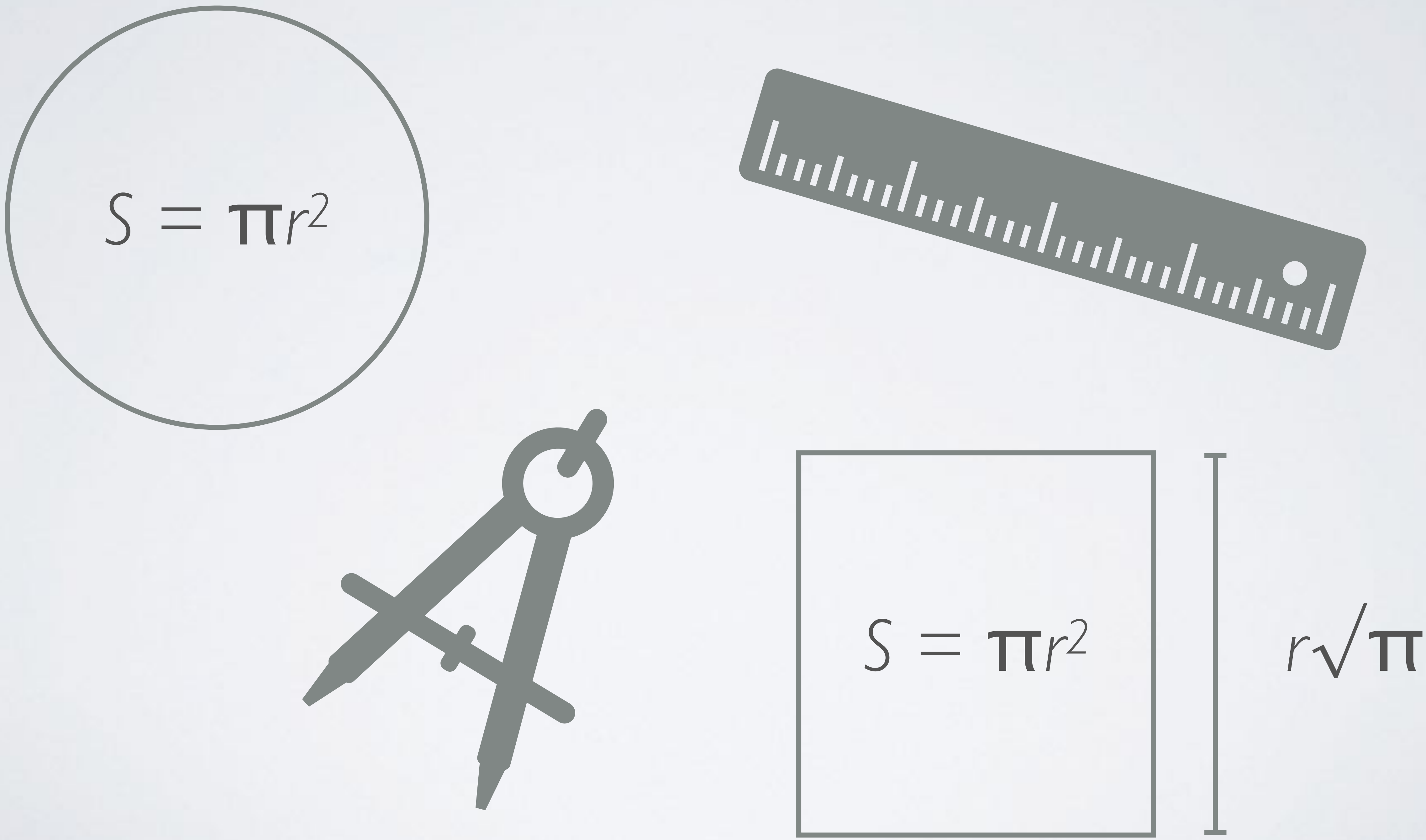
INTRODUCTION À L’INFORMATIQUE

Antonio E. Porreca
https://aeporreca.org/teaching

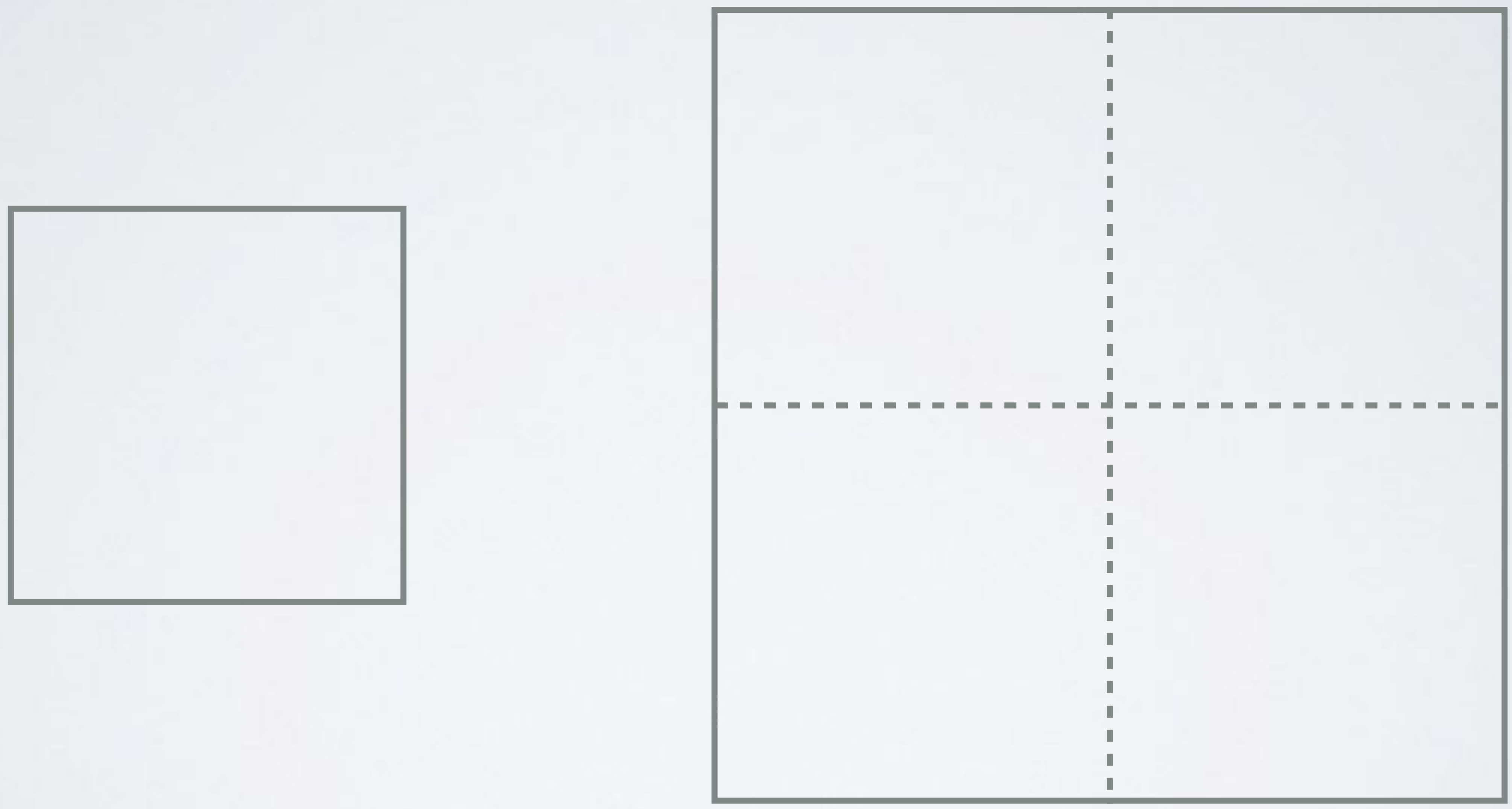
QUADRATURE DU CERCLE



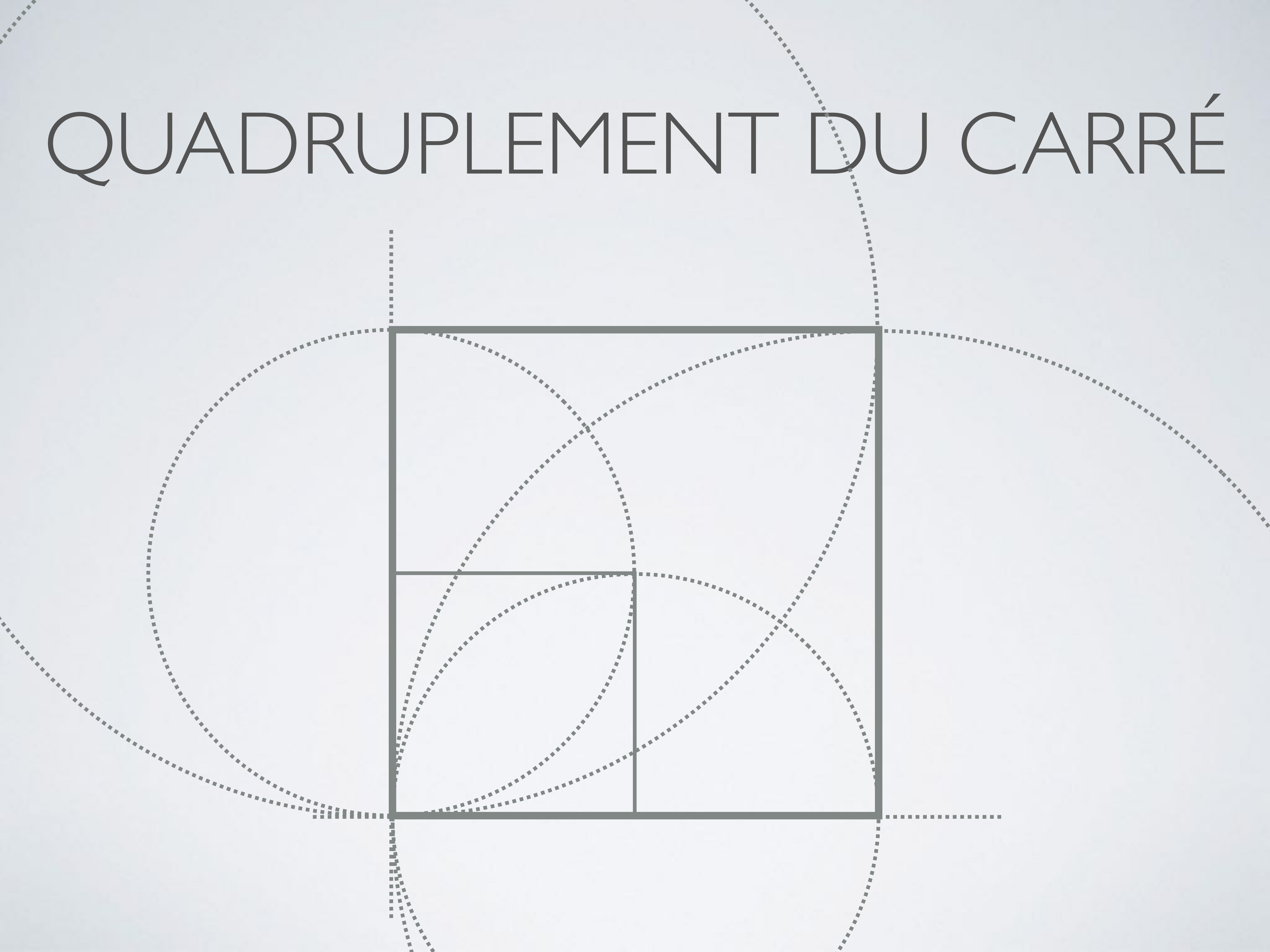
THÉORÈME :
La quadrature du cercle à la règle
et au compas est impossible

–Ferdinand von Lindenmann, 1882

QUADRUPLEMENT DU CARRÉ



QUADRUPLEMENT DU CARRÉ



THÉORÈME :
Le quadruplement du carré à la règle
et au compas est bien possible

–Le petit Aléxandros (9 ans), Grèce antique

POSSIBILITÉ ET IMPOSSIBILITÉ EN MATHÉMATIQUES

- Prouver que quelque chose est bien possible semble plus simple
- (Spoiler : ce n'est pas toujours le cas…)
- Pour prouver que quelque chose est impossible il faut, en général, en donner une définition rigoureuse

CALCULABILITÉ EN INFORMATIQUE

- Les Babyloniens avaient déjà des algorithmes pour faire de l'arithmétique et de l'algèbre (–3000)
- On a dû attendre Alan M. Turing (1936) pour une formalisation satisfaisante de la notion d'algorithme
- Maintenant on sait qu'il existe des problèmes « bien formés » qui n'ont pas d'algorithme

