# CM 11

## Graphes

### Marches, chemins et distances

Cours Optimisation Discrète 12 mai 2011

Friedrich Eisenbrand
EPFL



Notes

## La borne et le prix

**Prix de 1.000 2.000 Francs Suisses**

Pour l'étudiant de mon cours (inscrit en Optimisation Discrète en 2011 à l'EPFL) qui prouvera une borne supérieure polynômiale sur $D(n, m)$ ou qui réfutera l'existence d'une telle borne pour l'abstraction de la base.



Notes

## Toutefois, il existe un algorithme polynômial pour la programmation linéaire

**Taille de l'entrée**

- Taille de l'entier $a$ : $\lceil \log(|a| + 1) \rceil$
- Taille du nombre rationnel $p/q$ avec $\gcd(p, q) = 1$ : $\text{taille}(p) + \text{taille}(q)$
- Taille de la matrice $A \in \mathbb{Q}^{m \times n}$ : $m \cdot n \cdot \text{taille}(U)$, où $U$ est un majorant des numérateurs et dénominateurs des entrées.
- Taille du vecteur $v \in \mathbb{Q}^n$ : $n \cdot \text{taille}(U)$, où $U$ est un majorant des numérateurs et dénominateurs des entrées.



Notes

## Algorithme en temps polynômial pour la programmation linéaire

**Théorème (Khachiyan 79)**

*Il existe un algorithme pour résoudre le programme linéaire*

$$\max\{c^T x \colon x \in \mathbb{R}^n,\ Ax \leq b\}$$

*qui effectue un nombre polynômial d'opérations arithmétiques sur des nombres rationnels de taille polynômiale. Polynômial signifie ici $O(n^k)$ pour une constante $k$, et $n$ est un majorant des tailles de $A$, $b$ et $c$.*



Notes

## Graphes orientés

**Définition**

Un graphe orienté est un couple $G = (V, A)$, où $V$ est un ensemble fini, dont les éléments sont appelés sommets de $G$ et $A \subseteq (V \times V)$ est l'ensemble des arcs de $G$. Nous indiquons un arc par ses deux sommets définissant $(u, v) \in A$. Les sommets $u$ et $v$ sont appelés extrémité initiale et extrémité finale de l'arc $(u, v)$ respectivement.

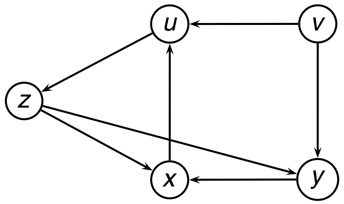



**FIGURE:** Exemple d'un graphe orienté avec 5 sommets et 7 arcs.



Notes

## Marches et chemins

**Définition (Marche, chemin, distance)**

Une marche est une séquence de la forme

$$P = (v_0, a_1, v_1, \ldots, v_{m-1}, a_m, v_m),$$

où $a_i = (v_{i-1}, v_i) \in A$ pour $i = 1, \ldots, m$. Si les sommets $v_0, \ldots, v_m$ sont tous différents alors $P$ est un chemin. La longueur de $P$ est $m$. La distance entre deux sommets $u$ et $v$ est la longueur d'un plus court chemin de $u$ à $v$ qu'on note par $d(u, v)$.

**Exemple**

Ce qui suit représente une marche et un chemin dans le graphe de la Figure 10.

$$u, (u, z), z, (z, x), x, (x, u), u, (u, z), z, (z, y), y$$
$$u, (u, z), z, (z, y), y$$



Notes

## Représentation

**Représentation**

Un graphe ayant $n$ sommets est représenté comme un tableau $A[v_1, \ldots, v_n]$, où la composante $A[v_i]$ est un pointeur vers une liste de sommets, les voisins de $v_i$. $N(v_i) = \{u \in V : (v_i, u) \in A\}$.

