

INFORMATIQUE TRONC COMMUN

DEVOIR SURVEILLÉ 3

Chemins optimaux dans un graphe temporel

Question 1 Donner la spécification précise de l'algorithme de Dijkstra : quelles sont les entrées, quelles propriétés doivent-elles vérifier, et quelles sont alors les sorties renvoyées par l'algorithme ?

Dans ce sujet, nous allons étudier les graphes temporels, dans lesquels les arcs représentent le trajet d'un moyen de transport comme un bus ou un métro, avec une heure de départ et une heure d'arrivée données, et quel type d'algorithme est utilisé pour calculer des itinéraires optimaux dans ce type de contexte.

On se fixe un ensemble de sommets $S = \llbracket 0, n-1 \rrbracket$. Un **arc temporel** (qu'on abrègera en arc dans la suite) est un quadruplet (u, v, td, ta) , où u est le sommet de départ, v le sommet d'arrivée, td le temps de départ et ta le temps d'arrivée, avec $td \leq ta$.

Un **graphe temporel** est un ensemble d'arcs temporels.

Un **chemin de u à v à partir du temps t** dans un graphe temporel est une suite d'arcs

$(u_0, v_0, td_0, ta_0) \dots (u_k, v_k, td_k, ta_k)$ (possiblement vide si $u = v$) telle que :

- $u = u_0$ et $t \leq td_0$
- $v = v_k$
- $\forall i \in \llbracket 0, k-1 \rrbracket, v_i = u_{i+1}$ et $ta_i \leq td_{i+1}$

Les contraintes sur les temps indiquent qu'on ne peut pas prendre un moyen de transport parti avant qu'on n'arrive. Dans ce modèle, on peut noter que le temps pris pour la correspondance entre deux moyens de transport est négligé (on peut considérer qu'il est déjà inclus dans les temps d'arrivée).

Question 2 Écrire une fonction `chemin_valide` prenant en argument une liste de quadruplets et trois nombres u, v, t , et testant si la liste correspond à un chemin de u à v à partir du temps t .

Nous allons représenter les graphes temporels en Python à l'aide de listes d'adjacence faisant intervenir des listes d'horaires entre deux sommets u et v . On définit ainsi le graphe suivant (les différentes représentations des horaires vont être expliquées dans la suite) :

```
horaires_0_1 = [(3, 10), (7, 15), (9, 16), (15, 22)]
horaires_0_2 = [(2, 5, 12, 50), (11,)]
horaires_1_2 = [(1, 5), (6, 8), (5, 9), (7, 11, 4, 31), (6,)]
horaires_1_3 = [(1, 20), (10, 30), (21, 32), (3, 28, 5, 48)]
horaires_1_4 = [(6, 10), (18, 22), (28, 33), (40, 44), (12,)]
horaires_2_1 = [(3, 7), (6, 8), (13, 15), (2, 6, 4, 38), (6,)]
horaires_2_4 = [(4, 11, 10, 54)]
horaires_3_4 = [(3, 10), (10, 18), (22, 31), (35, 44)]
horaires_4_3 = [(6, 14, 10, 46)]
```

```
G = [(1, horaires_0_1), (2, horaires_0_2)],
[(2, horaires_1_2), (3, horaires_1_3), (4, horaires_1_4)],
[(1, horaires_2_1), (4, horaires_2_4)],
[(4, horaires_3_4)],
[(3, horaires_4_3)]
```

L'exemple le plus simple d'horaire est un couple (td, ta) , indiquant la présence d'un arc (u, v, td, ta) dans le graphe (notons qu'il y a en général plusieurs arcs entre deux sommets). Par exemple, la liste `horaires_0_1` indique qu'il y a 4 trajets existants pour aller directement de 0 à 1, le premier partant au temps 3 et arrivant au temps 10.

Question 3 Écrire une fonction `meilleur_horaire` prenant en argument une liste d'horaires ne contenant que des couples (comme `horaires_0_1`), et un temps t , et renvoyant le couple (td, ta) de ta minimal vérifiant $t \leq td$. Ainsi, `meilleur_horaire(horaires_0_1, 8)` doit renvoyer $(9, 16)$. On renverra un couple de valeurs infinies s'il n'existe plus de trajet disponible.

Un autre type d'horaire est représenté par un un-uplet, comme $(11,)$ dans `horaires_0_2`. Il correspond à un moyen de transport ne nécessitant jamais d'attente, à l'exemple de la marche à pied. Sa valeur d est alors la **durée** du trajet, et il représente donc ainsi infinité d'arcs de temps d'arrivée $td + d$ pour tout temps de départ td .

Enfin, les quadruplets (di, ai, p, df) représentent des horaires périodiques :

- di est le temps du premier départ de la série ;
- ai est le temps de la première arrivée de la série ;
- p est la période espaçant les temps de départ, ainsi que les temps d'arrivée ;
- df est le temps du dernier départ de la série ($df - di$ étant donc un multiple de p).

Question 4 Compléter la fonction `meilleur_horaire`, pour qu'elle prenne en compte tous les types d'horaires. Ainsi :

- `meilleur_horaire(horaires_1_2, 0)` doit renvoyer $(1, 5)$ (cas d'un horaire ponctuel) ;
- `meilleur_horaire(horaires_1_2, 10)` doit renvoyer $(11, 15)$ (cas d'un horaire périodique) ;
- `meilleur_horaire(horaires_1_2, 12)` doit renvoyer $(12, 18)$ (cas de la marche).

Les élèves quittant le lycée à un temps t donné ont hâte de rentrer pour se mettre au travail sur le prochain DM. On cherche ainsi à calculer pour un sommet source s et pour tout sommet de destination u , le **chemin de s à u à partir du temps t arrivant le plus tôt**. Pour cela, on donne la variante suivante de l'algorithme de Dijkstra :

Fonction `Dijkstra_temporel(G, s, t)` :

```

T[s] vaut t
T[u] est infinie pour les autres sommets u
P[u] = None pour chaque sommet u
les sommets sont non marqués
tant qu'il reste un sommet non marqué:
    soit u sommet non marqué de T[u] minimal
    on marque u
    pour chaque successeur v de u:
        soit (td, ta) le meilleur horaire pour aller de u à v à partir de T[u]
        Si ta < T[v]:
            T[v] = ta
            P[v] = (u, td, ta)
on renvoie T, P

```

Question 5 Appliquer à la main cet algorithme sur le graphe G en partant du sommet 0 (le lycée) et du temps 0. On indiquera clairement la valeur de T et P à la fin de chaque itération.

Question 6 M. Pompigne a encore fini sa colle en retard. Combien le fait de passer de $t = 0$ à $t = 20$ comme temps de départ fait-il perdre de temps à un élève se rendant au sommet 3 ?

Question 7 Implémenter la fonction `Dijkstra_temporel` en Python.

Question 8 Écrire une fonction `chemin` prenant en argument un tableau de père P et un sommet u accessible depuis s , et renvoyant le chemin (sous forme de liste de quadruplets) de s à u .

Question 9 Le matin, les élèves veulent le chemin **partant le plus tard** et arrivant cinq minutes avant le début du cours. Expliquer comment modifier l'algorithme pour calculer ce type de résultat.

Question 10 Démontrer formellement qu'en sortie de l'algorithme, T contient le temps d'arrivée au plus tôt depuis s à partir de t pour chaque sommet.