



**HAL**  
open science

## Posters de médiation scientifique III: Réseaux de Tri

Nicolas Nisse

► **To cite this version:**

| Nicolas Nisse. Posters de médiation scientifique III: Réseaux de Tri. 2017. hal-01645164

**HAL Id: hal-01645164**

**<https://inria.hal.science/hal-01645164>**

Submitted on 23 Nov 2017

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Posters de médiation scientifique III : Réseaux de Tri

Nicolas Nisse<sup>\*,\*\*</sup>

Université Côte d'Azur, Inria, CNRS, I3S, France

**Abstract.** Ces posters font partie d'une série de posters que nous présentons lors de divers interventions de médiation (vulgarisation) scientifique (Fête de la Science, intervention dans des écoles, etc.). Nous essayons d'y présenter des bases théoriques (mathématiques) de l'algorithmique (Un algorithme est une suite finie et non ambiguë d'opérations ou d'instructions permettant de résoudre un problème ou d'obtenir un résultat) et structures de données (comment "coder" un nombre, une image, etc.)..

Ici, nous présentons une façon (un algorithme) de trier des éléments (par exemple, des nombres) grâce aux réseaux de tri. Pour les plus curieux d'entre vous, nous décrivons brièvement certains constituants élémentaires d'un ordinateur et comment ceux-ci permettent de faire des calculs (comment ils peuvent être assemblés pour trier).

Ces posters sont accessibles dès l'école primaire (exceptée, peut-être, la partie relative à l'architecture des ordinateurs).

---

\* nicolas.nisse@inria.fr

\*\* Merci à Frédéric Havet, Dorian Mazauric et les autres membres de l'équipe COATI pour leur aide et leurs conseils.

# RÉSEAUX de TRI

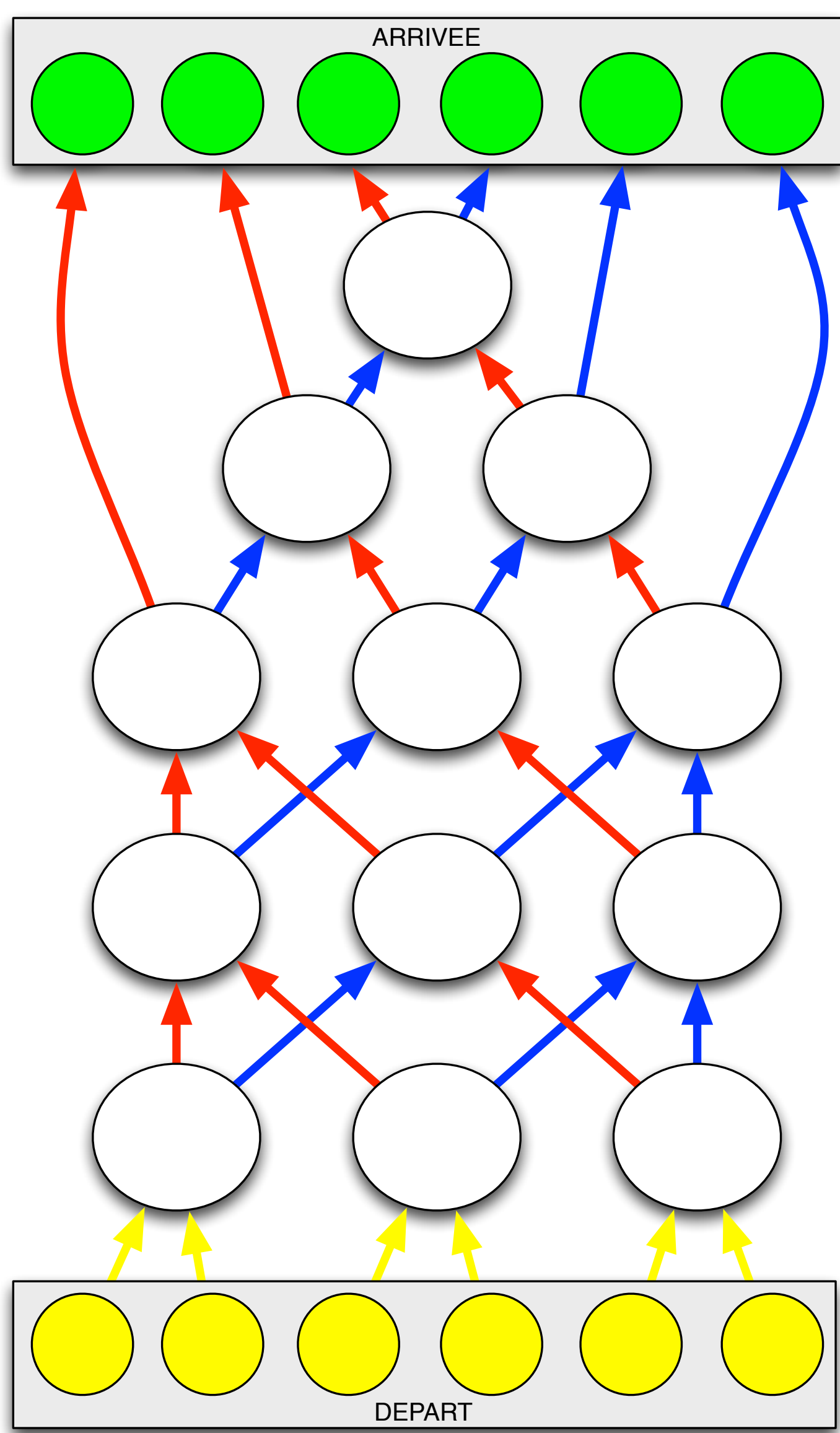


**Pourquoi (et quoi) trier ?  
Et comment le faire  
vite ?**

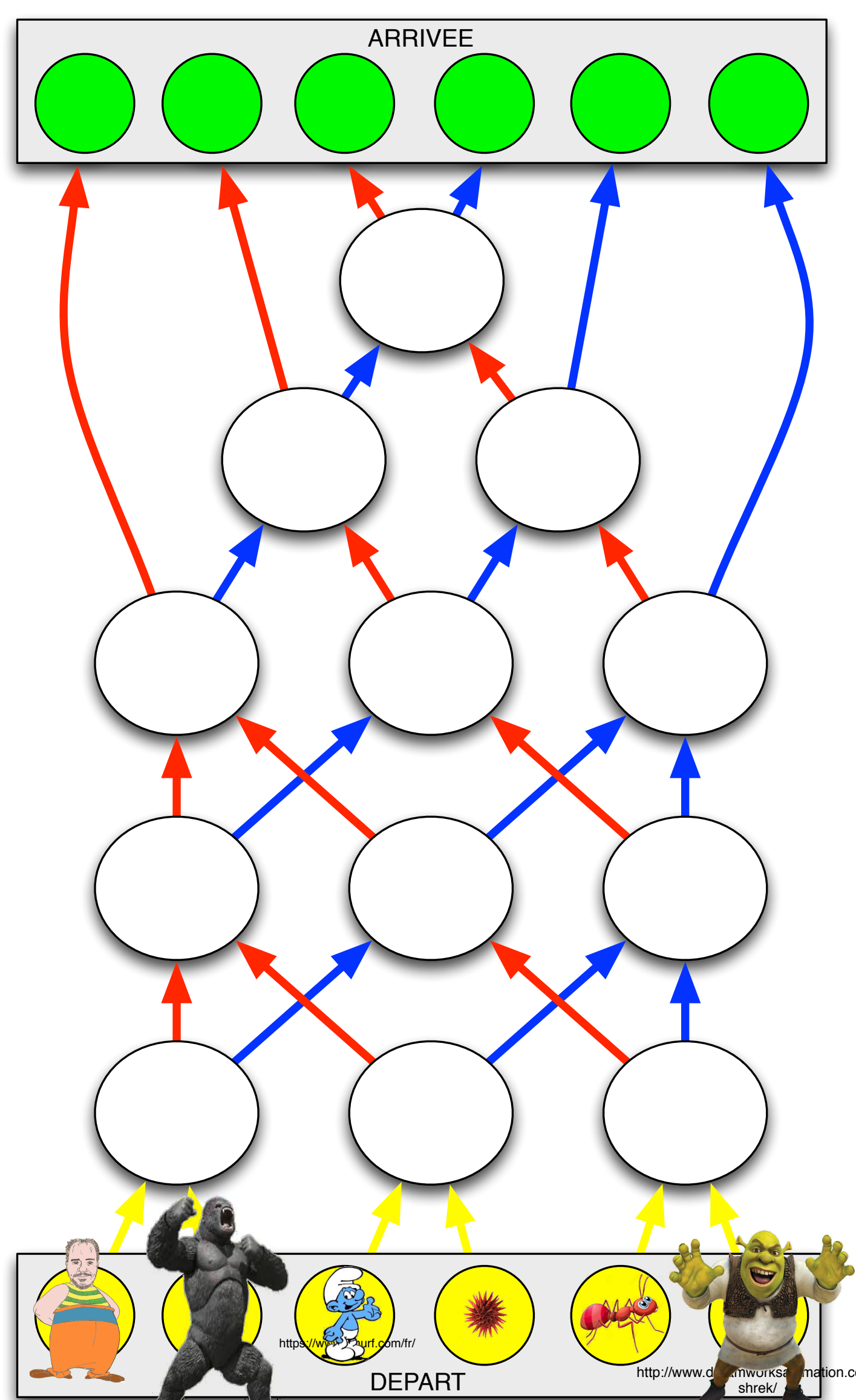
# RÉSEAUX de TRI

**Règle du jeu :** Six **joueurs** (de tailles différentes) se présentent à l'entrée (DEPART) du **réseau**. Ils progressent (de bas en haut) de cercle blanc en cercle blanc (qu'on appellera un **trieur**) en suivant la règle simple suivante : lorsque **2** joueurs sont dans un cercle, le **plus grand** suit le chemin **bleu** et le **plus petit** suit le chemin **rouge** (**interdit** d'avancer tant qu'on n'est pas deux dans un cercle).

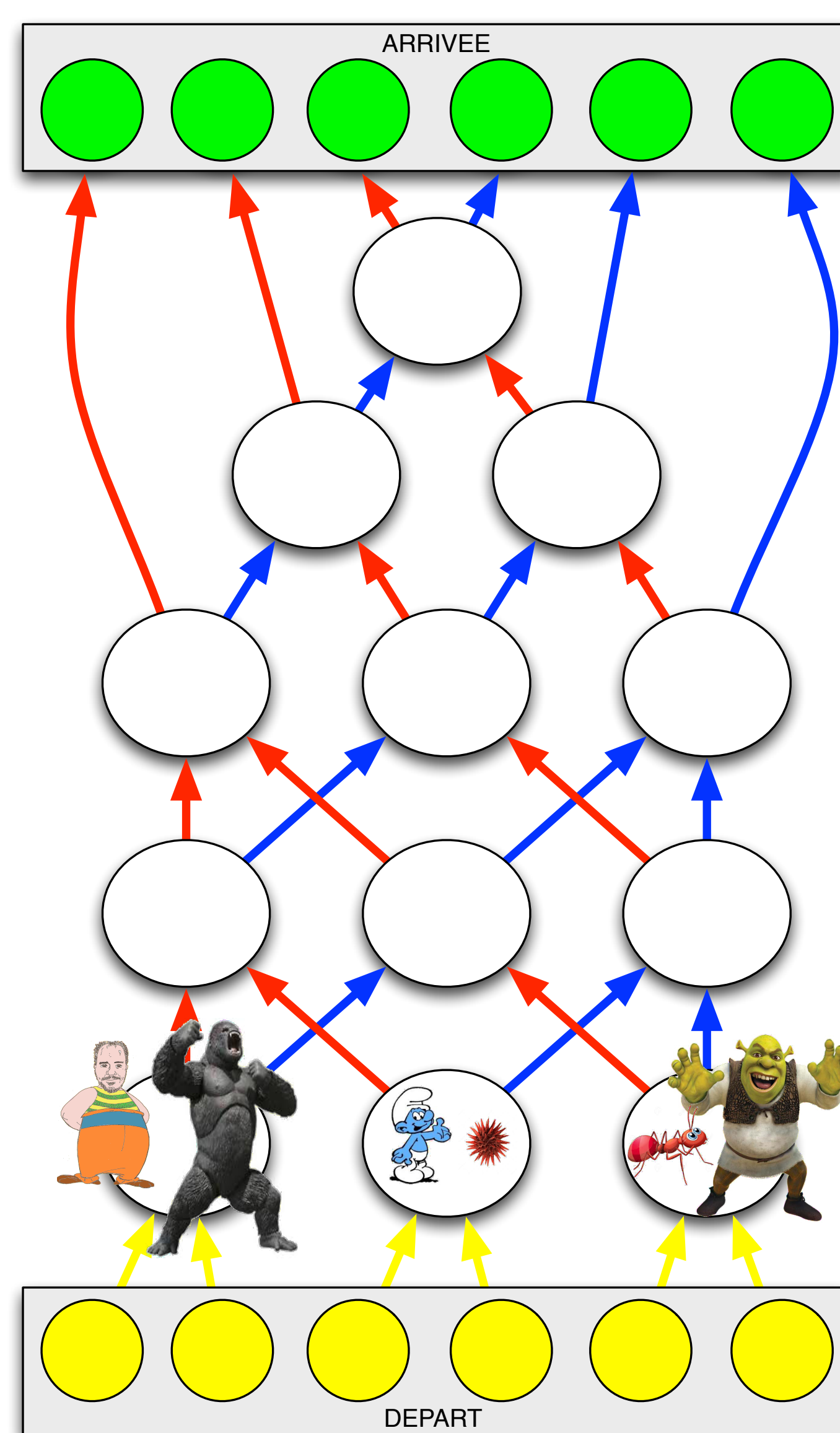
**Pour commencer**, chaque joueur fait un pas en avant (flèches jaunes) pour atteindre un premier cercle blanc.



