



Master2 Informatique

UE: ALGORITHMIQUE DISTRIBUÉES – 4TIN913U

Responsable : M. Gavaille

Date : 8 janvier 2025

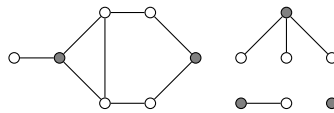
Durée : 1h30

Document: une feuille A4 recto-verso autorisée.

## Ensemble dominant

Un *ensemble dominant* d'un graphe  $G$  est un ensemble  $S$  de sommets de  $G$  tel que, pour tout sommet  $u \in V(G)$ , soit  $u \in S$  soit  $u$  a un voisin  $v \in S$ . Comme  $S = V(G)$  constitue trivialement un ensemble dominant pour  $G$ , on essaye généralement d'en trouver un ayant peu de sommets (dans la mesure du possible).

Par exemple, deux sommets opposés d'un cycle de longueur six forment un ensemble dominant. Pour une étoile (un arbre de hauteur un), un seul sommet suffit : la racine. Notez qu'un sommet isolé dans  $G$ , c'est-à-dire de degré 0, appartient à tout ensemble dominant de  $G$ . Voir aussi les exemples ci-dessous.



On considère l'algorithme appelé FOREST, qui a pour vocation de calculer grâce à une règle locale un petit ensemble dominant, en tout cas dans certains graphes. Son principe est le suivant :

**FOREST** « Un sommet de  $G$  est sélectionné dans l'ensemble  $S$  si son degré est au moins deux. »

On supposera le modèle LOCAL où chaque sommet  $u$  de  $G$  ne peut utiliser que les deux primitives de communication suivantes :

- SENDALL( $m$ ) : envoie le message  $m$  à tous les voisins de  $u$ .
- RECEIVEALL() : renvoie la liste des messages reçus par  $u$  et émis à la ronde courante.

Notez bien qu'un sommet  $u$  n'a pas accès directement à la liste de ses voisins dans  $G$ . Mais il peut calculer cette liste grâce à son identifiant  $ID(u)$  et aux deux primitives ci-dessus. Vous pourrez utiliser NEWROUND pour d'indiquer le démarrage d'une nouvelle ronde.

**Question 1.** Décrivez l'algorithme FOREST dans le modèle LOCAL et le style du cours en précisant clairement les rondes. Vous supposerez que chaque sommet  $u$  possède une variable locale SELECT  $\in \{\text{TRUE}, \text{FALSE}\}$  indiquant si  $u \in S$  ou pas.

**SOLUTION.** Le code de tout sommet  $u$  ne possède qu'une seule ronde qui est :

NEWROUND

- SENDALL( $ID(u)$ )
- SELECT := ( $|RECEIVEALL()| > 1$ )

**Question 2.** Démontrez que l'algorithme FOREST est correct, c'est-à-dire qu'il calcule bien, via les variables locales SELECT, un ensemble dominant  $S$  pour  $G$ , si  $G$  est connexe et possède au moins trois sommets.

**SOLUTION.** Tous les sommets de  $G$  sont dans  $S$ , sauf ceux de degré un,  $G$  ne pouvant avoir de sommet de degré isolé puisqu'il est connexe et possède  $> 1$  sommets. Soit  $u$  un sommet ayant un seul voisin  $v$ . Comme  $u$  et  $v$

sont dans une composante connexe avec au moins trois sommets,  $\deg(v) \geq 2$ , les cas  $\deg(v) \in \{0, 1\}$  étant impossibles. En particulier,  $v \in S$  et donc domine  $u$ . Donc  $S$  domine bien tous les sommets.

**Question 3.** Comme suggéré par la question 2, expliquez pourquoi l'algorithme FOREST n'est pas correct pour certains graphes.

SOLUTION. Si  $G$  possède des composantes connexes avec un ou deux sommets. Dans ces deux cas, aucun sommet ne sera sélectionné ce qui est incorrect, il faudrait en sélectionner au moins un.

On dira qu'un ensemble dominant  $S$  pour  $G$  est *non trivial* si  $S \neq V(G)$ . Il est facile de voir que  $G$  possède un ensemble dominant non trivial si et seulement si  $G$  possède au moins une arête.