UN « PETIT » PROBLÈME SANS SOLUTION ALGORITHMIQUE

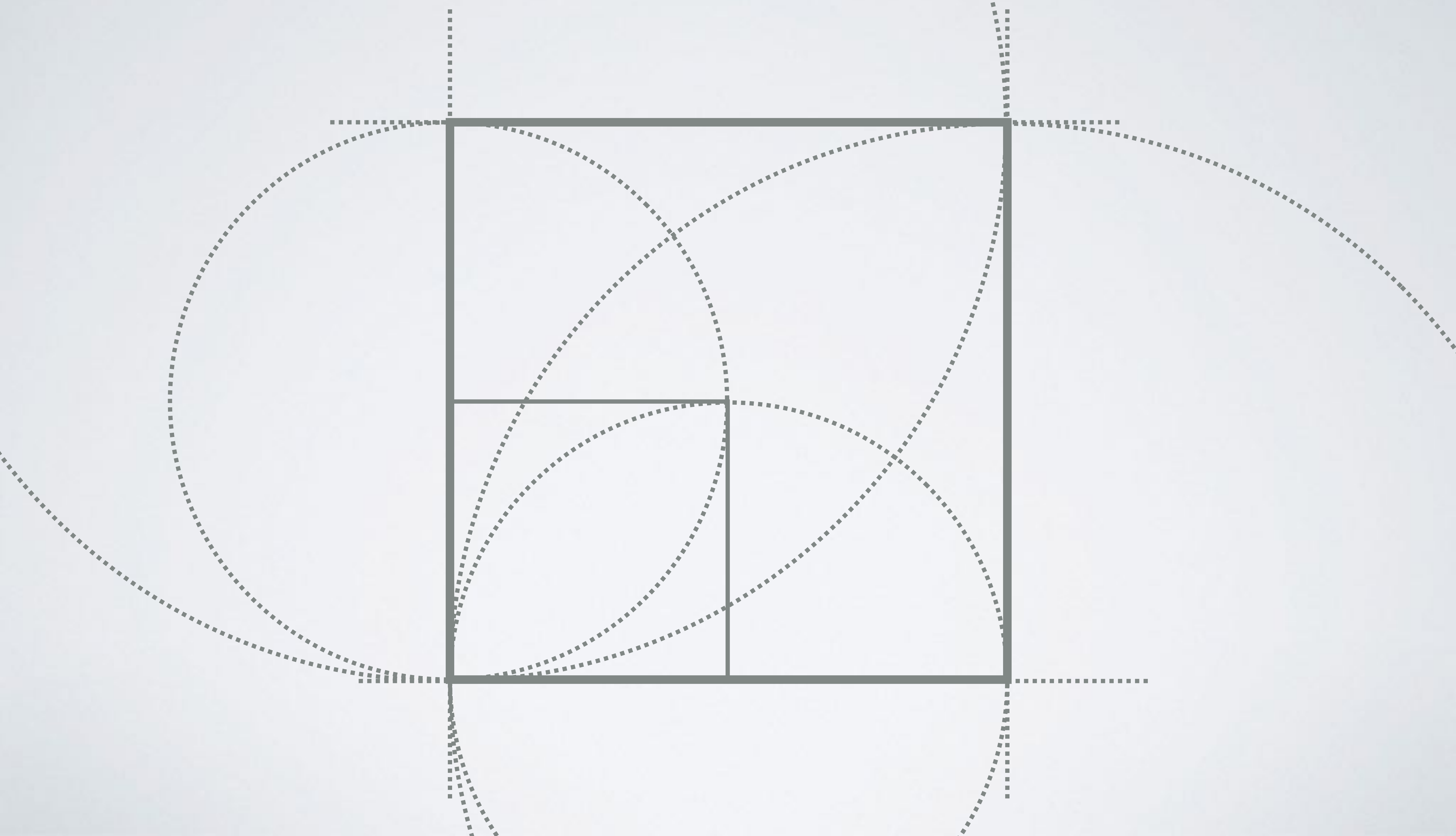
- Entrée : une proposition arithmétique **φ** formalisée

  Par exemple : $(\forall n>2)\ (\nexists x,y,z \neq 0)\ (x^n + y^n = z^n)$

- Sortie : **oui** si on peut prouver **φ**, **no** si on ne peut pas

- Ce problème n'a pas d'algorithme !