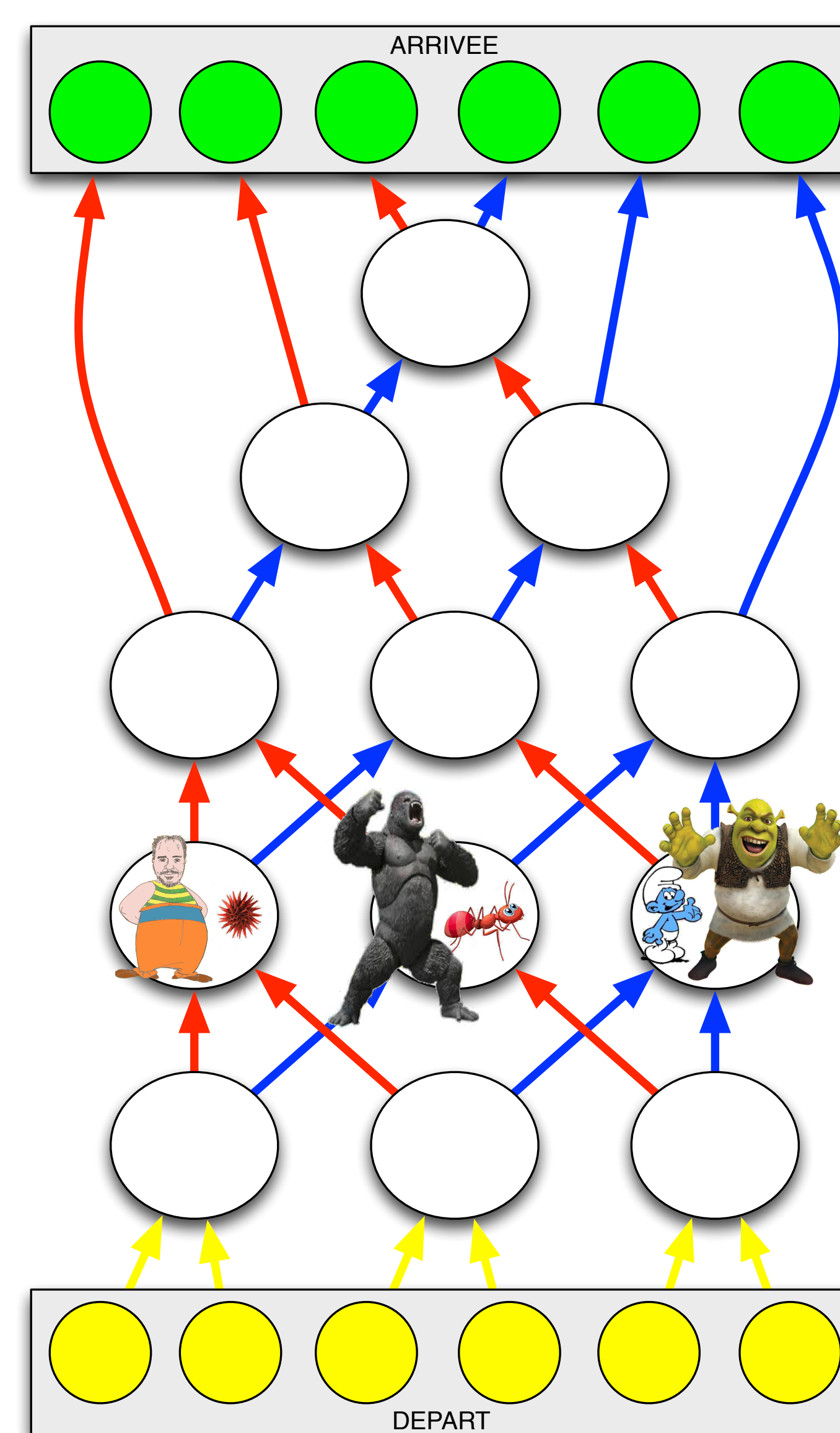
Réseau de Tri pour 6 éléments



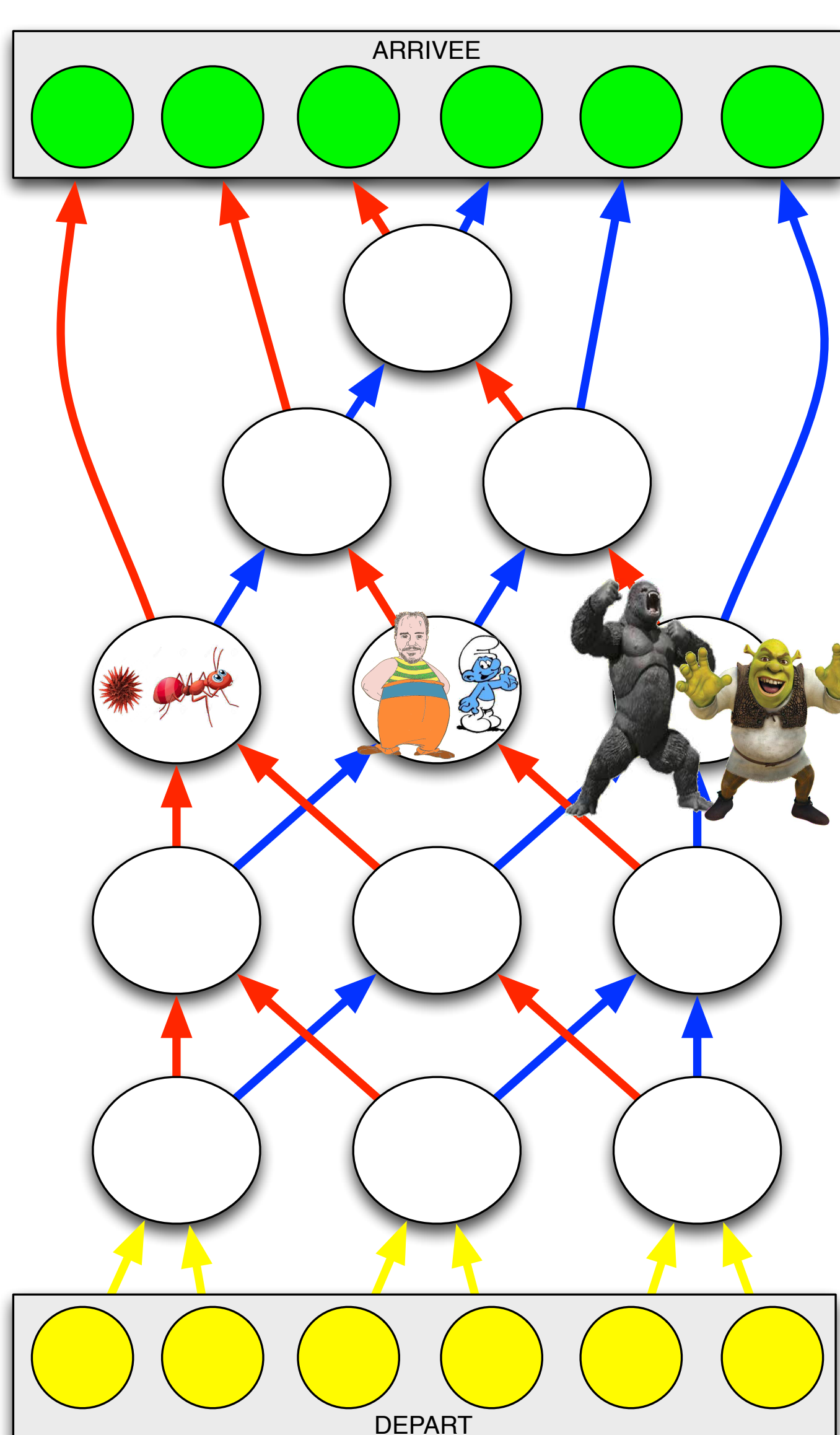
0) Les joueurs sont au départ



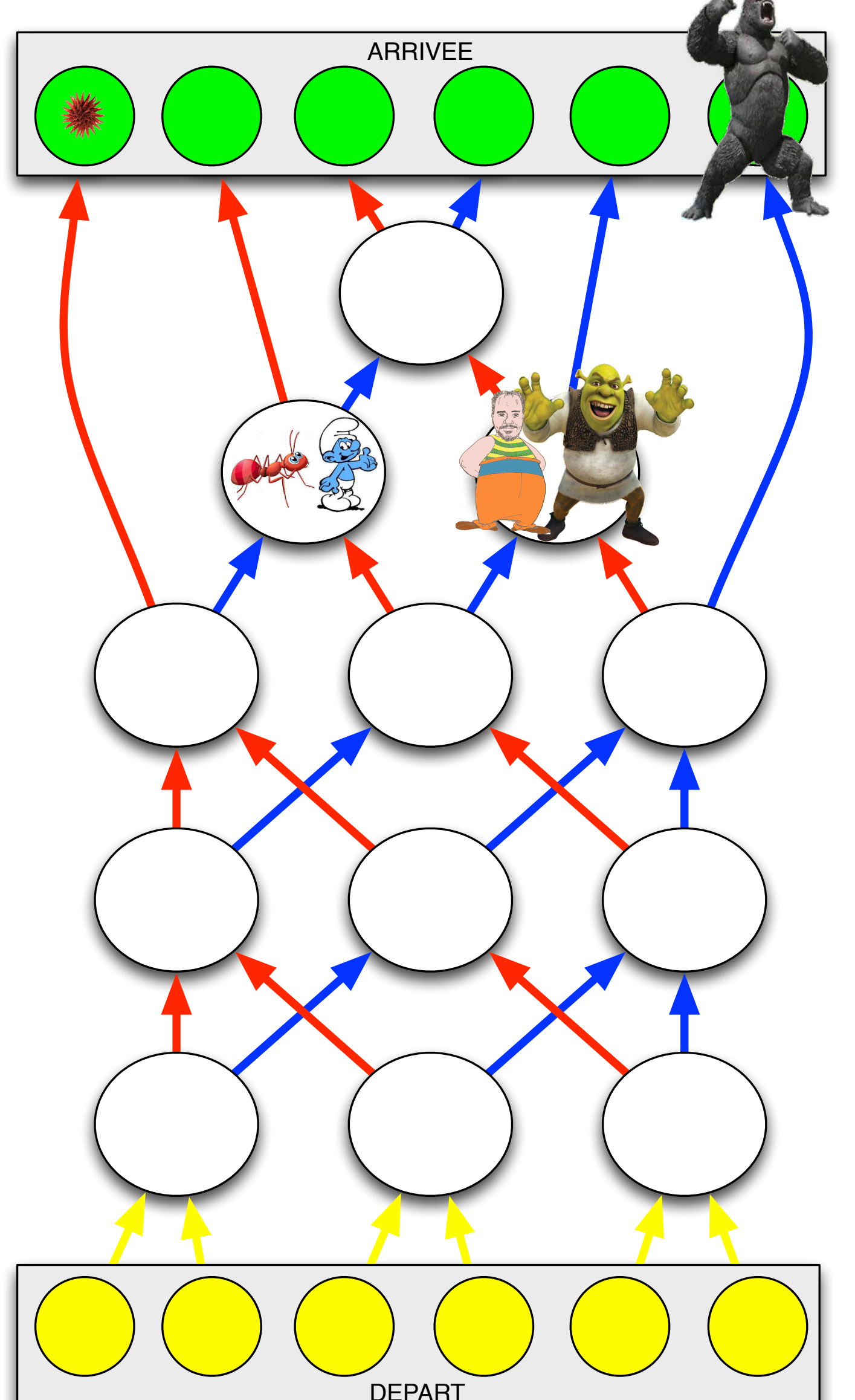
1) Premières comparaisons



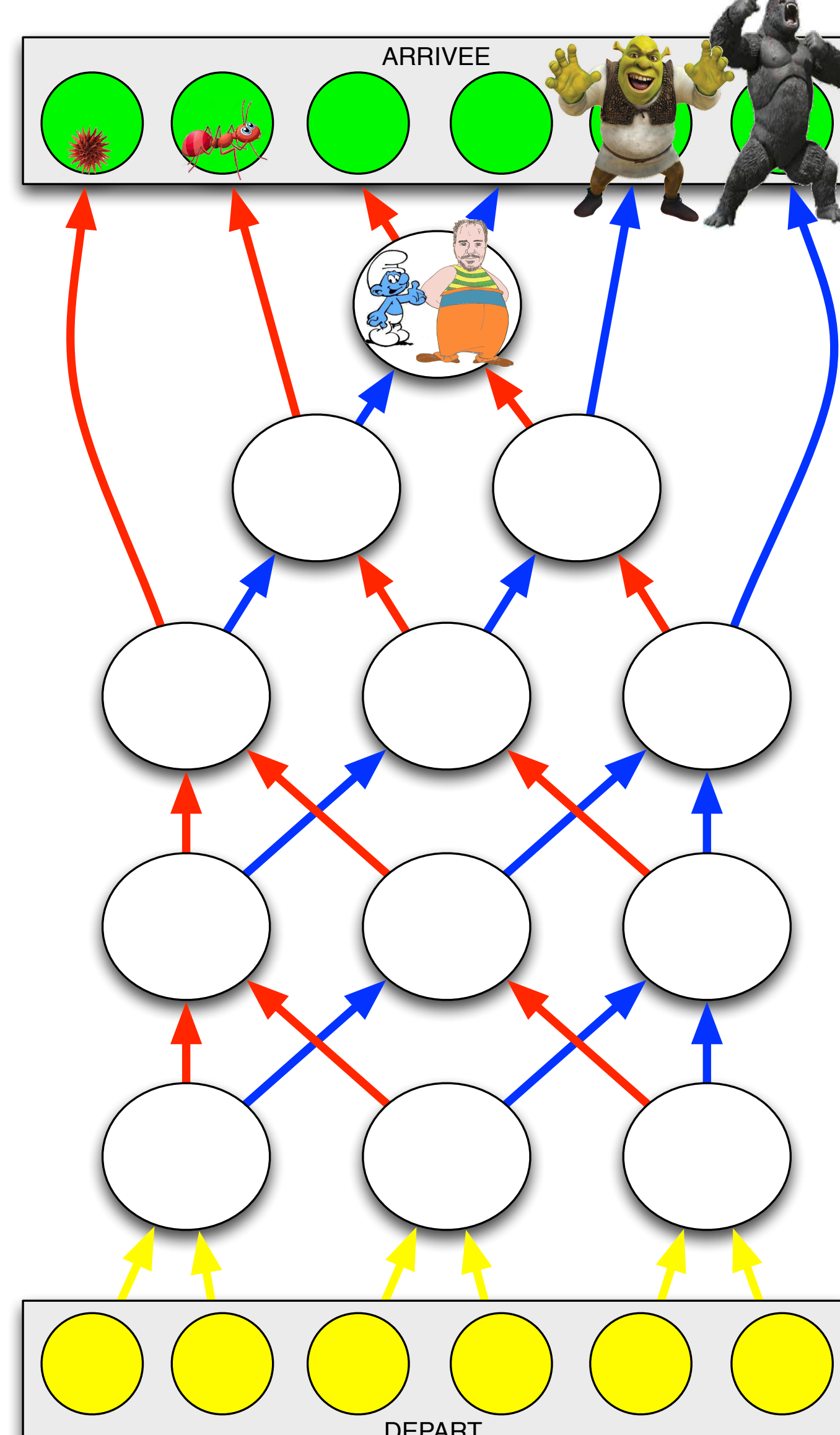
2) Deuxièmes comparaisons



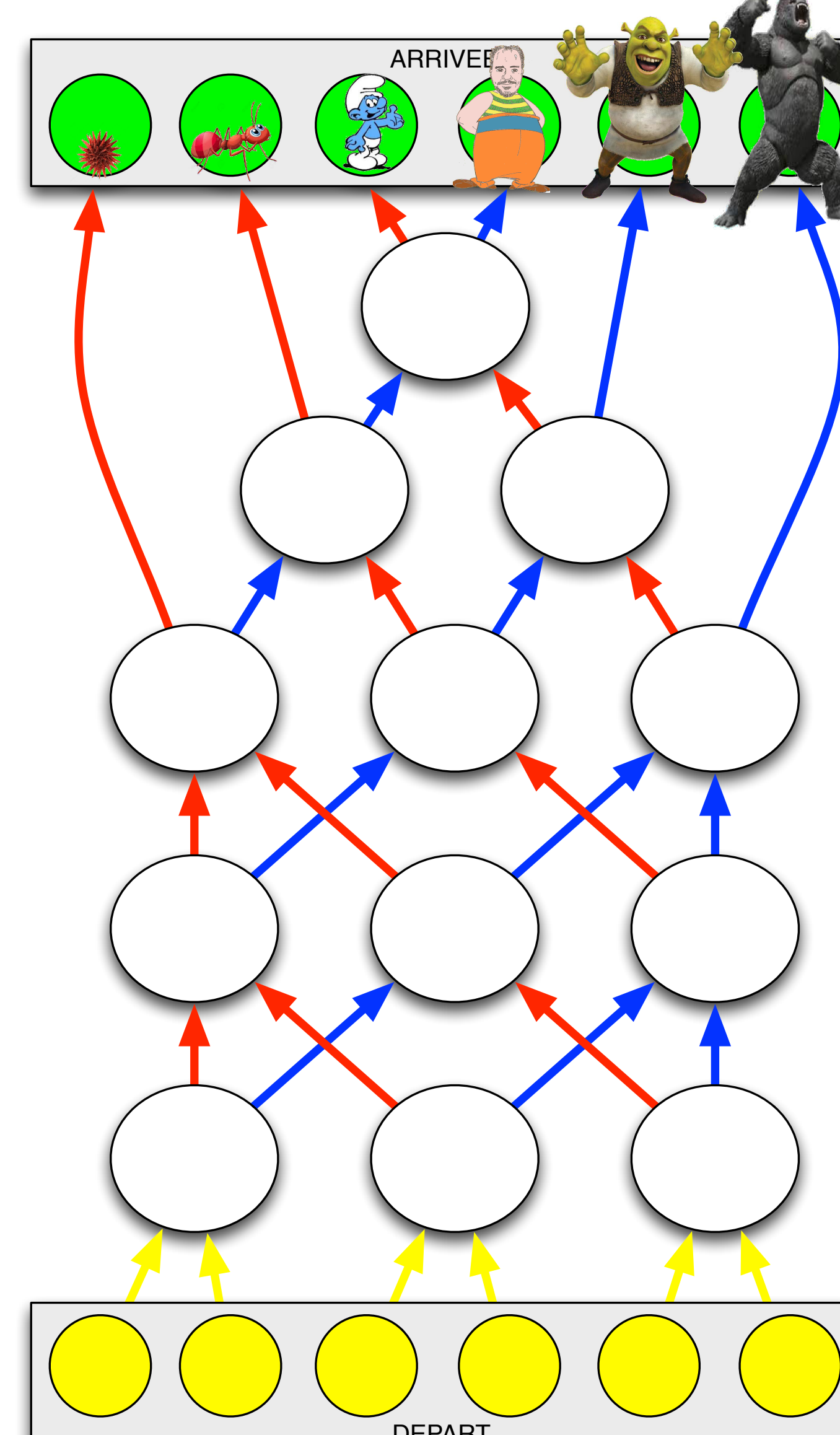
