

Systemes Distribués

Cours 2

Alain Cournier

MIS, UPJV



PLAN

- Introduction
- Système distribué
- Algorithmique distribuée



Introduction

- Réseaux d'ordinateurs : pb particuliers
 - Programmes coopératifs
 - BD (modèle relationnel ?)
 - Sécurité
 - ...
- Prog :

Algorithmique dans les systèmes communicants (réseaux fixes, à mobilité passive – MANET, VANET... – , à mobilité active – cohortes de robots... –)
- Ce cours : limité aux réseaux fixes, sans panne



Systeme distribue

Systemes distribues =
modele(s) pour les reseaux d'ordinateurs

Grappe :

- 1 noeud = 1 machine (processeur, processus...)
- 1 arête = 1 lien de communication entre 2 machines



Systeme distribue

Topologie

Graphe

- fini (nb de noeuds et de liens fini)
- non oriente :
 - pas d'orientation sur les liens entre noeuds (arêtes) : communication possible dans les 2 sens
- simple :
 - au plus 1 arête entre 2 noeuds
 - pas de boucle (arête dont les extremités sont un même noeud)
- et connexe :
 - entre 2 noeuds distincts il existe toujours un chemin (suite d'arêtes) qui les joint

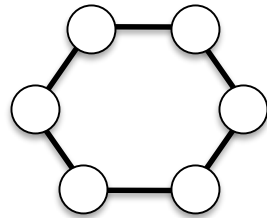


Système distribué

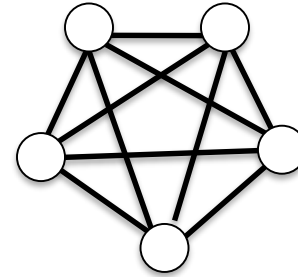
Topologie

Exemples

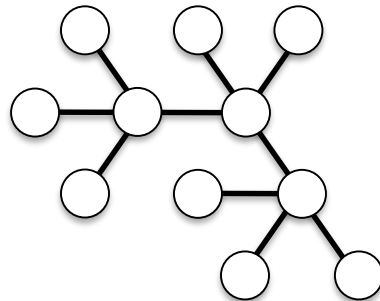
Anneau :



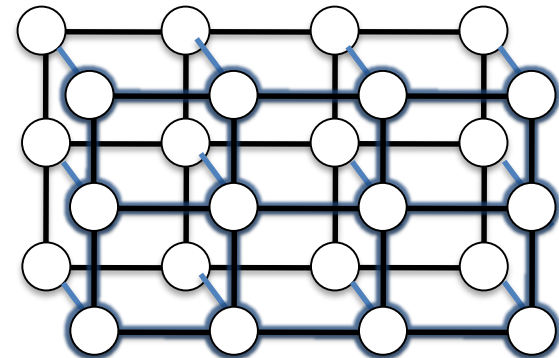
Graphe complet :



Arbre :



Grille :

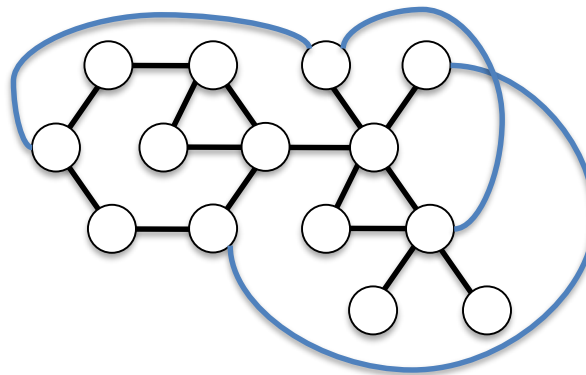
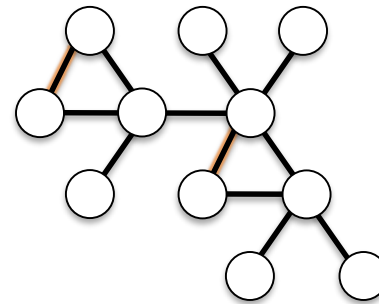
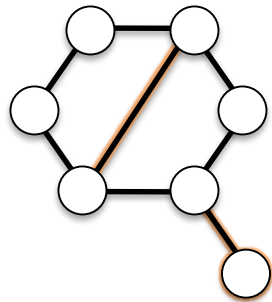


