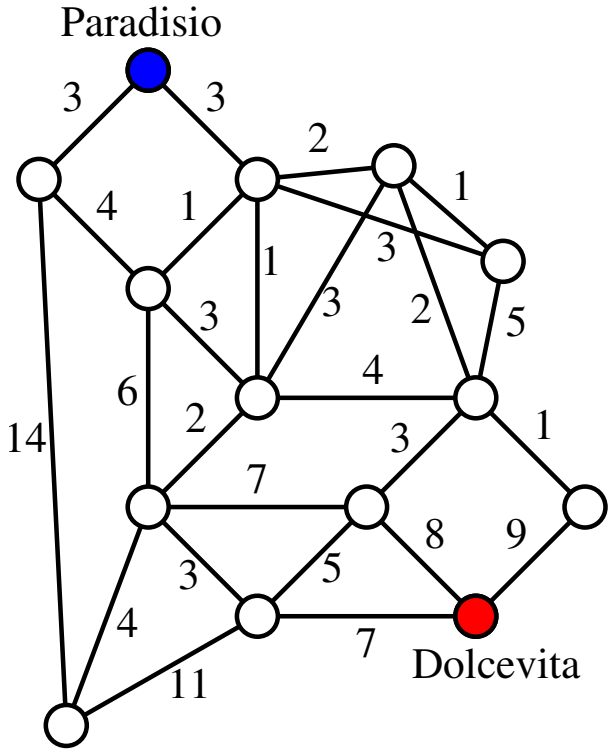

FIGURE 3 – La carte de la region de Paradisio avec les temps de trajets entre villes.

a. Quelle est la voie que l'armee rebelle doit couper pour maximiser le temps de trajet de Paradisio a Dolcevita ?

b. Dans le cas general, etant donne un graphe G pondere positivement et deux sommets A et B, proposer un algorithme simple pour trouver l'arete a retirer de G afin d'augmenter au maximum la longueur du plus court chemin entre les sommets A et B.

c. Quelle est la complexite de votre algorithme ?

d. Proposez un algorithme lineaire qui decide s'il est possible d'augmenter la longueur du plus court chemin entre A et B en retirant une seule arete de G.

> **Indication.** Adapter un peu l'algorithme de Dijkstra pour qu'il garde trace de tous les plus courts chemin entre la source et les autres sommets.