3) Encore des comparaisons...



4) Et encore...



5) Et encore...



6) Et finalement...

**Que s'est-il passé ?** Les joueurs sont **TRIÉS** par ordre **CROISSANT** de taille !! 😊  
(Du plus petit au plus grand)

Cela fonctionne-t-il quel que soit l'ordre initial ?

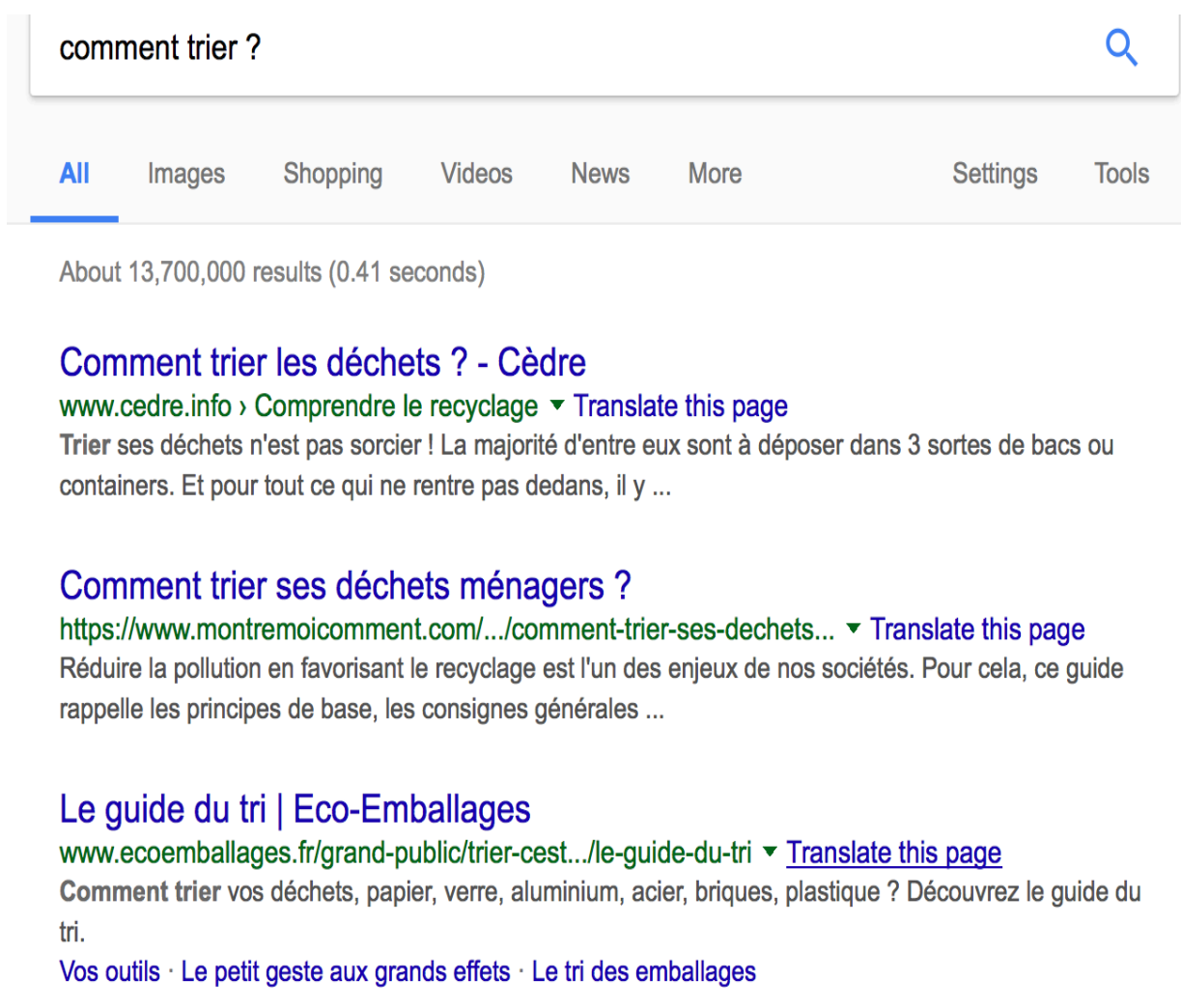
**OUI !!** (essayez)