Système distribué

Topologie

Exemples

Graphes quelconques :

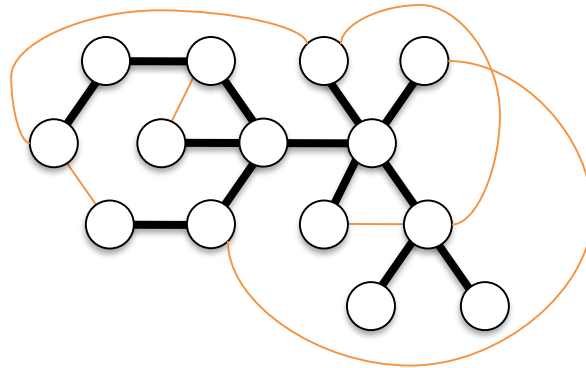


Systeme distribue

Topologie

Attention une topologie peut etre virtuelle

Ex : Arbre couvrant sur graphe quelconque :



Systeme distribue

Memoire locale

Pas de memoire commune :

Informations communiquees par messages

Les proc. envoient des messages a leurs voisins (relies par un lien de communication) ou a d'autres proc. de proche en proche



Systeme distribue

Asynchronisme

- Horloges locales :
Chaque proc. a une horloge qui a sa propre vitesse
- Temps d'acheminement des messages
Temps d'acheminement fini mais non borne

Systeme distribue

Liens

- Bidirectionnels :

Soient x et y 2 proc. voisins, alors x et y peuvent s'envoyer des messages par le même lien de communication

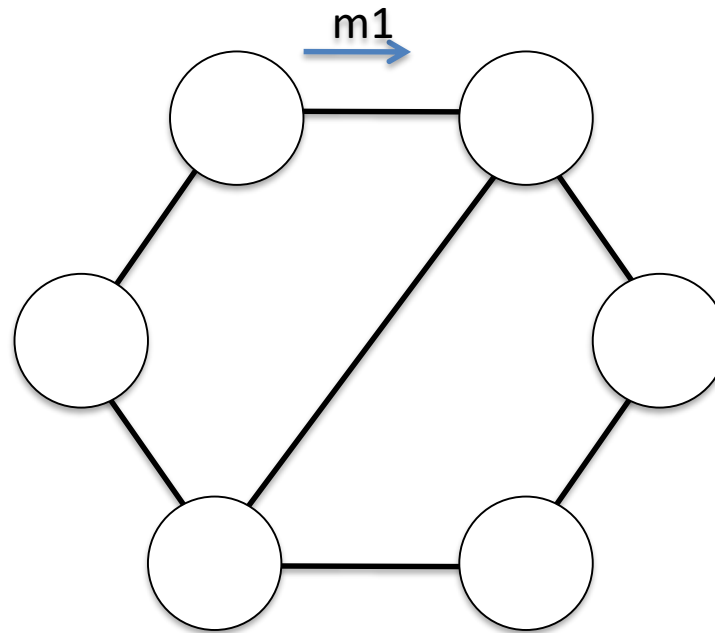
- FIFO

Les messages envoyés par x à y par leur lien de communication arriveront dans le même ordre

Système distribué

Liens

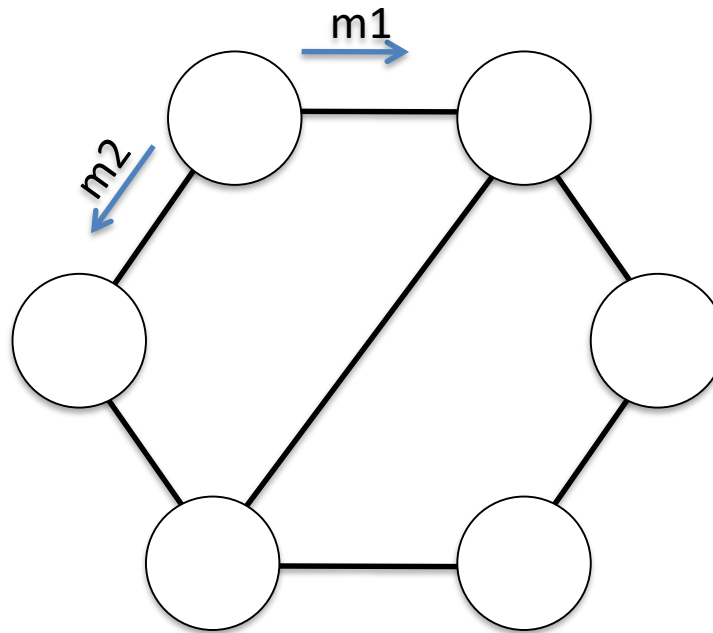
FIFO : Attention cela n'est valable que pour des messages envoyés par le même proc. et par le même lien :