**Question 4.** Modifier votre algorithme de la question 1 de façon à ce qu'il calcule, pour toute forêt  $G$ , un ensemble dominant non trivial lorsque c'est possible.

SOLUTION.

(code du sommet  $u$ )

NEWROUND

- SENDALL(ID( $u$ ))
- $L := \text{RECEIVEALL}()$
- SELECT :=  $(|L| \geq 2) \vee (|L| = 0) \vee ((|L| = 1) \wedge (\text{ID}(u) < L[0]))$

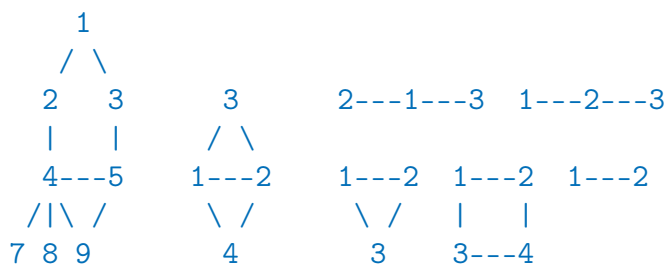
On considère maintenant un second algorithme, appelé OUTER, dont le principe est le suivant :

**OUTER** « Chaque sommet détermine son degré qu'il envoie à ses voisins. Puis, un sommet est sélectionné dans l'ensemble  $S$  si son degré est au moins 4, ou bien si tous ses voisins sont de degré plus petit que 4. »

Cet algorithme est censé être efficace pour les graphes *planaires-extérieurs*, c'est-à-dire les graphes pouvant être dessinés dans le plan sans croisement d'arêtes où tous les sommets appartiennent au bord de la face extérieure. Les étoiles, arbres, forêts et tous ceux de l'exemple initial sont planaires-extérieurs. On admettra, sans le démontrer, que tout graphe planaire-extérieur possède toujours un sommet de degré au plus 2.

**Question 5.** Ajoutez des règles locales supplémentaires à OUTER permettant de garantir, pour tout graphe planaire-extérieur  $G$ , le calcul d'un ensemble dominant non trivial pour  $G$ , si c'est possible. [Aide : Attention aux sommets de petits degrés (chemins, cycles).] Justifiez. Combien de rondes sont nécessaires en tout ? [NB : On ne demande pas le code précis, la description des règles et leur validité suffisent.]

SOLUTION. Attention aux cas comme ceux-ci :



Voici les règles que l'on peut ajouter en plus de celles sur les degré 4, les sommets isolés étant déjà gérés par ces règles, l'ejeux étant d'arriver à supprimer au moins un sommet de degré 2 par composantes connexes avec plus de trois sommets :

[R0] Délectionner  $u$  si son degré est 1 et que son voisin aussi et qu'il a le plus petit identifiant. (Arête isolée).

[R1] Désélectionner  $u$  si son degré est 2 et qu'après l'application des règles précédentes, ses deux voisins sont sélectionnés, et qu'au moins un de ses voisins a un degré qui n'est pas 2. (Ici les cycles gardent tous leurs sommets.)

[R2] Désélectionner  $u$  si son degré est 2 et qu'après l'application de R1 ses deux voisins sont encore sélectionnés, ses deux voisins sont de degré 2 et qu'il a le plus petit identifiant. (Chaîne de trois sommets de degrés 2.)

Seules les règles R1 et R2 posent potentiellement des difficultés. On remarque cependant que R1 ne peut s'appliquer qu'on mieux pour deux sommets voisins  $u-v$  de degré 2. Et dans ce cas, leurs deux autres voisins seront nécessairement sélectionnés et pas de degré 2 (donc ne pourront pas appliquer la règle R1). Pour R2, il n'y a pas deux sommets voisins qui peuvent l'appliquer (à cause du plus petit identifiant). Donc R1 et R2 suppriment des sommets tout en gardent un ensemble dominant.

On peut aussi vérifier que dans toute composante connexe, au moins un sommet de degré 2 sera éliminé : c'est bien le cas, si ni R1 ni R2 s'applique. C'est encore le cas si R1 ou R2 s'appliquent.

À priori, il suffit de deux rondes : une pour le calcul du degré, une autre pour l'échange de ce degré à ses voisins, les règles R1 et R2 pouvaient être exprimées en une seule puisqu'elles dépendent que du degré des voisins.