Comment faire (simple modification des règles) pour trier dans l'ordre **DÉCROISSANT** ?  
(Du plus grand au plus petit)

# Pourquoi/Comment TRIER ?

**Quand ?** Vous triez (ou utilisez des choses triées) tout le temps !!

(si, si...)



|         |                            |                                       |             |           |                   |
|---------|----------------------------|---------------------------------------|-------------|-----------|-------------------|
| 842 €   | 14:30 - 17:35 <sup>1</sup> | Emirates, Malaysia                    | 21 h 05 min | 2 escales | DXB, KUL          |
| 843 €   | 14:30 - 21:50 <sup>1</sup> | Emirates                              | 25 h 20 min | 1 escale  | 13 h 00 min à DXB |
| 979 €   | 13:05 - 10:20 <sup>1</sup> | Aeroflot                              | 15 h 15 min | 1 escale  | 2 h 15 min à SVO  |
| 826 €   | 13:05 - 13:55 <sup>1</sup> | Aeroflot, Bangkok Airways             | 18 h 50 min | 2 escales | SVO, BKK          |
| 972 €   | 18:05 - 19:20 <sup>1</sup> | Air France, Silkair                   | 19 h 15 min | 2 escales | CDG, SIN          |
| 1 029 € | 12:20 - 17:35 <sup>1</sup> | Malaysia                              | 23 h 15 min | 2 escales | AMS, KUL          |
| 1 034 € | 14:30 - 16:00 <sup>1</sup> | Emirates, Bangkok Airways             | 19 h 30 min | 2 escales | DXB, BKK          |
| 1 046 € | 18:30 - 18:20 <sup>1</sup> | Etihad, Alitalia - Alitalia Cityliner | 17 h 50 min | 2 escales | FCO, AUH          |
| 1 564 € | 18:55 - 18:25 <sup>1</sup> | Lufthansa, TSN                        | 17 h 30 min | 2 escales | FRA, BKK          |



dictionnaire

moteurs de recherche

partir en voyage ?

mettre de l'ordre ?

jouer

**Pourquoi ? Permet de trouver PLUS RAPIDEMENT !!**

Un ordinateur passe également son temps à trier des choses (nombres, mots, images...).

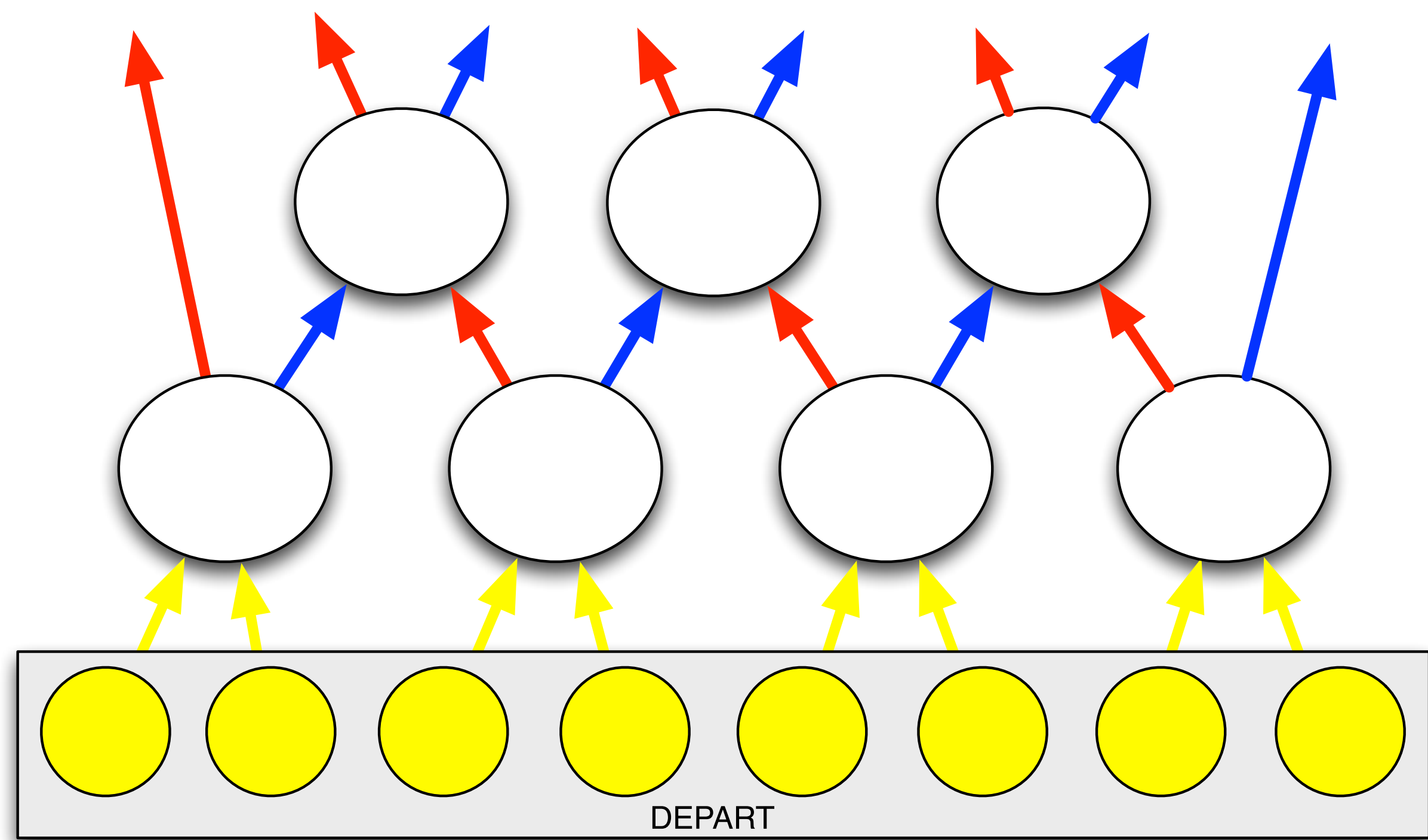
Pour cela, il existe de nombreux **algorithmes** (= méthode systématique "expliquable" à un ordinateur) qui comparent ces choses deux-à-deux.

Pour aller **plus vite**, il faut faire des comparaisons simultanées (**parallélisme**). C'est ce que permettent les **réseaux de tri** !! (utilisés, par exemple, dans certaines **cartes graphiques** des ordinateurs)

**Comment ?** Les réseaux de tri permettent de trier autant de nombres que l'on veut !! Mais ça demande de la place (**nombre** de trieurs) et du temps (« **hauteur** » du réseau) ...

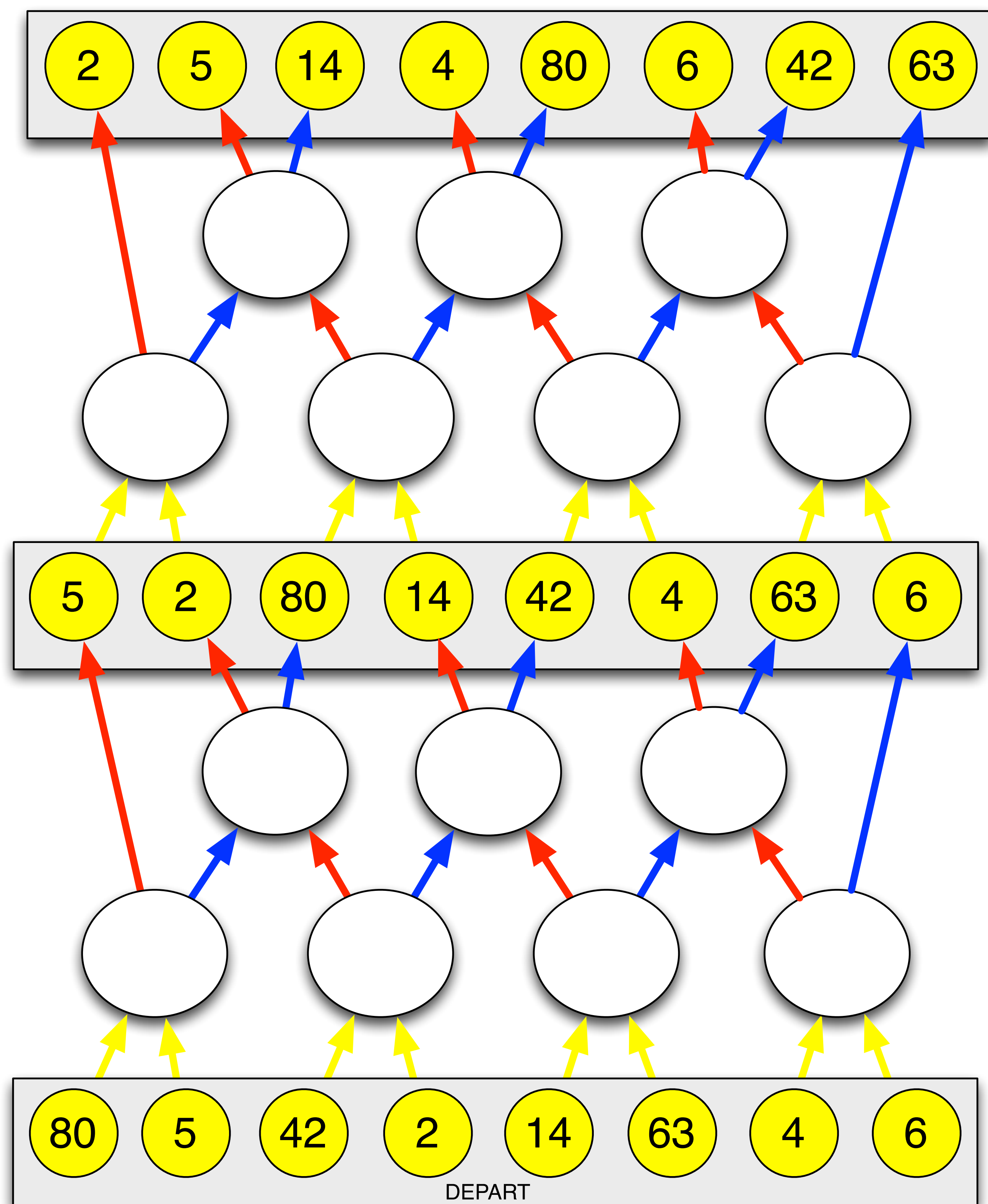
## Essayons de trier 8 nombres.

