INF554: Utilisation de l'aléatoire en algorithmique — **Automne 2014**

Cours 6 — 3 novembre

Enseignant : Frédéric Magniez — *Rédacteur : Thibault GROUEIX*

6.1 Exercice 1

Input: Stream de n entiers dans $[1, n]$

Output:

1. Vérifier qu'ils ont 2 a 2 distincts
2. Vérifier qu'il y ait exactement un doublon α (β manquant).Trouver α.

Contraintes: Mémoire en $O(\log n)$, une passe.

Préliminaires

$$U(X) = \sum_{k=1}^{n} X^{a_k - 1} = \sum_{k=1}^{n} m_k \cdot X^{k-1}$$

$$V(X) = \sum_{k=1}^{n} X^{k-1}$$

Soit p premier et $p > n^2$

$$U \equiv V \pmod p \iff \forall i, j \text{ tel que } i \neq j \text{ }, a_i \neq a_j$$

$$\mathbb{P}_{a \in \{1 \dots n\}} (U(a) \equiv V(a) \pmod p) \leq \frac{1}{n}$$

Algorithme Solution 1

$f_u = f_v = i = 0$

$n^2 \leq p \leq 2 \cdot n^2$ p premier

choisir b dans {0,...,p-1 } au hasard

faire tant que stream $\neq \emptyset$:
— $f_u \equiv f_u + b^{a_i} \pmod p$
— $f_v \equiv f_v + b^i \pmod p$
— i = i+1

Si $f_u = f_v$ Alors ACCEPT Sinon REJECT

Algorithme Solution 2

$f_u = f_v = i = 0$

$S_1 = S_2 = 2$

$n^2 \leq p \leq 2 \cdot n^2$ p premier

choisir b dans {0,...,p-1 } au hasard

faire tant que stream $\neq \emptyset$:
— $S_1 = S_1 + a_i$
— $S_2 = S_2 + a_i^2$
— $f_u \equiv f_u + b^{a_i} \pmod{p}$
— $f_v \equiv f_v + b^i \pmod{p}$
— i = i+1

$$\begin{aligned} S_1 &= \frac{n(n+1)}{2} + \alpha - \beta \\ S_2 &= \frac{n(n+1)(2n+1)}{6} + \alpha^2 - \beta^2 \end{aligned}$$

Retourner l'unique (car $\alpha \neq \beta$) solution du système, puis :

$f_v \equiv f_v - b^{\alpha-1} + b^{\beta-1} \pmod{p}$

Si $f_u = f_v$ Alors ACCEPT Sinon REJECT

6.2 Exercice 2 : Graphes

Input Stream: Arêtes d'un graphe non orienté à n sommets (n connu) dans un ordre arbitraire.
Exemple : $\{2,3\}, \{1,4\}, \{5,2\}, ...$

Output: Couplage M de plus grande taille possible.

Mémoire : $O(n \times \text{polylog}(n)) = O(n \times \log^{O(1)}(n))$

Définition 6.1. *M est une C-approximation de Max Cardinality Graph Matching si $|M| \geq C \times |M^*|$ où M^* est un couplage de plus grande taille du graphe.*

Théorème 6.2. *Il existe un algorithme déterministe à 1 passe, de mémoire $O(n \log n)$ qui renvoie une $\frac{1}{2}$-approximation de Max Cardinality Graph Matching.*

Théorème 6.3. (2012) *Il n'existe pas d'algorithme en 1 passe et mémoire $O(n \times \text{polylog}(n))$ qui retourne mieux qu'une $\frac{2}{3}$-approximation (même probabiliste).*

Théorème 6.4. (2013) *Idem : pas mieux qu'une $(1 - 1/e)$-approximation.*

Remarque : Cette borne reste vraie dans le modèle "vertex arrival order".