On peut montrer que l'algorithme FOREST produit, si les composantes connexes de  $G$  sont assez grandes, des ensembles dominants de taille optimale à un facteur 3 près si  $G$  est une forêt, et pour OUTER à un facteur 5 près si  $G$  est un graphe planaire-extérieur. On peut aussi démontrer qu'aucun algorithme distribué utilisant un nombre constant de rondes ne peut faire mieux en terme de facteur d'approximation pour les forêts et les planaires-extérieurs.

Soit  $Q_n$  le graphe planaire-extérieur obtenu à partir d'un chemin  $P_n$  à  $n$  sommets et en ajoutant toutes les arêtes entre sommets à distance deux dans  $P_n$ , voir l'exemple ci-après.



**Question 6.** Montrez que pour tout  $n$ ,  $P_n$  possède un ensemble dominant d'au plus  $\lceil n/3 \rceil$  sommets.

SOLUTION. En numérotant les sommets de  $P_n$  contenu dans  $Q_n$ ,  $v_1, \dots, v_n$ , il suffit de prendre les  $i$  sommets suivant :  $v_2, v_5, v_8, \dots, v_{2+3(i-1)} = v_{3i-1}$  où  $i$  est le plus petit indice tel que  $3i - 1 \geq n - 1$ . (Si le dernier sommet sélectionné  $v_k$  a un indice  $k < n - 1$ , le sommet  $v_n$  ne sera pas dominé.) Il suit que  $i \geq n/3$ , et donc le nombre recherché vaut précisément  $i = \lceil n/3 \rceil$ .

**Question 7.** Montrez que pour tout  $n$ ,  $Q_n$  possède un ensemble dominant d'au plus  $\lceil n/5 \rceil$  sommets.

SOLUTION. De manière similaire, il suffit de prendre les  $i$  sommets suivant :  $v_3, v_8, v_{13}, \dots, v_{3+5(i-1)} = v_{5i-2}$  où  $i$  est le plus petit indice tel que  $5i - 2 \geq n - 2$ . (Si le dernier sommet sélectionné  $v_k$  a un indice  $k < n - 2$ , le sommet  $v_n$  ne sera pas dominé.) Il suit que  $i \geq n/5$ , et donc le nombre recherché vaut précisément  $i = \lceil n/5 \rceil$ .

**Question 8.** En supposant  $n$  assez grand, combien de sommets de  $P_n$  sont sélectionnés au pire par l'algorithme FOREST modifié comme à la question 4 ? Même question pour l'algorithme OUTER modifié comme à la question 5 pour le graphe  $Q_n$  ? Justifiez.

SOLUTION. Pour FOREST sur  $P_n$ , c'est  $n - 2$  car les deux sommets de degré 1 ne seront pas sélectionnés. Pour OUTER, c'est aussi  $n - 2$  car les deux sommets de degré trois dans  $Q_n$  ne seront pas sélectionnés (car voisins de sommets de degré 4). Du coup, la règle R1 ou R2 ne va pas s'appliquer sur les deux autres sommets de degré 2.

# Complexité de communication

**Question 9.** Expliquez ce qu'est un protocole de communication qui calcule une fonction booléenne  $f$  ? Quelles en sont les principales règles ? Quel est l'objectif recherché ?

**SOLUTION.** C'est un algorithme qui permet à deux participants, Alice et Bob, d'échanger des messages jusqu'à obtenir la valeur  $f(x, y)$  où  $x, y$  sont respectivement les entrées d'Alice et Bob. Ils s'échangent des messages chacun à leur tour, Alice commençant, et le dernier bit doit être la réponse. Bien sûr, chacun ne connaît que son entrée et le protocole. L'objectif est de minimiser le nombre total de bits échangés.

**Question 10.** À quel problème d'algorithmique distribué vu en cours la complexité de communication nous a-t-elle servi ? Citez le résultat.

**SOLUTION.** On s'est servi de la complexité de communication pour démontrer que dans le modèle  $b$ -CONGEST, il fallait  $\Omega(\sqrt{n}/b)$  rondes dans le pire cas pour détecter un  $C_4$  dans un graphe à  $n$  sommets.