**Combien de fois** faut-il parcourir (répéter) le « morceau » de réseau ci-dessous pour les trier ? (Essayez !!)



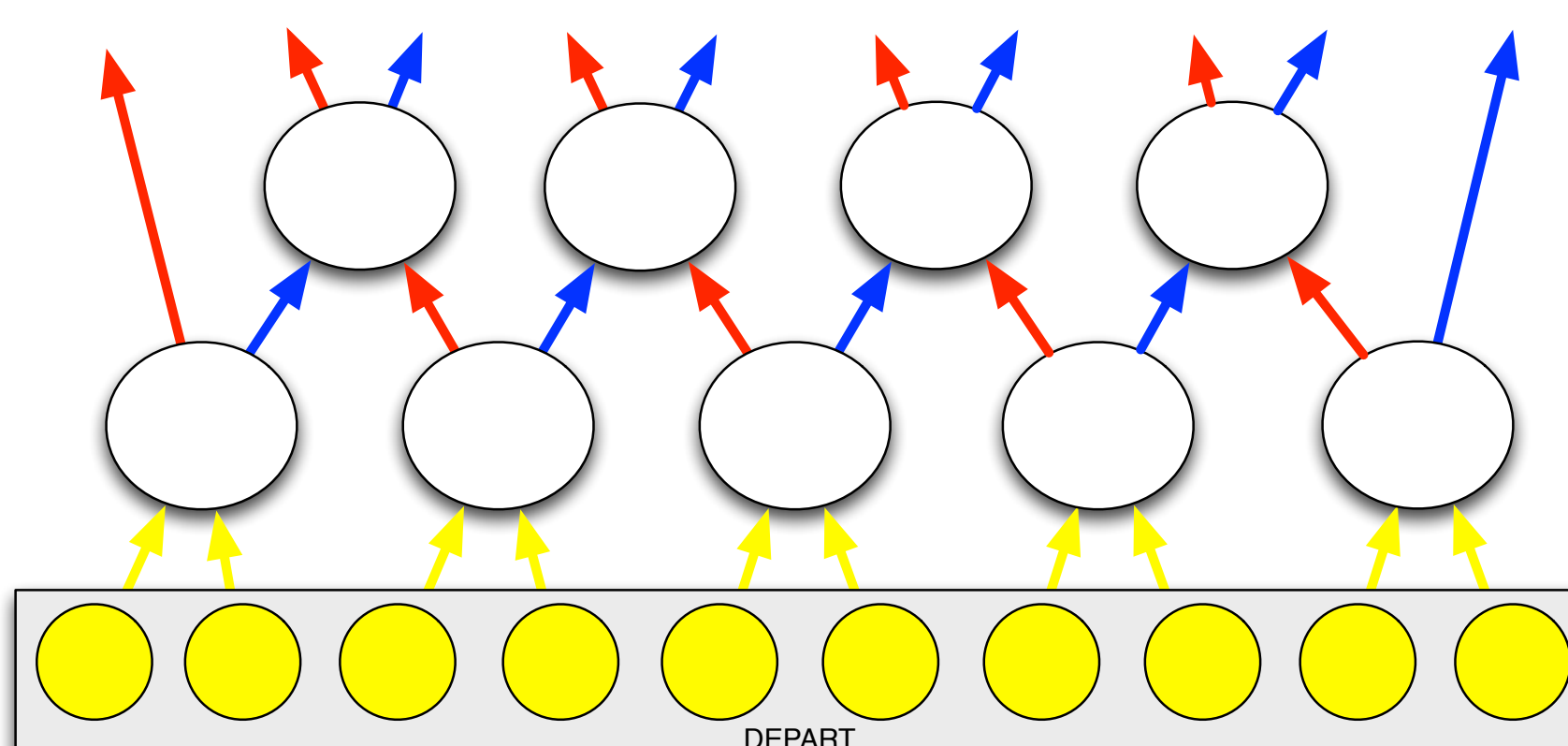
Voyons (illustration à droite) ce qui se passe si on le répète 2 fois.

Réponse : il faut le répéter 4 fois, soit 28 trieurs !!



Adapter le réseau ci-dessus permet (en le répétant) de trier autant de nombres que l'on veut.

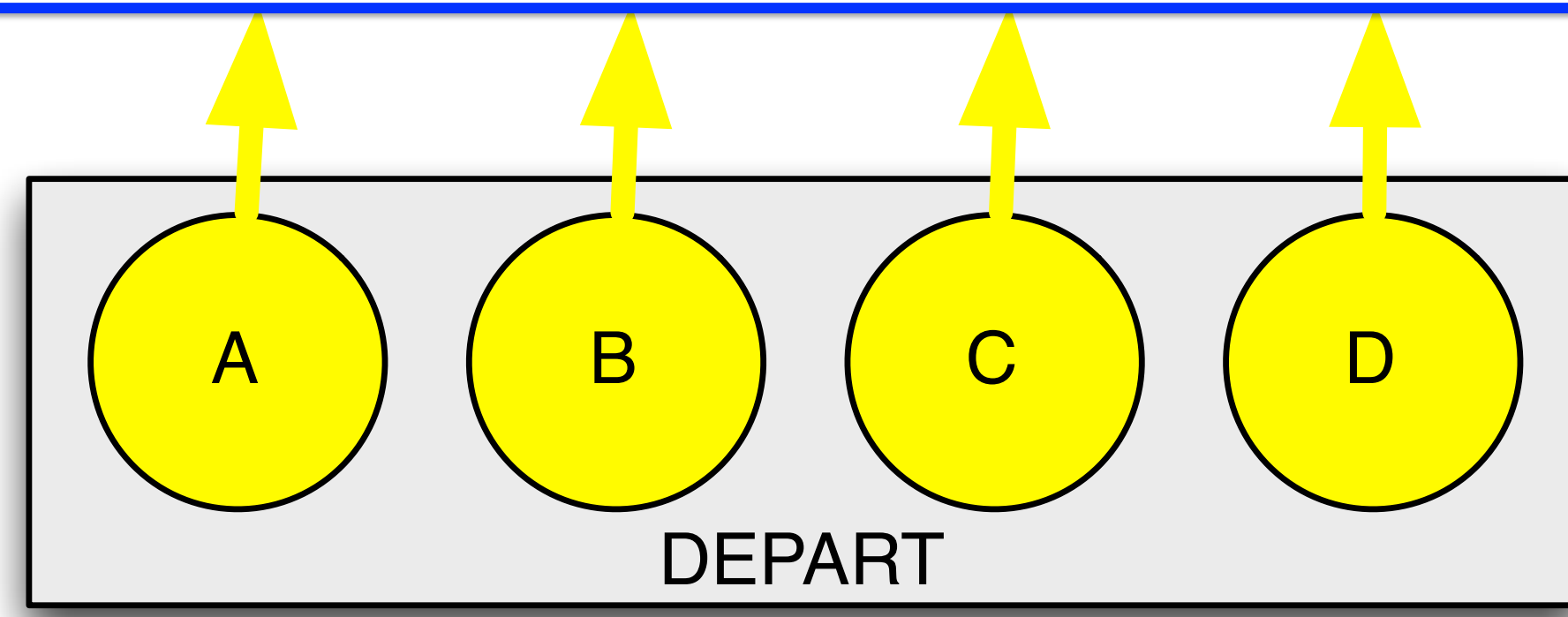
Par exemple, répéter cinq fois (45 trieurs) le réseau de droite permet de trier 10 nombres.