Système distribué

Liens

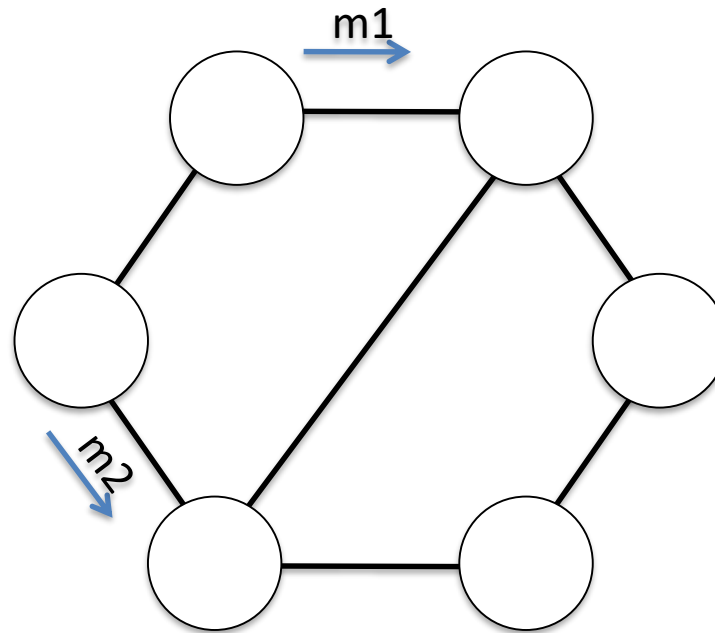
FIFO : Attention cela n'est valable que pour des messages envoyés par le même proc. et par le même lien :



Système distribué

Liens

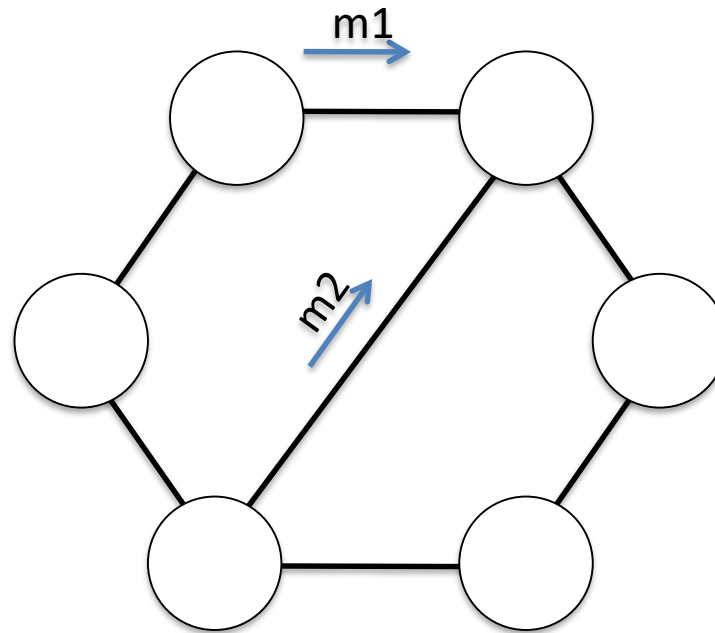
FIFO : Attention cela n'est valable que pour des messages envoyés par le même proc. et par le même lien :



