

# Examen du cours de algorithmique avancée, M1 Informatique, Université Paris Diderot

Michel Habib

19 mai 2011

## 1 Certificats

Rappelons qu'un certificat est une chaîne de caractères qui permet de vérifier le résultat d'un calcul. Par exemple si un programme répond qu'un nombre entier  $n$  est composé, deux entiers  $p, q$  tels que  $n = pq$  constituent un certificat du fait que  $n$  ne soit pas premier.

Il s'agit de proposer des certificats pour les problèmes algorithmiques précisés ci-dessous. Dans chacun des cas, on précisera la taille du certificat en fonction de celle de la donnée, ainsi que la complexité de l'algorithme permettant de calculer le certificat ainsi que celle de l'algorithme qui permet de vérifier le certificat.

1. Un plus court chemin entre deux sommets  $a$  et  $b$
2. Un couplage de cardinal maximum

## 2 Arbres de poids minimum

On considère un graphe  $G = (X, E)$  connexe, muni d'une valuation  $\omega : E \rightarrow \mathbb{N}$ . L'algorithme de Prim décrit ci-dessous calcule un arbre recouvrant de poids minimum de  $G$ .

**Algorithme de Prim;**  
 $F \leftarrow \emptyset;$   
 $A \leftarrow \{x_0\};$   
**tant que**  $|F| < n - 1$  **faire**  
    Calculer l'arête  $e = (a, x)$  (avec  $a \in A$ ) de coût minimal de  $\theta(A)$  (le  
    cocycle associé à  $A$ );  
     $F \leftarrow F \cup \{e\};$   
     $A \leftarrow A \cup \{x\}$   
**fin**

1. Rappeler la preuve de cet algorithme.
2. Ecrire un algorithme qui calcule le 3ème meilleur arbre recouvrant de poids minimum, s'il existe. Peut-on produire un certificat dans tous les cas ?
3. On suppose disposer d'un arbre recouvrant  $T$  de poids minimum. Le poids d'une arête  $xy \in V$  augmente d'une quantité  $\delta > 0$ , écrire un algorithme qui calcule le nouvel arbre recouvrant  $T'$  de  $G$  à partir de  $T$ .

### 3 Caractérisations des ordres de parcours

1. Reprenons l'algorithme de Prim décrit ci-dessus, peut-on le réécrire comme un parcours de graphe dont le résultat soit un ordre total  $\sigma$  sur les sommets ?
2. Peut-on caractériser un tel parcours avec une caractérisation du type de celle vue pour les parcours BFS, DFS ... ?  
Piste : on peut introduire la valuation  $\omega$ .
3. Peut-on étendre aux graphes orientés les caractérisations vues sur les parcours dans les graphes orientés, si oui faire la preuve pour un parcours en largeur puis en profondeur.
4. Même idée avec l'algorithme de Dijkstra, que peut-on dire de l'ordre  $\sigma$  du parcours. Proposer une condition nécessaire.
5. \* Proposer une condition nécessaire et suffisante (une caractérisation).

### 4 Flots et parité

On considère un réseau de transport  $R = (X, U)$  dont les capacités sont entières.

1. Un flot entier est dit pair (resp. impair) si toutes ses valeurs sont paires (resp. impaires). Est-il vrai que :
  - (a) si toutes les capacités sont paires, il existe un flot maximum pair.
  - (b) si toutes les capacités sont impaires, il existe un flot maximum impair.
2. Est-il possible qu'il existe un flot maximum sur  $R$  qui ne soit pas à valeurs entières ?
3. Proposer un algorithme qui le transforme en un flot maximum entier.
4. \* Si les capacités sont irrationnelles, est-il possible que l'algorithme de Ford-Fulkerson standard ne se termine pas et qu'il converge vers une valeur qui ne soit pas le flot maximum ?

## 5 Encore des Flots

1. Donner un argument simple qui permet d'affirmer qu'il existe une suite d'au plus  $m$  chaînes améliorantes qui permet de passer du flot de valeur 0 au flot maximal.
2. \* Considérons maintenant l'implémentation de Ford-Fulkerson qui choisit à chaque étape la chaîne améliorante qui augmente le plus la valeur du flot. Montrer que cette variante utilise au plus  $O(m \log C)$  améliorations successives (où  $C$  représente la valeur maximale de la capacité d'un arc).
3. Comment implémenter cet algorithme ?
4. Déterminer le coût global de la méthode.