**Cependant, on peut faire mieux !!!**  
c'est-à-dire utiliser **MOINS de trieurs, MOINS d'étapes !!**

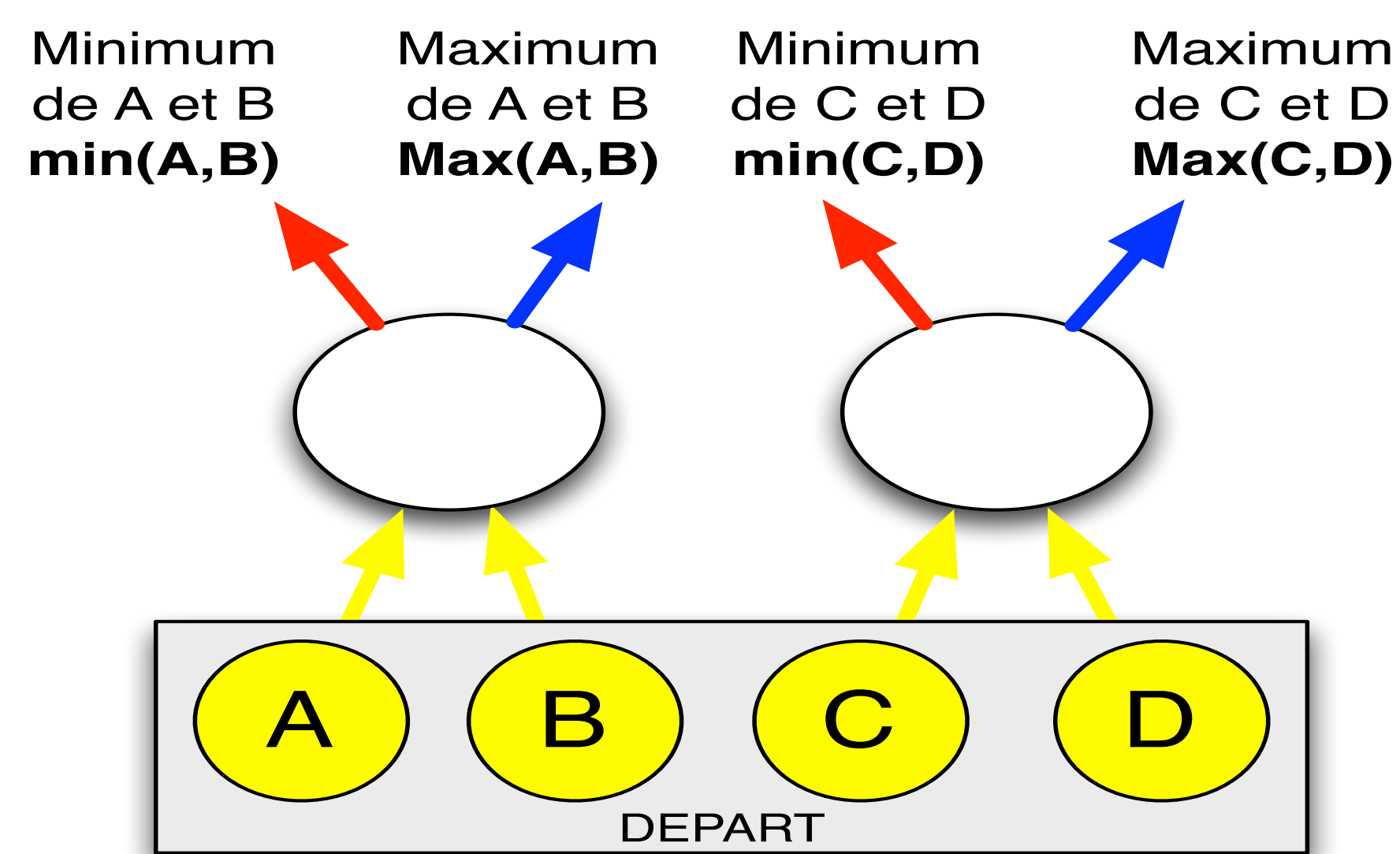
# Construire un «PETIT» réseau de tri

1 Essayons de trier 4 éléments !



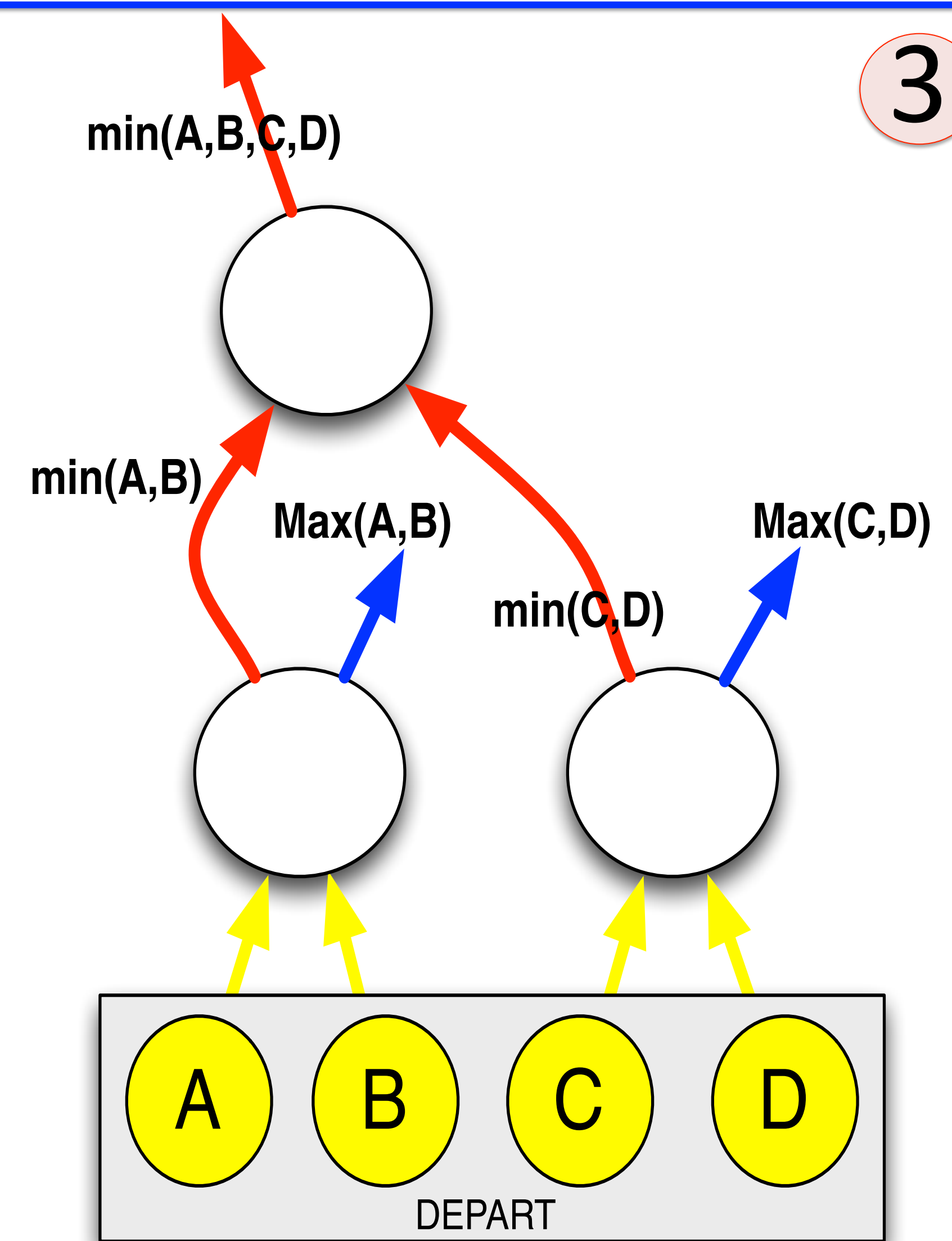
2

Intuitivement, on compare d'une part A et B, et d'autre part C et D.

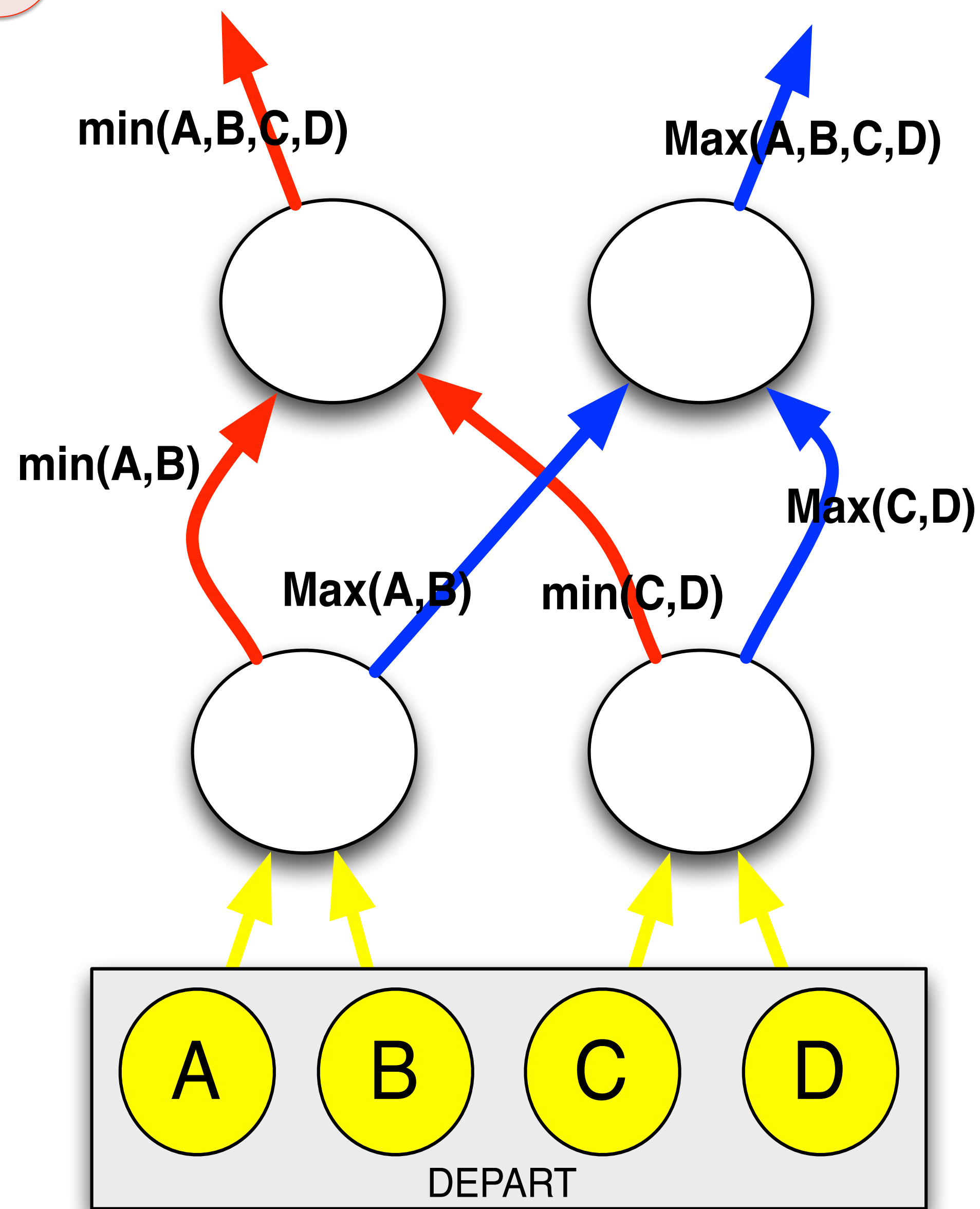


L'expérience montre que l'on est tenté de comparer la flèche bleue de gauche ( $\text{Max}(A,B)$ ) et celle rouge de droite ( $\text{Min}(C,D)$ ). Pourtant, on peut faire **mieux** (construire un réseau de tri **avec moins de trieurs**)

Comparer les deux plus petits (flèches rouges) permet de déterminer « le plus petit des plus petits », donc le plus petit de tous.

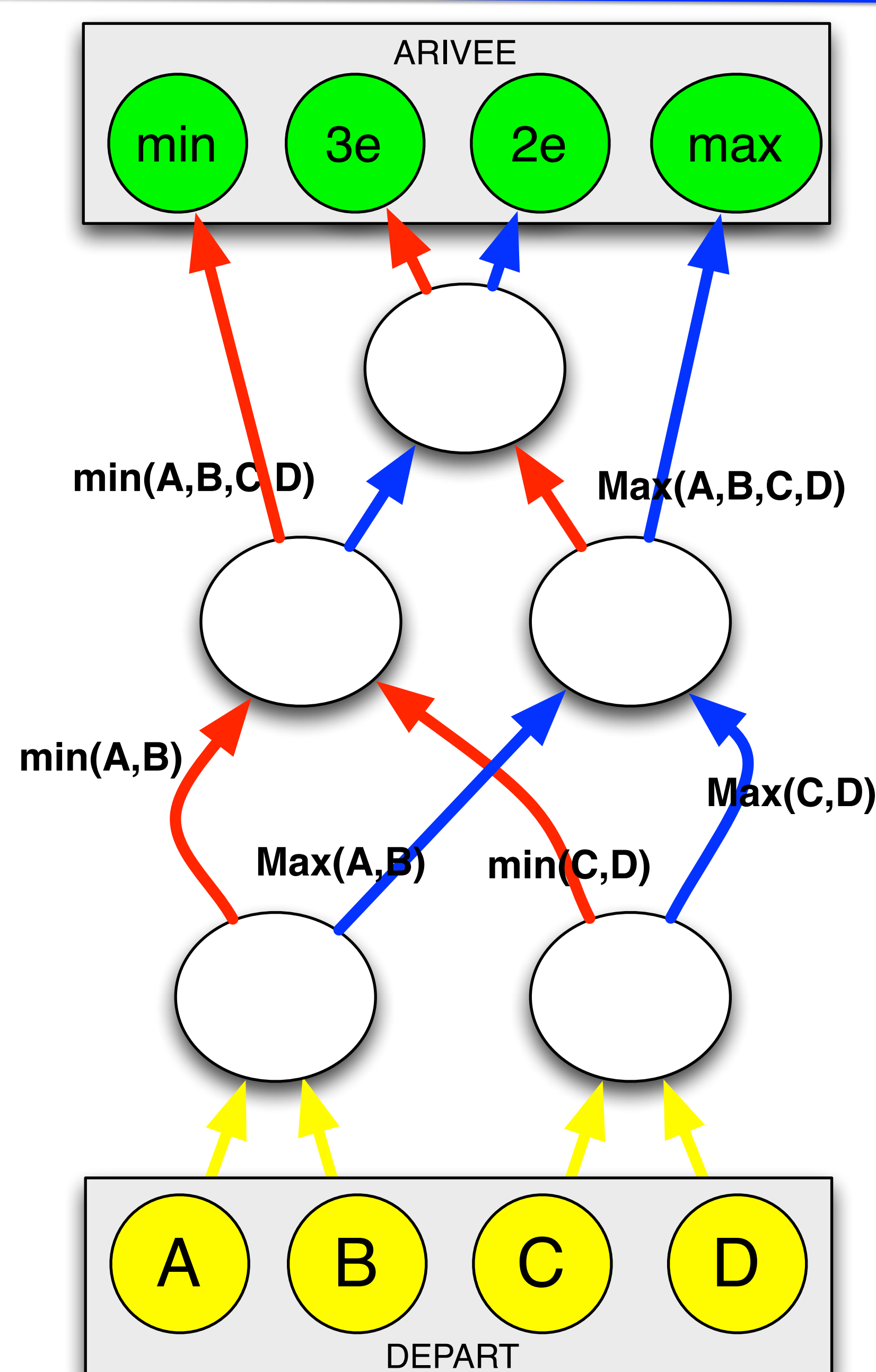


4



De même, comparer les deux plus grands (flèches bleues) permet de déterminer « le plus grand des plus grands », donc le plus grand de tous.