On considère la fonction booléenne  $\text{pand}_n(x, y) \in \{0, 1\}$  qui étant donnés deux mots binaires  $x, y$ , chacun de  $n$  bits, renvoie 1 si et seulement si le mot  $x \wedge y$  (ET-bit-à-bit) contient un nombre impair de 1. Dit autrement  $\text{pand}_n(x, y)$  renvoie la parité du nombre de 1 de  $x \wedge y$ .

Par exemple,  $\text{pand}_6(110111, 011011) = 1$  car  $110111 \wedge 011011 = 010011$  qui contient un nombre impair de 1. On a aussi  $\text{pand}_6(110111, 011010) = 0$ . Notez que  $\text{pand}_n(2^i, 2^j) = 1$  si et seulement si  $i = j$ . On notera  $A(n)$  la matrice de la fonction  $\text{pand}_n$ .

**Question 11.** Donnez les matrices  $A(1)$  et  $A(2)$ .

**SOLUTION.**

$A(1) =$	<table style="border-collapse: collapse;"> <tr><td style="padding: 0 5px;"><math>x \setminus y</math></td><td style="padding: 0 5px;"> </td><td style="padding: 0 5px;">0</td><td style="padding: 0 5px;">1</td></tr> <tr><td style="padding: 0 5px;">0</td><td style="padding: 0 5px;"> </td><td style="padding: 0 5px;">0</td><td style="padding: 0 5px;">0</td></tr> <tr><td style="padding: 0 5px;">1</td><td style="padding: 0 5px;"> </td><td style="padding: 0 5px;">0</td><td style="padding: 0 5px;">1</td></tr> </table>	$x \setminus y$		0	1	0		0	0	1		0	1	$A(2) =$	<table style="border-collapse: collapse;"> <tr><td style="padding: 0 5px;"><math>x \setminus y</math></td><td style="padding: 0 5px;"> </td><td style="padding: 0 5px;">00</td><td style="padding: 0 5px;">01</td><td style="padding: 0 5px;">10</td><td style="padding: 0 5px;">11</td></tr> <tr><td style="padding: 0 5px;">00</td><td style="padding: 0 5px;"> </td><td style="padding: 0 5px;">0</td><td style="padding: 0 5px;">0</td><td style="padding: 0 5px;">0</td><td style="padding: 0 5px;">0</td></tr> <tr><td style="padding: 0 5px;">01</td><td style="padding: 0 5px;"> </td><td style="padding: 0 5px;">0</td><td style="padding: 0 5px;">1</td><td style="padding: 0 5px;">0</td><td style="padding: 0 5px;">1</td></tr> <tr><td style="padding: 0 5px;">10</td><td style="padding: 0 5px;"> </td><td style="padding: 0 5px;">0</td><td style="padding: 0 5px;">0</td><td style="padding: 0 5px;">1</td><td style="padding: 0 5px;">1</td></tr> <tr><td style="padding: 0 5px;">11</td><td style="padding: 0 5px;"> </td><td style="padding: 0 5px;">0</td><td style="padding: 0 5px;">1</td><td style="padding: 0 5px;">1</td><td style="padding: 0 5px;">0</td></tr> </table>	$x \setminus y$		00	01	10	11	00		0	0	0	0	01		0	1	0	1	10		0	0	1	1	11		0	1	1	0
$x \setminus y$		0	1																																										
0		0	0																																										
1		0	1																																										
$x \setminus y$		00	01	10	11																																								
00		0	0	0	0																																								
01		0	1	0	1																																								
10		0	0	1	1																																								
11		0	1	1	0																																								

**Question 12.** Démontrez que, pour tout  $n \geq 1$ ,  $A(n+1) = \begin{bmatrix} A(n) & A(n) \\ A(n) & \bar{A}(n) \end{bmatrix}$  où  $\bar{A}(n)$  représente la matrice  $A(n)$  où tous les bits ont été complétés. En déduire que  $A(n+1)$  possède une diagonale de zéros.

**SOLUTION.** [3] Les quatre cadrants de  $A(n+1)$  correspondent aux mots  $x', y' \in \{0, 1\}^{n+1}$  de la forme :  $(0x, 0y)$ ,  $(0x, 1y)$ ,  $(1x, 0y)$ ,  $(1x, 1y)$  avec  $x, y \in \{0, 1\}^n$ . Il est clair que la parité des trois premières sous-matrices ne peut changer par rapport à  $A(n)$  puisque figure toujours un 0 dans comme premier bit de  $x'$  ou de  $y'$ . Donc les trois premières sous-matrices valent  $A(n)$ . Pour la dernière, la parité est inversée puisque  $x'$  et  $y'$  commencent tous les deux par 1, et donc elle vaut  $\bar{A}(n)$ .