QUADRUPLEMENT DU CARRÉ



EFFICACITÉ DES CONSTRUCTIONS À LA RÈGLE ET AU COMPAS

- Six « opérations » pour quadrupler le carré
- Est-ce qu'on peut faire mieux que ça ?
- Est-ce qu'on peut prouver qu'on ne peut pas faire mieux que ça ?
- On peut mesurer aussi la quantité d'espace (taille du papier) utilisé par une construction

EFFICACITÉ DES ALGORITHMES ET COMPLEXITÉ DES PROBLÈMES

- Le nombre d'opérations dépend, de quelque façon, de la taille des données d'entrée
- Taille $n \rightarrow f(n)$ opérations (exemple : $f(n) = n^2$ ou $f(n) = n + 5$)
- Est-ce qu'on peut faire la même chose en moins de $f(n)$ opérations ?
- Possible définition : complexité d'un problème = efficacité du meilleur algorithme pour le résoudre
- On peut aussi mesurer l'espace (quantité de mémoire) utilisé par un algorithme (exemple : n bits au-delà de la taille des données)

ALGORITHMES EFFICACES OU PAS

$$\begin{array}{r l} 234 & + \\ 281 & = \\ \hline 515 & \end{array}$$

234
235
236
237
...
513
514
515

approximativement
3 opérations

au moins
281 opérations

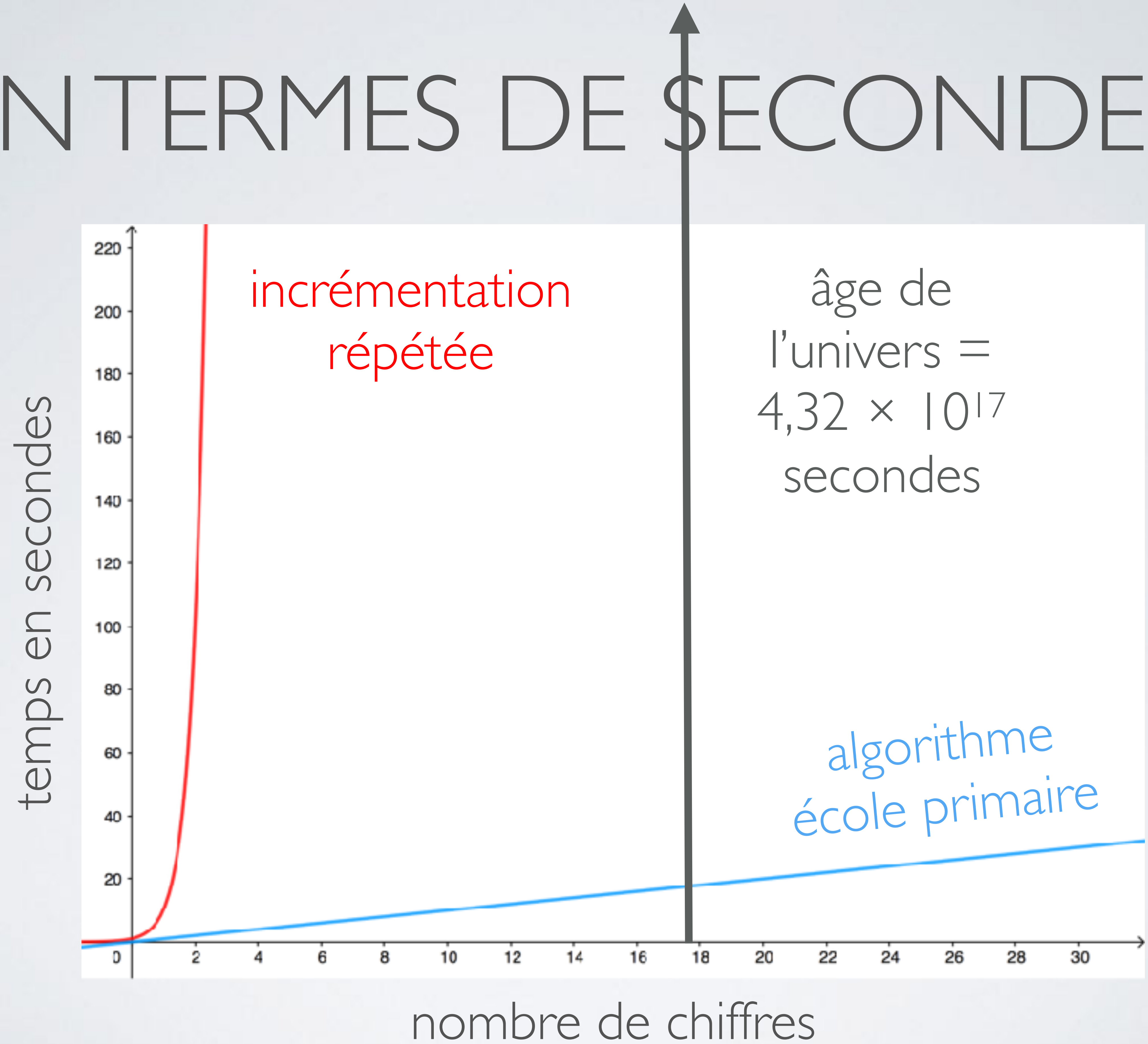
EFFICACITÉ DES ALGORITHMES POUR L'ADDITION

- Algorithme de l'école primaire : approximativement n opérations pour additionner des nombres de n chiffres

- Algorithme de l'« incrémentation répétée » : au moins 10^n opérations pour des nombres de n chiffres !