5 Il ne reste qu'à comparer les 2 derniers éléments.

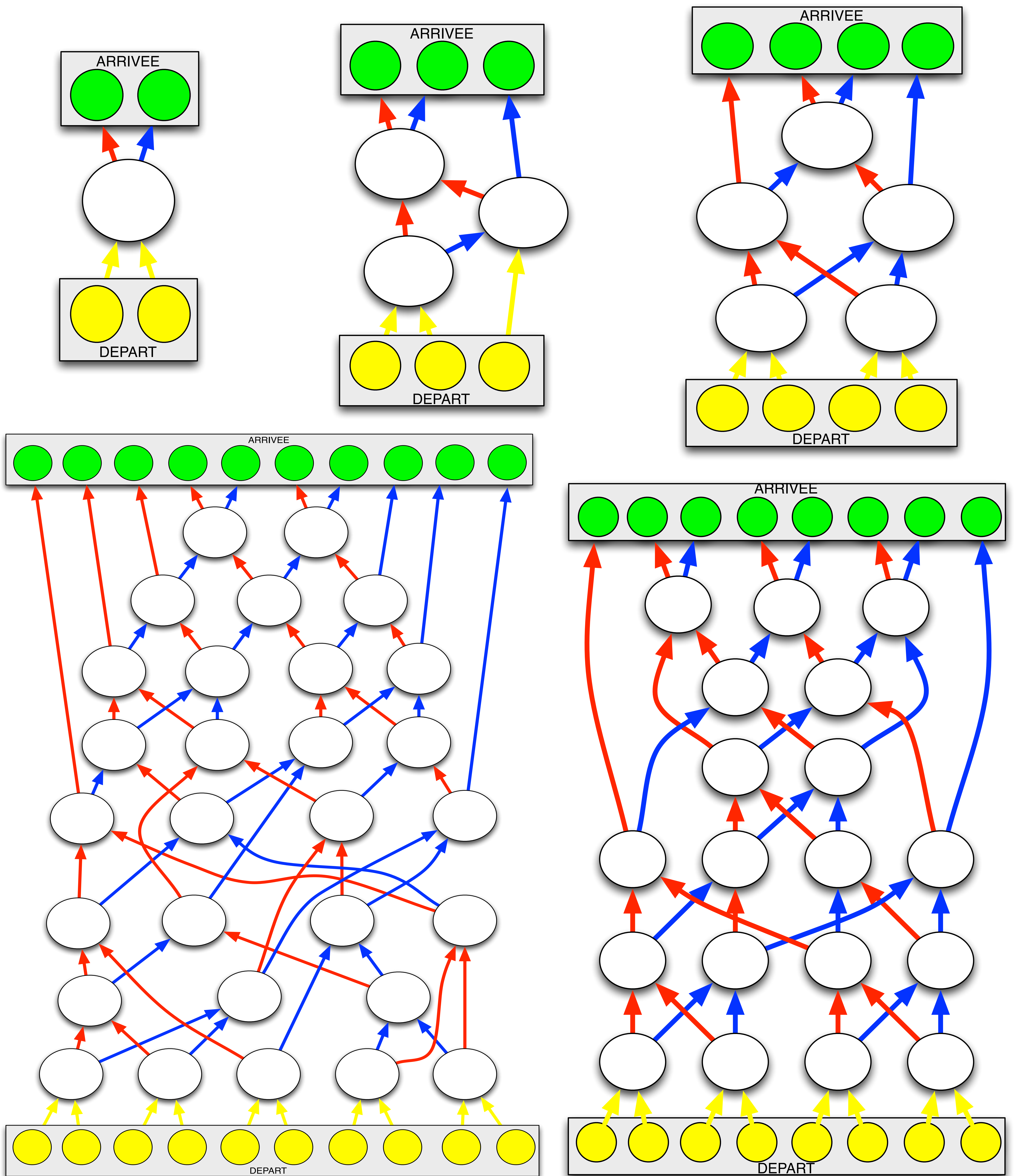


Comparer 4 éléments nécessite au plus 5 trieurs.

(essayez de faire mieux, i.e., d'utiliser moins de trieurs...)

# Quelques réseaux de tri optimaux

Pour trier jusqu'à 16 nombres, on connaît des réseaux de tri **optimaux** (utilisant le moins de trieurs)



Pour trier **8** éléments, le réseau ci-dessus à droite (avec **19** trieurs) est **optimal** (on ne peut pas faire moins).  
Pour trier **10** éléments, le réseau ci-dessus à gauche (avec **29** trieurs) est **optimal**.

Pour trier **16** éléments, il faut **60** trieurs ! (on ne sait que c'est **optimal** que depuis 2017 !!)

**Pour plus que 16 éléments, on ne connaît pas le plus petit réseau de tri**

(pour trier 17 éléments il faut au moins 60 trieurs et au plus 71)

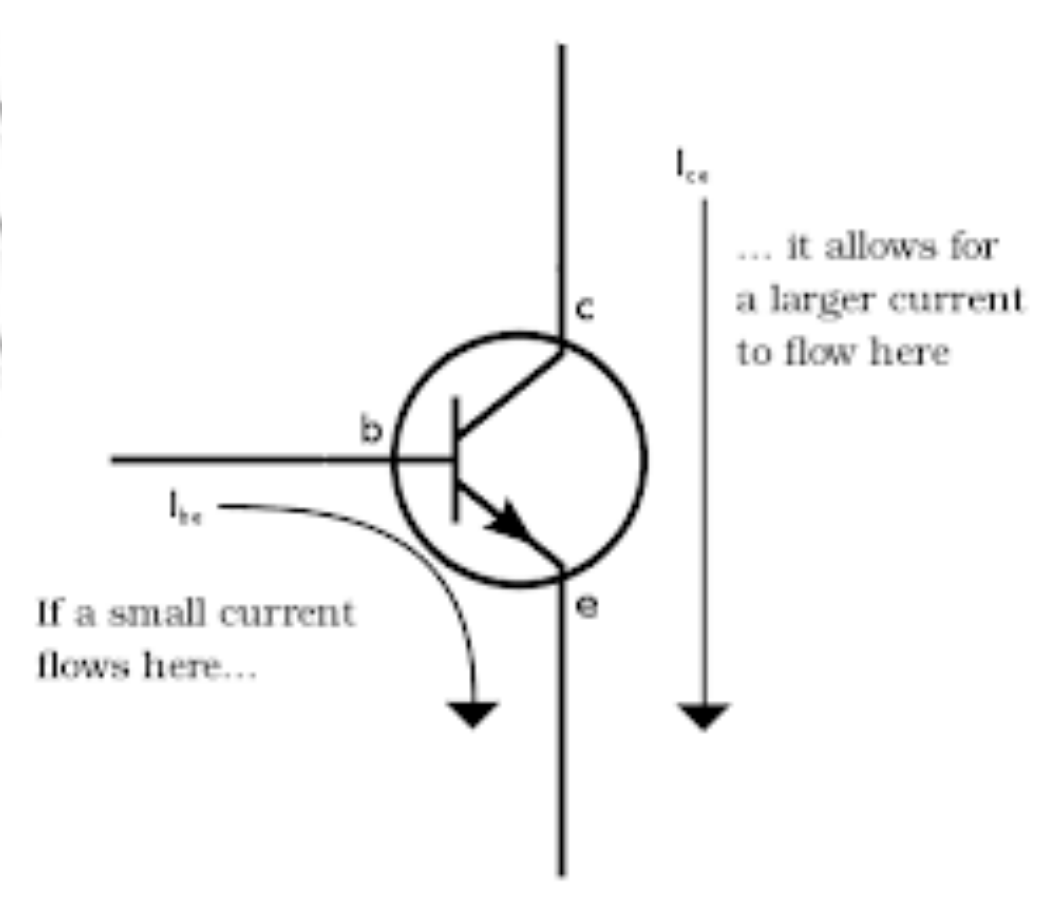
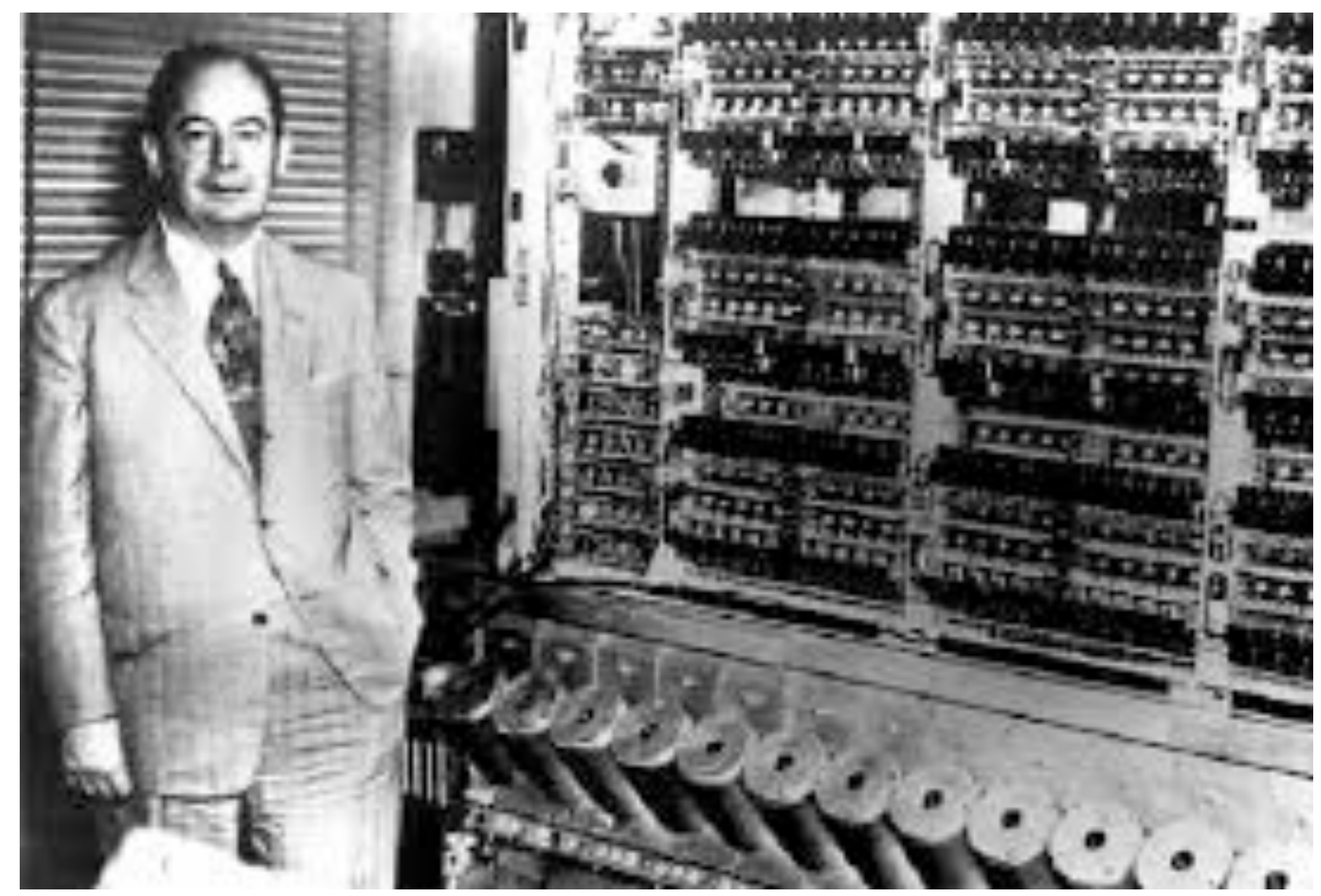
[D.E. Knuth. The Art of Computer Programming, Vol. III: Sorting and Searching. Addison-Wesley, 1973.]

[D.Bundala, M.Codish, L. Cruz-Filipe, P.Schneider-Kamp, J. Zavodny. Optimal-depth Sorting Networks, <https://arxiv.org/abs/1412.5302>, 2017]

# Et un ordinateur, (1/2)

## Il compare comment « en vrai » ?

**Rappel :** Un ordinateur ne comprend que 2 symboles (**binaire**) : **1** (signal) et **0** (pas de signal).  
Un ordinateur est une combinaison **d'interrupteurs** (signal/pas de signal)



John VON NEUMANN (1903-1957) invente l'ordinateur. L'invention du **TRANSISTOR** (= interrupteur « commandé ») par J.BARDEEN, W.BRATTAIN, W.SHOCKLEY (**Nobel** de physique 1956) est une REVOLUTION !!

**Opérations binaires :** Combiner des signaux binaires **a** et **b** (qui valent 0 ou 1)

### ET

**Table de vérité ET (AND)**

| Entrée 1 | Entrée 2 | Sortie |
|----------|----------|--------|
| a        | b        | s      |
| 0        | 0        | 0      |
| 0        | 1        | 0      |
| 1        | 0        | 0      |
| 1        | 1        | 1      |

La sortie vaut 1 si (et seulement si) les 2 entrées (a ET b) valent 1

### OU

**Table de vérité OU (OR)**

| Entrée 1 | Entrée 2 | Sortie |
|----------|----------|--------|
| a        | b        | s      |
| 0        | 0        | 0      |
| 0        | 1        | 1      |
| 1        | 0        | 1      |
| 1        | 1        | 1      |

La sortie vaut 1 si (et seulement si) au moins une des 2 entrées (a OU b) vaut 1

### NON

**Table de vérité NON (NOT)**

| Entrée 1 | Sortie |
|----------|--------|
| a        | s      |
| 0        | 1      |
| 1        | 0      |

La sortie vaut 1 si (et seulement si) l'entrée vaut 0

**Avec ces 3 portes (en fait 2 suffisent), on peut tout faire !!**

Par exemple, étant donnés deux signaux **a** et **b** binaires (0 ou 1) on peut :

Déterminer lequel est le plus grand, ou s'il y a égalité : (a>b ? a<b ? a=b ?)