ALGORITHME GLOUTON

$M \leftarrow \emptyset$

Tant que Stream non vide

$e \leftarrow$ lire arête du Stream

Si $M \cup \{e\}$ est encore un couplage

$M \leftarrow M \cup \{e\}$

Retourner M

Mémoire : $O(n \log n)$ car $|M| \leq \frac{n}{2}$.

Lemme 6.5. *M est maximal pour l'inclusion.*
C'est à dire que $\forall e \in G$:

- *Soit $e \in M$*
- *Soit $M \cup \{e\}$ n'est pas un couplage.*

Preuve: Vrai par construction de M.
On prouve par induction qu'à chaque instant, M est maximal pour les arêtes déjà lues. □

Lemme 6.6. *Soit M^* un couplage de plus grande taille. Soit M un couplage maximal pour l'inclusion. Alors*

$$|M| \geq \frac{1}{2}|M^*|$$

Notations On note M^* un couplage de plus grande taille du graphe G.
Pour M couplage quelconque, on note $V(M)$ l'ensemble des sommets couverts par M.

Preuve: f fonction de M* dans $V(M)$
si $x \in V(M)$ alors $f(e = \{x, y\}) = x$
sinon $f(e = \{x, y\}) = y$

f est définie car M est maximal
f est injective car M* est un couplage
Or $|V(M)| = 2 \cdot |M|$ Donc $|M| \geq \frac{1}{2} \cdot |OPT|$ □

La combinaison des deux lemmes précédents prouve le **Théorème 5.2**.

Remarque La démonstration précédente n'utilise pas le fait que M* est le couplage optimal pour l'inclusion. Elle reste donc valide pour tout couplage.

6.3 MATCHING Avec Poids

Les aretes ont toutes un poids $w_e \geq 0$

Input: Stream de (e, w_e)

Output: MATCHING M de poids maximal

Le but est de trouver une $\frac{1}{4+\epsilon}$ -approximation où ϵ est une constante donnée, en mémoire $O(\frac{1}{\epsilon} \cdot n \cdot \text{polylog}(n))$.

Question 1 Soit $w_{max} = \max w_e$

$$F = \{e \in E, w_e \geq \frac{2 \cdot \epsilon \cdot w_{max}}{n}\}$$

OPT(E) = un couplage de poids maximal dans E
OPT(F) = un couplage de poids maximal dans F
Montrer que $(1 + \epsilon)\rho(OPT(F)) \geq \rho(OPT(E))$

$$H = \{e \in E, w_e \leq \frac{2 \cdot \epsilon \cdot w_{max}}{n}\}$$

$$\rho(H) \leq \frac{n}{2} \cdot \frac{2 \cdot \epsilon \cdot w_{max}}{n}$$

$$\rho(H) \leq \epsilon \cdot w_{max}$$

$e_{max} \in F \implies \rho(OPT(F)) \geq w_{max}$

$$\rho(OPT(F)) \geq \rho(OPT(E)) - \rho(H)$$

$$\rho(OPT(F)) \geq \rho(OPT(E)) - \epsilon \cdot w_{max}$$

$$\rho(OPT(F)) \geq \rho(OPT(E)) - \epsilon \cdot \rho(OPT(F))$$

Finalement,

$$(1 + \epsilon)\rho(OPT(F)) \geq \rho(OPT(E))$$

Question 2
$\forall i \in \mathbb{Z}, F_i = \{e \in F : \rho(e) \geq (1 + \epsilon)^i\}$
$\forall i \in \mathbb{Z}, F_{i+1} \subset F_i$
Montrer que F_i prend des valeurs non triviales pour $\theta(\frac{1}{\epsilon} \cdot \log(n))$ valeurs a préciser.