Système distribué

Liens

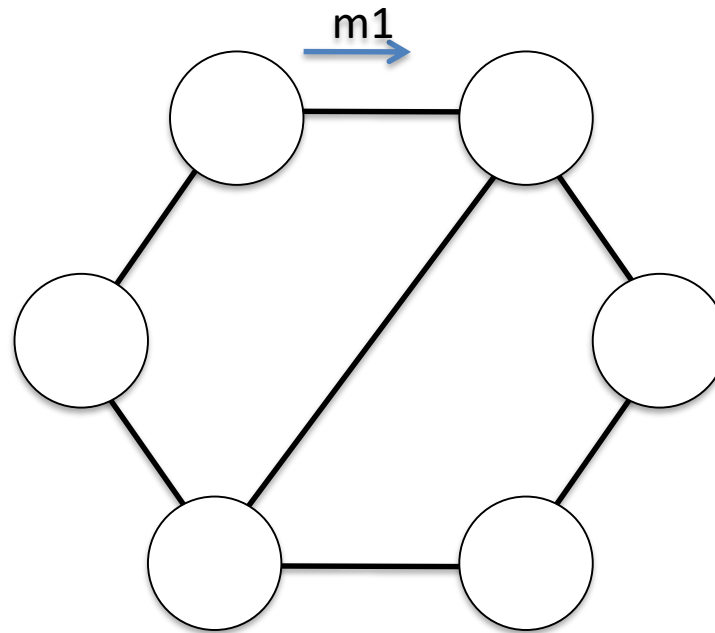
FIFO : Attention cela n'est valable que pour des messages envoyés par le même proc. et par le même lien :



Systeme distribue

Liens

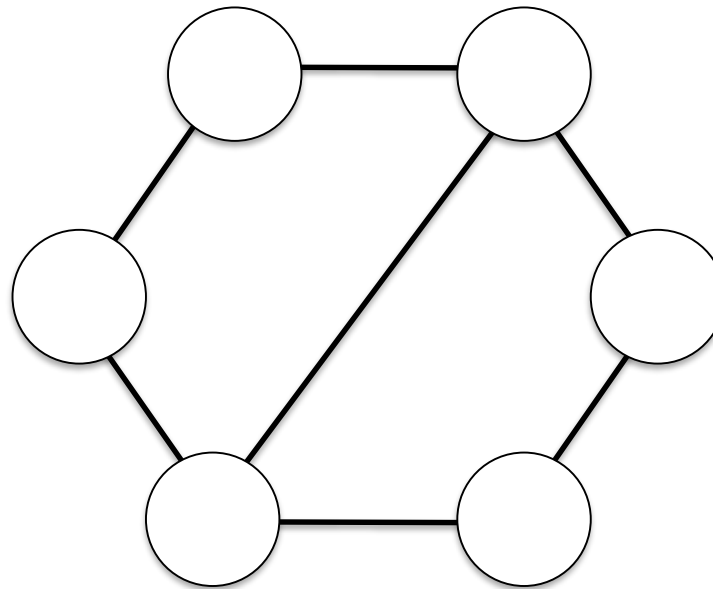
FIFO : Attention cela n'est valable que pour des messages envoyés par le même proc. et par le même lien :



Systeme distribue

Liens

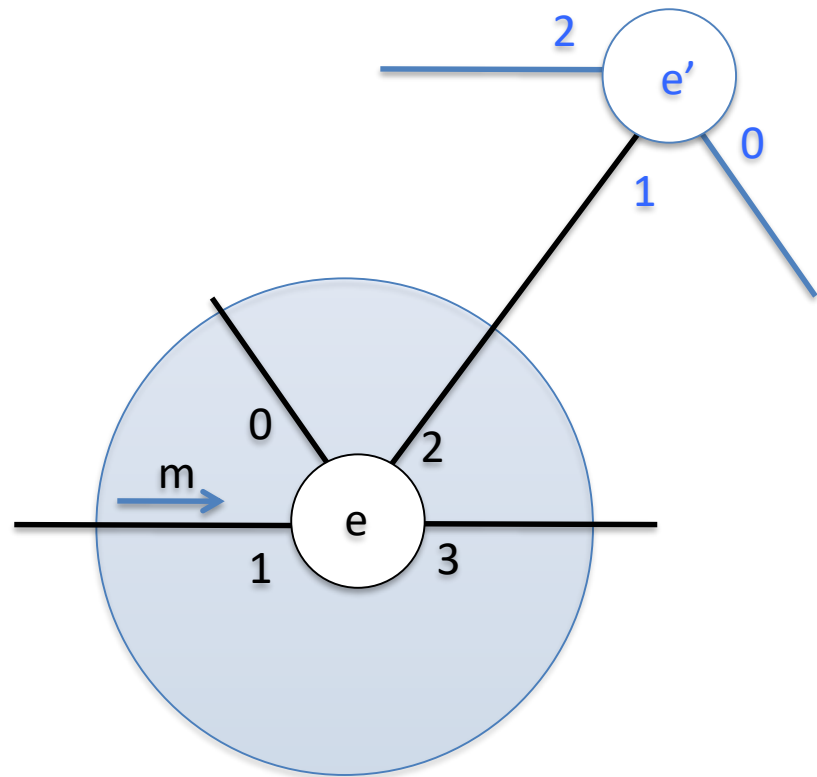
FIFO : Attention cela n'est valable que pour des messages envoyés par le même proc. et par le même lien :