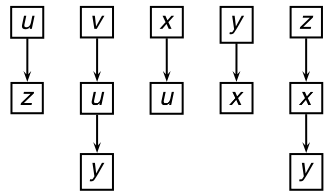



**FIGURE:** Représentation du graphe dans la Figure 10 par liste d'adjacence.



Notes

## Breadth-first-search

**Lemme**

*Soit $V_i \subseteq V$ l'ensemble de sommets qui sont à distance $i$ de $s$. Pour $i = 1, \ldots, n-1$, l'ensemble $V_i$ est égal à l'ensemble de sommets $v \in V \backslash (V_0 \cup \cdots \cup V_{i-1})$ tel qu'il existe un arc $(u, v) \in A$ avec $u \in V_{i-1}$.*



Notes

## Breadth-first-search

L'algorithme maintient à jour les tableaux

$$D[v_1 = s, v_2, \ldots, v_n]$$
$$\pi[v_1 = s, v_2, \ldots, v_n]$$

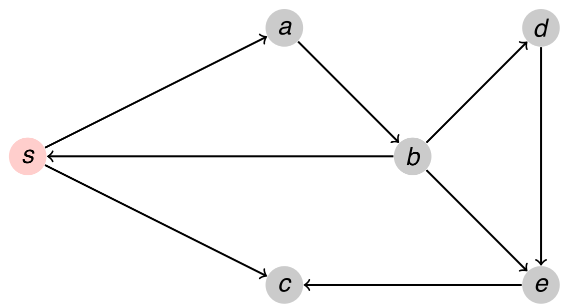
et une file d'attente $Q$ qui contient seulement $s$ au début.

**breadth-first-search**

**while** $Q \neq \emptyset$
  $u := head(Q)$
  **for** each $v \in N(u)$
    **if** $(D[v] = \infty)$
      $\pi[v] := u$
      $D[v] := D[u] + 1$
      $enqueue(Q, v)$
  $dequeue(Q)$



## Exemple




$Q = [s]$



Notes

Notes

## Correction

**Théorème**

*Breath-first-search attribue les labels $D$ et $\pi$ correctement.*



Notes

## Arbres

**Définition (Arbre)**

Un arbre orienté est un graphe orienté $T = (V, A)$ avec $|A| = |V| - 1$ et dans lequel il y a un sommet $r \in T$ tel qu'il existe un chemin de $r$ à tous les autres sommets de $T$.

**Lemme**

*Considérons les tableaux $D$ et $\pi$ quand l'algorithme de parcours en largeur a terminé. Le graphe $T = (V', A')$ où $V' = \{v \in V \colon D[v] < \infty\}$ et $A' = \{\pi(v)v \colon 1 \leq D[v] < \infty\}$ est un arbre.*

**Définition**

L'arbre $T$ mentionné ci-dessus est l'arbre des plus courts chemins du graphe orienté (non-pondéré) $G = (V, A)$.



Notes

## Analyse de l'algorithme

**Théorème**

*L'algorithme de parcours en largeur (breadth-first-search) se déroule en temps $O(|V| + |A|)$.*



## Graphes pondérés

**Définition**

Une marche pour laquelle le sommet de départ et celui d'arrivée coïncident est appelée cycle.

**Définition**

Soit un graphe orienté $D = (V, A)$ ainsi qu'une fonction de longueur $c : A \longrightarrow \mathbb{R}$. La longueur d'une marche $W$ est définie comme

$$c(W) = \sum_{\substack{a \in A \\ a \in W}} c(a).$$

**Théorème**

*Supposons que tout cycle dans $D$ est de longueur non-négative et supposons qu'il existe une marche de $s$ à $t$ dans $D$. Alors il existe un chemin reliant $s$ à $t$ qui est de longueur minimale parmi toutes les marches reliant $s$ et $t$.*



Notes

Notes

## Bellman-Ford

**Algorithme de Bellman-Ford**

i) $f_0(s) = 0$, $f_0(v) = \infty$ pour tout $v \neq s$
ii) Pour $k < n$ si $f_k$ a été trouvé, calculer

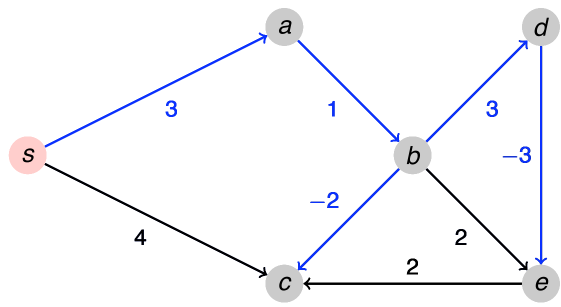
$$f_{k+1}(v) = \min\{f_k(v), \min_{(u,v)\in A}\{f_k(u) + c(u,v)\}$$

pour tout $v \in V$.



Notes

## Exemple






Notes