Il est facile de vérifier par induction que  $A(n+1)$  possède donc une diagonale montante de 0. C'est effectivement vrai pour  $n = 1$ . Puis, dans  $A(n+1)$  on a deux matrices  $A(n)$  alignées sur la même diagonale, ce qui forme donc bien une diagonale de 0 dans  $A(n+1)$ .

On rappelle qu'un ensemble discriminant pour une fonction booléenne  $f$  et une valeur  $v \in \{0, 1\}$  est un ensemble  $F_v$  d'entrées  $(x, y)$  pour  $f$  tel que, pour tout  $(x, y) \neq (x', y') \in F_v$ ,  $f(x, y) = f(x', y') = v$  et  $\bar{v} \in \{f(x, y'), f(x', y)\}$ , où  $\bar{v} = 1 - v$  est la valeur complémentaire de  $v$ . Cela revient donc à dire qu'on ne peut pas avoir  $f(x, y') = f(x', y) = v$ . On notera  $CC(f)$  la complexité de communication de  $f$ .

**Question 13.** Démontrez que si  $f$  possède un ensemble discriminant  $F_v$  tel que  $|F_v| = 2^k$ , pour un certain entier  $k \geq 1$  et une valeur  $v \in \{0, 1\}$ , alors  $CC(f) \geq k + 1$ . [Aide : Utilisez les liens établis dans le cours entre  $CC(f)$ , le nombre de scripts et la cardinalité des ensembles discriminants pour  $f$ .]

SOLUTION. [3] Par définition de  $F_v$ , la matrice de  $f$  doit contenir au moins une entrée  $(x_0, y_0)$  avec  $f(x_0, y_0) = \bar{v}$ , sinon  $F_v$  ne serait pas discriminant de cardinalité  $2^k \geq 2$ . Donc cette entrée  $(x_0, y_0)$  forme un ensemble discriminant  $\{(x_0, y_0)\}$  pour  $\bar{v}$ . D'après le cours,  $CC(f) \geq \log_2 \sigma(f)$ , le nombre de script minimum pour  $f$ , et  $\sigma(f) \geq |F_v| + |\{(x_0, y_0)\}| = 2^k + 1$ . Et donc  $CC(f) \geq \log_2(2^k + 1) > \log_2(2^k) = k$ . Comme  $CC(f)$  et  $k$  sont des entiers, cela implique que  $CC(f) \geq k + 1$ .

**Question 14.** Soient  $F_v$  et  $F'_v$  deux ensembles discriminants pour une fonction  $f$  et la valeur  $v$ . Est-il vrai que  $F_v \cup F'_v$  est aussi un ensemble discriminant pour  $f$  et la valeur  $v$ ? Justifiez.

SOLUTION. [3] De manière générale, c'est faux. Pour  $\text{pand}_2$  par exemple, on peut considérer  $F_0 = \{(11, 00), (10, 01)\}$  et  $F'_0 = \{(01, 10), (00, 11)\}$ . Pourtant, les entrées  $\{(11, 00), (00, 11)\} \subset F_0 \cup F'_0$  ne violent pas la propriété du rectangle.

**Question 15.** Donnez un ensemble discriminant  $F_0$  pour  $\text{pand}_2$  le plus grand possible, sans démontrer l'optimalité.

SOLUTION. Il y a plusieurs solutions possibles. Par exemple, sur la diagonale montante de zéros,  $F_0 = \{(11, 00), (10, 01), (01, 10)\}$ . Attention ! on ne pouvait pas mettre le 4e élément diagonal.

**Question 16.** De manière générale, démontrez que pour tout  $n \geq 1$ ,  $\text{pand}_n$  possède un ensemble discriminant de cardinalité au moins  $n$ . Justifiez le caractère discriminant de cet ensemble. [Aide : Pensez aux entrées puissances de deux.]

SOLUTION. [3] On prend  $F_1 = \{(2^i, 2^i) : i \in \{0, \dots, n-1\}\}$ , c'est-à-dire des entrées situées sur la diagonale descendante composée d'un seul bit. Clairement,  $\text{pand}_n(2^i, 2^j) = 1$  si et seulement si  $i = j$ . Dit autrement,  $F_1$  est discriminant pour la valeur 1. De plus  $|F_1| = n$  comme demandé.