Système distribué

Connaissance locale

- numérotation des liens
- état du proc.
- messages reçus



Systeme distribue

Hypotheses de base

- Systeme :
 - fini (nb de proc. et de liens fini)
 - connexe (tous les proc. sont relies directement ou indirectement)
 - memoire locale (communication par messages)
 - asynchrone (horloges locale, temps d'acheminement des messages fini mais non borne)
- Reseau identifie (tous les proc. ont un identificateur distinct)
- Liens :
 - bidirectionnels (messages dans les 2 sens)
 - FIFO (les messages ne se doublent pas dans un meme lien)
- Connaissance locale (numerotation des liens, etat du proc. et messages recus)

Algorithmique distribuée

Algorithmes

Texte :

- Un texte sur chaque proc. (algo local)
- Suite de blocs exécutés indépendamment

Trois nouvelles primitives :

- « Spontanément »
Entête du bloc réservé au démarrage de l'algo par le(s) initiateur(s)
(1 seul par algo local)
- « À la réception de M par C »
Entête d'un bloc à exécuter à la réception d'un message, M est le message reçu, C est le lien (canal) par lequel M est arrivé
- « Envoyer M à C »
Instruction simple consistant à envoyer le message M par le canal C , C peut aussi désigner un ensemble de canaux



Algorithmique distribuée

Algorithmes

Exécution :

- Un ou plusieurs initiateurs exécutent leur bloc « Spontanément »
- A la réception d'un message, un proc. exécute le bloc correspondant

Attention :

- Le nombre d'initiateurs doit être précisé
- Tout bloc est exécuté sans interruption
- Les réceptions de messages sont traitées de façon séquentielle
- Les messages arrivant par un même canal sont traités dans l'ordre d'arrivée (FIFO)
- Les messages arrivant de canaux différents sont traités dans n'importe quel ordre

Algorithmique distribuée

Algorithmes

Complément sur les hypothèses de base :

- Système :
 - fini (nb de proc. et de liens fini)
 - connexe (tous les proc. sont reliés directement ou indirectement)
 - mémoire locale (communication par messages)
 - asynchrone (horloges locale, temps d'acheminement des messages fini mais non borné)
- Réseau identifié (tous les proc. ont un identificateur distinct)
- Liens :
 - bidirectionnels (messages dans les 2 sens)
 - FIFO (les messages ne se doublent pas dans un même lien)
- Connaissance locale (numérotation des liens, état du proc. et messages reçus)
- Temps d'exécution fini (instructions simples)
- Initialisation par tous les initiateurs en temps fini (à développer plus tard)

Algorithmique distribuée

Algorithmes

Algorithmes déterministes ou probabilistes ?

Algo local déterministe :

quel que soit p un proc. quelconque, alors l'algorithme exécute toujours les mêmes actions en fonction de l'état de p , du message reçu par p et du canal de réception.

Algo local probabiliste :

il existe un proc. p , un état de p , un message reçu par p sur un canal pour lesquels les actions de p peuvent varier.

En général, l'exécution globale d'un algo distribué asynchrone est non déterministe :

- La cause principale : la vitesse imprévisible des messages (asynchronisme)
- Les conséquences : l'évolution des valeurs des variables des processeurs peuvent être différentes pour deux exécutions se déroulant sur un même réseau avec les mêmes conditions initiales.

Un algo distribué sera néanmoins appelé « déterministe » s'il est localement déterministe, « probabiliste » sinon.

Dans ce cours : algos déterministes en général.

Algorithmique distribuée

Algorithmes

Des algorithmes pour tous...

De façon générale nous étudierons dans ce cours des algos dont l'exécution fait intervenir, à un moment ou un autre, tous les processeurs du réseau.