On cherche α et β tel que $\alpha \leq \beta$,

$$F_\alpha = F, F_{\alpha+1} \neq F$$

$$F_\beta = \emptyset, F_{\beta-1} \neq F$$

On a d'une part,

$$(1 + \epsilon)^\alpha \leq \frac{w_{max} \cdot 2 \cdot \epsilon}{n}$$

$$\alpha = \frac{log(\frac{w_{max} \cdot 2 \cdot \epsilon}{n})}{log(1+\epsilon)}$$

et d'autre part,

$$(1+\epsilon)^{\beta} \geq w_{max}$$

$$\beta = \frac{log(w_{max}}{log(1+\epsilon)}$$

Ainsi,

$$\alpha - \beta = \frac{log(\frac{2 \cdot \epsilon}{n})}{log(1+\epsilon)}$$

$$\alpha - \beta = \theta(\frac{1}{\epsilon} \cdot log(n))$$

Algorithme MATCHING

$\forall \alpha \leq i \leq \beta,\ C_i = \emptyset$

faire tant que stream $\neq \emptyset$:
— lire la prochaine arete (e, w_e)
— $\forall \alpha \leq i \leq \beta,$ tel que $w_e \geq (1+\epsilon)^i$:
— Si $C_i \cap e = \emptyset$ alors $C_i = C_i \cup e$

Algorithme glouton sur $(C_\beta, ..., C_\alpha)$. On obtient un couplage T

Question 3
Posons $T_i = T \cap F_i$
Montrer que

$$\forall\ i, |T_i| \geq \frac{1}{4} \cdot |OPT \cap F_i|$$

Notons OPT'_i le couplage de taille maximale dans F_i Nous avons C_i un couplage maximale dans F_i, donc :

$$|C_i \geq \frac{1}{2} \cdot |OPT'_i|$$

De plus, OPT$\cap F_i$ est un couplage de F_i donc,

$$|OPT'_i| \geq |OPT \cap F_i|$$

Enfin remarquons que par construction, T_i est un couplage maximal dans $\bigcup_{j \in [i,\beta]} C_j$, donc :

$$|T_i \geq \frac{1}{2} \cdot |C_i|$$

Ainsi :

$$\forall\ i, |T_i| \geq \frac{1}{4} \cdot |OPT \cap F_i|$$

Question 4
Montrer qu'il existe une fonction f de OPT dans T telle que :
- $\forall\ e \in OPT,\ \rho(e) \leq (1+\epsilon)\cdot\rho(f(e))$
- $\forall\ t \in T,\ |f^{-1}(t)| \leq 4$

On construit une fonction en prenant les aretes de OPT qui apparaissent successivement dans les $OPT \cap F_i$ décroissants.
- A chaque element de $OPT \cap F_i$, on associe une arete de T_i qui a moins de 4 antécédents. La question 3 assure que cela est possible.
- les aretes de $OPT \cap F_i$ qui n'ont pas été affectées a l'étape $OPT \cap F_{i+1}$, ont donc un poids compris entre $(1+\epsilon)^i$ et $(1+\epsilon)^{i+1}$. Or ils sont associés a des éléments de T_i donc de F_i de poids supérieur a $(1+\epsilon)^i$. On en déduit l'inégalité demandée :

$$\forall\ e \in OPT,\ \rho(e) \leq (1+\epsilon)\cdot\rho(f(e))$$

Finalement,

$$\rho(OPT(E)) = \sum_{e \in OPT} \rho(e)$$
$$\rho(OPT(E)) \leq (1+\epsilon)\cdot\sum_{e \in OPT} \rho(f(e))$$
$$\rho(OPT(E)) \leq (1+\epsilon)\cdot 4\cdot\sum_{e \in T} \rho(e))$$
$$\rho(OPT(E)) \leq (1+\epsilon)\cdot 4\cdot\rho(T)$$

Quitte a prendre $\epsilon = \frac{1}{4}\cdot\epsilon$, nous avons bien :

$$\rho(T) \geq \frac{1}{4+\epsilon}\cdot\rho(OPT)$$

Nous avons bien une complexité en mémoire de taille $(\frac{1}{\epsilon}\cdot n\cdot \text{polylog}(n))$