Parenthèse : Pouvait-on faire plus grand que  $n$ ? On aurait pu, par exemple, considérer un ensemble maximum  $I_3$  de triplets d'indices  $\{i, j, k\}$  deux à deux disjoints (en fait s'intersectant sur un nombre impaires de positions : 0 ou 2 donc ici). En prenant  $F_1 = \{(x, x) : x = 2^i + 2^j + 2^k, \{i, j, k\} \in I_3\}$ , on avait également que  $F_1$  était discriminant pour la valeur 1, puisque ces éléments diagonaux ont  $\text{pand}_n = 1$  et les autres  $\text{pand}_n = 0$ . Et de même pour  $I_5, I_7, \dots$ , l'exemple demandé était pour  $I_1$ . Seulement le calcul de  $I_3$  n'est pas facile. C'est la taille d'une clique maximum du graphe de Kneser  $K(n, 3)$  où les sommets sont les sous-ensembles de  $\{1, \dots, n\}$  ayant  $3 = k$  éléments, et sont adjacents si leur intersection est vide. Il se trouve qu'il est connu que cette taille est  $\lfloor n/k \rfloor$ , ce qui est maximal quand  $k = 1 \dots$  Donc pas intéressant.

**Question 17.** Démontrez que  $CC(\text{pand}_{2^k}) \geq k + 1$  pour tout entier  $k \geq 1$ .

SOLUTION. [3] D'après les questions précédentes 16 et 13, la fonction possède un ensemble discriminant de cardinalité  $2^k$ . Et donc  $CC(\text{pand}_{2^k}) \geq k + 1$ .

Dans la suite on ne considèrera que des protocoles de communications sans-préfixe et sans mot vide. Soit  $f_0$  la fonction booléenne définie sur les mots binaires de 2 bits et dont la matrice est la suivante :

	y	00	01	10	11
x	00	1	1	0	0
01	1	1	0	0	
10	0	0	1	1	
11	0	0	1	1	

**Question 18.** Montrez que tout protocole de communication pour  $f_0$  nécessite au moins 4 scripts.

SOLUTION. [3] On peut construire deux ensembles discriminants  $F_0$  et  $F_1$  chacun de taille 2 (en considérant les deux 0 et deux 1 diagonaux centraux de la matrice). D'après le cours, le nombre de scripts doit être au moins de  $|F_0| + |F_1| = 4$ .

**Question 19.** Donnez un protocole de communication pour  $f_0$  de coût 2. [Son principe suffit.]

SOLUTION. [3] Alice indique à Bob si on entrée  $x$  commence par un 0 ou un 1. Puis Bob, avec  $y$ , détermine le numéro de colonne de la matrice, et en déduit  $f(x, y)$  qu'il retransmet à Alice.

On dira qu'une fonction booléenne  $f$  sur des mots de  $n$  bits est  $k$ -similaire si la matrice de  $f$  comporte exactement  $k$  lignes différentes (parmi les  $2^n$  possibles). Par exemple, la fonction  $f_0$  ci-dessus est 2-similaire.

**Question 20.** Proposez un protocole de communication pour une fonction  $f$  qui est  $k$ -similaire. Précisez, en fonction de  $k$  (éventuellement de  $n$ ), le nombre de scripts de votre protocole, ainsi que son coût. [Aide : Retrouver les résultats des questions 18 et 19.]

SOLUTION. [3] Il suffit à Alice de transmettre le numéro  $i$  de ligne  $x$  (parmi les  $k$  distinctes de la matrice) et Bob en déduit avec son entrée  $y$  la valeur  $b$  qu'il transmet à Alice. Cela suffit car ils connaissent tout deux la matrice. Dans ce protocole, les scripts sont donc de la forme  $(i, b)$  où  $i \in \{1, \dots, k\}$  et  $b \in \{0, 1\}$ , ce qui fait  $2k$  scripts différents en tout. Le coût est aussi de  $\lceil \log_2(k) \rceil + 1$  bits, puisqu'il suffit de  $\lceil \log_2(k) \rceil$  bits pour décrire  $i$  et de 1 bit pour  $b$ .