Ce n'est pas toujours le cas :

communication entre deux processeurs distants
mais le calcul de la table de routage fait appel à tous les processeurs.

Algorithmique distribuée

Exemple : Circulation de jeton unique sur un anneau (Le Lann, 1977),
Texte

Attention : 1 initiateur unique

Message J

Var C : canal

- Spontanément
Créer le jeton J
Envoyer J à 0
- À la réception de J par C
Envoyer J à $(C+1) \bmod 2$

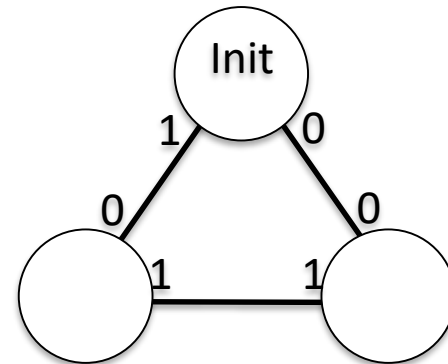
Algorithmique distribuée

Exemple : Circulation de jeton unique sur un anneau (Le Lann, 1977),
Exécution

Message J

Var C : canal

- Spontanément
Créer le jeton J
Envoyer J à 0
- À la réception de J par C
Envoyer J à $(C+1) \bmod 2$



Algorithmique distribuée

Exemple : Circulation de jeton unique sur un anneau (Le Lann, 1977),
Exécution

Message J

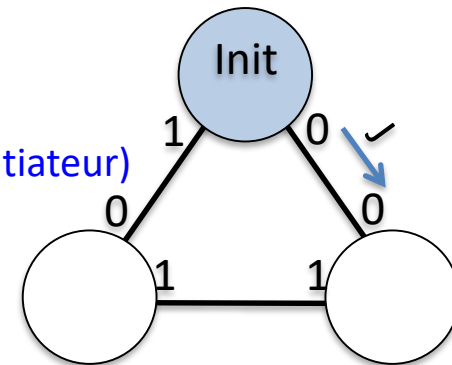
Var C : canal

- **Spontanément** (exécuté une seule fois par l'initiateur)

Créer le jeton J

Envoyer J à 0

- À la réception de J par C
Envoyer J à $(C+1) \bmod 2$



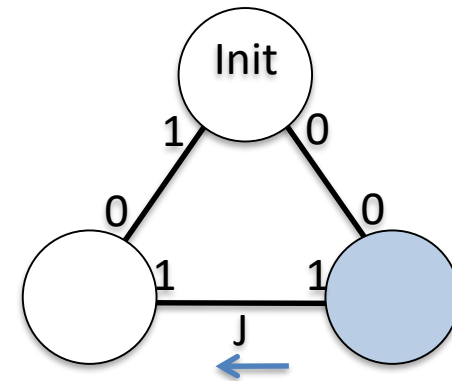
Algorithmique distribuée

Exemple : Circulation de jeton unique sur un anneau (Le Lann, 1977),
Exécution

Message J

Var C : canal

- Spontanément
Créer le jeton J
Envoyer J à 0
- À la réception de J par C
Envoyer J à $(C+1) \bmod 2$



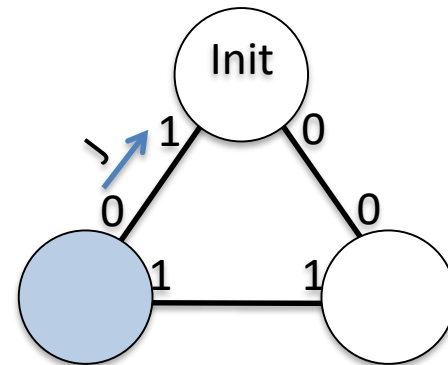
Algorithmique distribuée

Exemple : Circulation de jeton unique sur un anneau (Le Lann, 1977),
Exécution

Message J

Var C : canal

- Spontanément
Créer le jeton J
Envoyer J à 0
- À la réception de J par C
Envoyer J à $(C+1) \bmod 2$



Algorithmique distribuée

Exemple : Circulation de jeton unique sur un anneau (Le Lann, 1977),
Exécution

Message J

Var C : canal

- Spontanément
Créer le jeton J
Envoyer J à 0
- À la réception de J par C
Envoyer J à $(C+1) \bmod 2$