Implicitement, on interprète le signal de sortie : **0 = FAUX**  
**1 = VRAI**

### Comparateur (binaire)

| Entrée 1 | Entrée 2 | Sortie 1 | Sortie 2 | Sortie 3 |
|----------|----------|----------|----------|----------|
| a        | b        | a < b    | a = b    | a > b    |
| 0        | 0        | 0        | 1        | 0        |
| 0        | 1        | 1        | 0        | 0        |
| 1        | 0        | 0        | 0        | 1        |
| 1        | 1        | 0        | 1        | 0        |

Déterminer le minimum et le maximum : (quelle est la « taille » du plus grand/petit ?)

| Entrée 1 | Entrée 2 | Sortie 1 | Sortie 2 |
|----------|----------|----------|----------|
| a        | b        | min      | Max      |
| 0        | 0        | 0        | 0        |
| 0        | 1        | 0        | 1        |
| 1        | 0        | 0        | 1        |
| 1        | 1        | 1        | 1        |



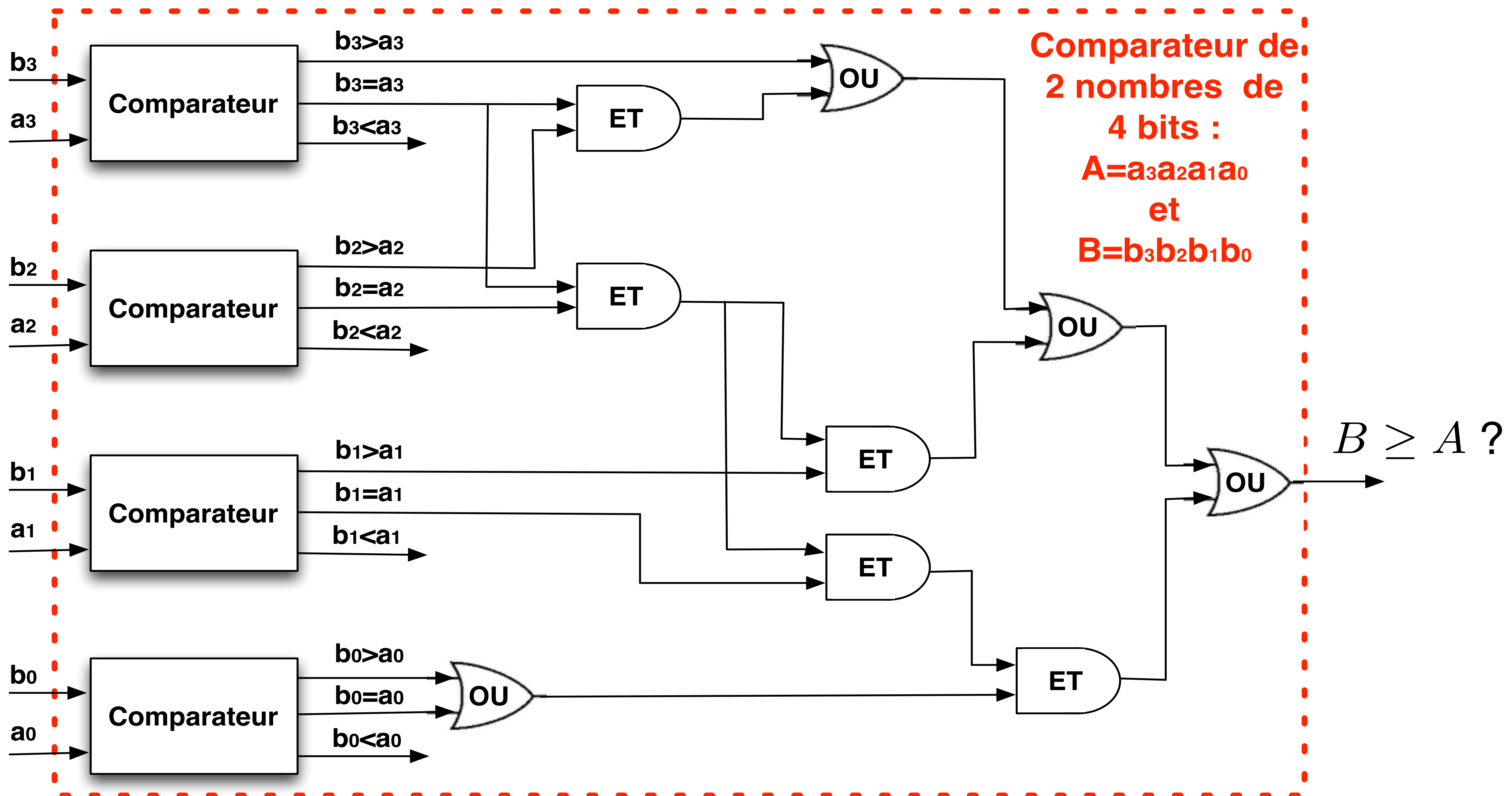
## Il compare comment « en vrai » ?

Comparer 2 nombres de plusieurs bits est à peine plus compliqué.

«Habituellement», pour comparer 2 nombres (décimaux), par exemple 20123 et 20115, on compare d'abord les dizaines de milliers (2 et 2), puis les milliers (0 et 0), puis les centaines (1 et 1), puis les dizaines (ici 2 et 1). On en déduit que le premier nombre est plus grand que le second.

Pour comparer 2 nombres **binaires**, c'est la «même chose» !!

Pour 2 nombres binaires, on compare les bits 2 à 2, en partant des bits de poids fort :

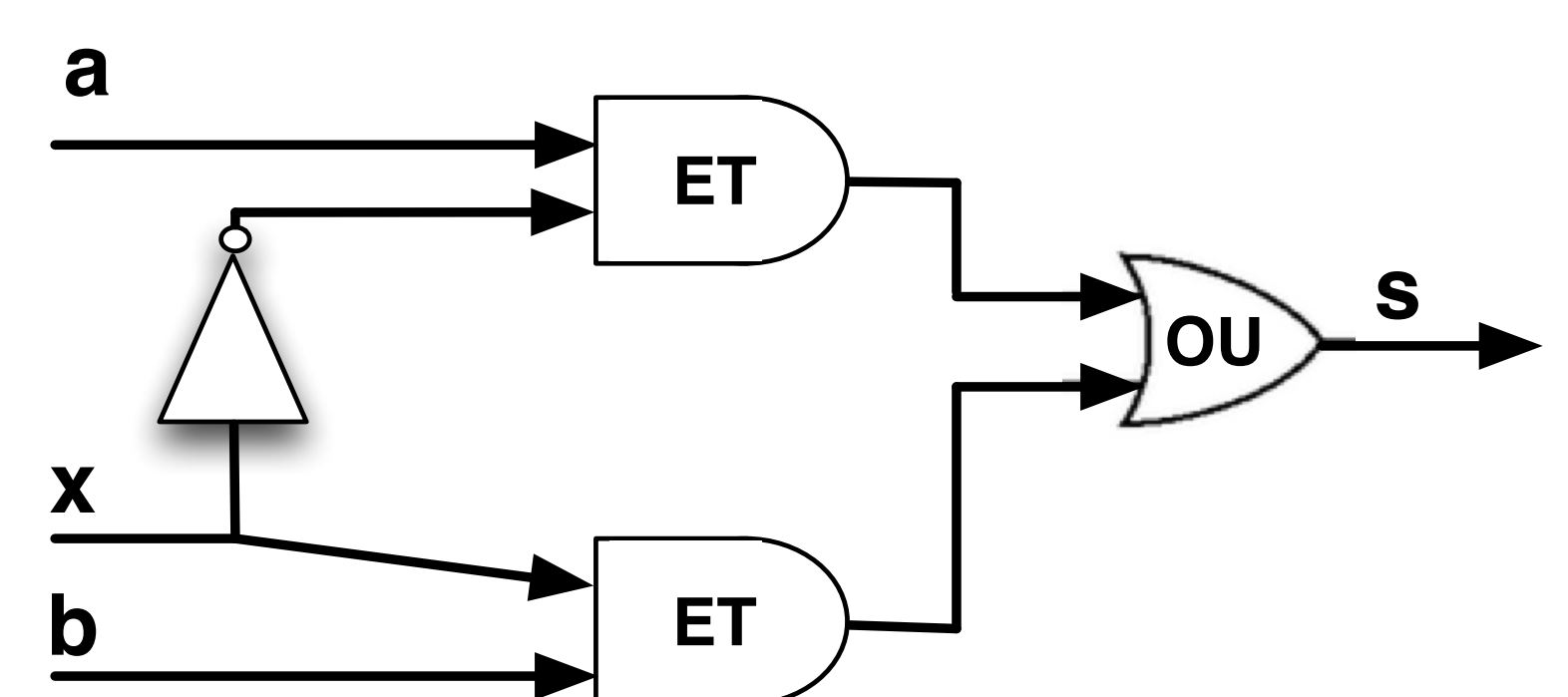


À droite, un **multiplexeur** (binaire) pouvant choisir (en fonction du signal binaire  $x$ ) entre 2 signaux (binaires)  $a$  et  $b$ .

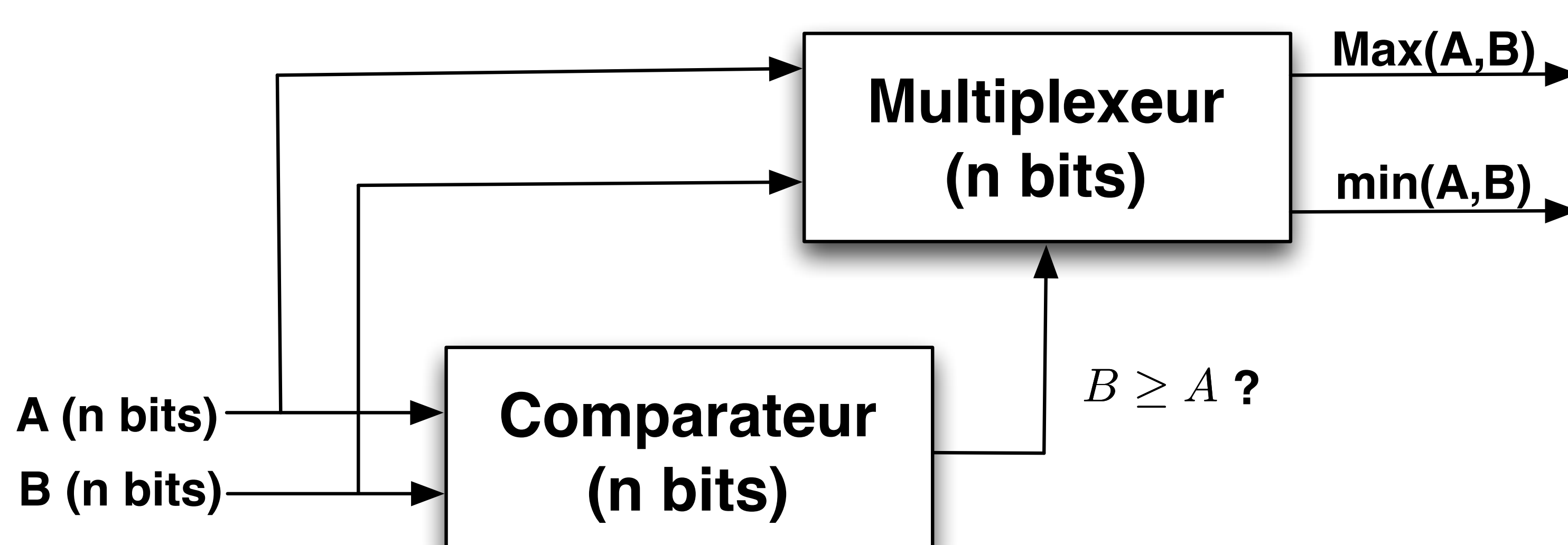
On peut facilement généraliser à des entrées avec n'importe quel nombre de bits.

### Multiplexeur (binaire)

Si  $x$  alors  $s = b$   
Sinon,  $s = a$



Finalement, pour déterminer les maximum et minimum de 2 entrées  $A$  et  $B$  :



Le circuit de gauche est une façon « d'implémenter » les trieurs (les « ronds blancs » des réseaux de tri)