- Supposons qu'on fasse une opération par seconde

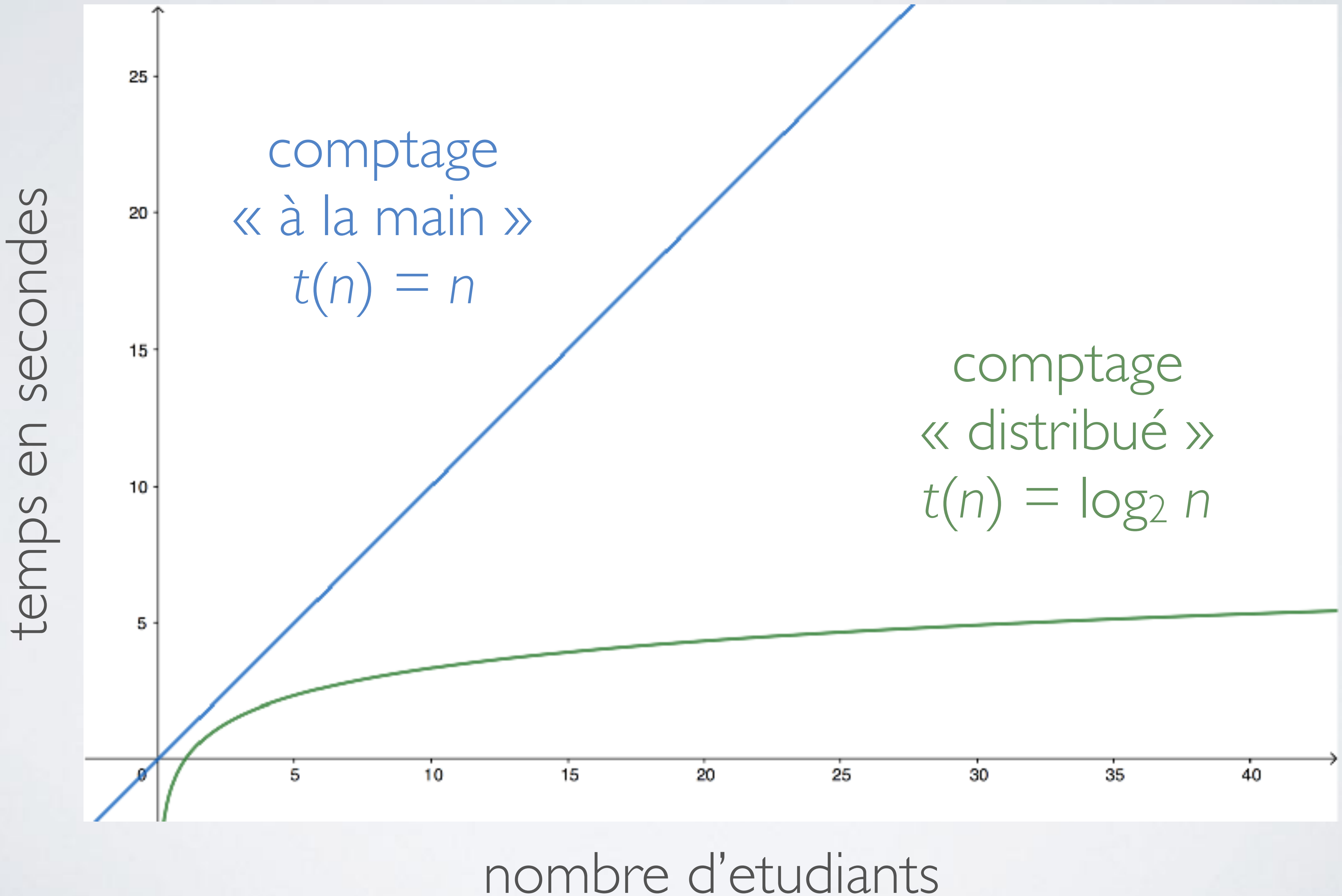
EN TERMES DE SECONDES



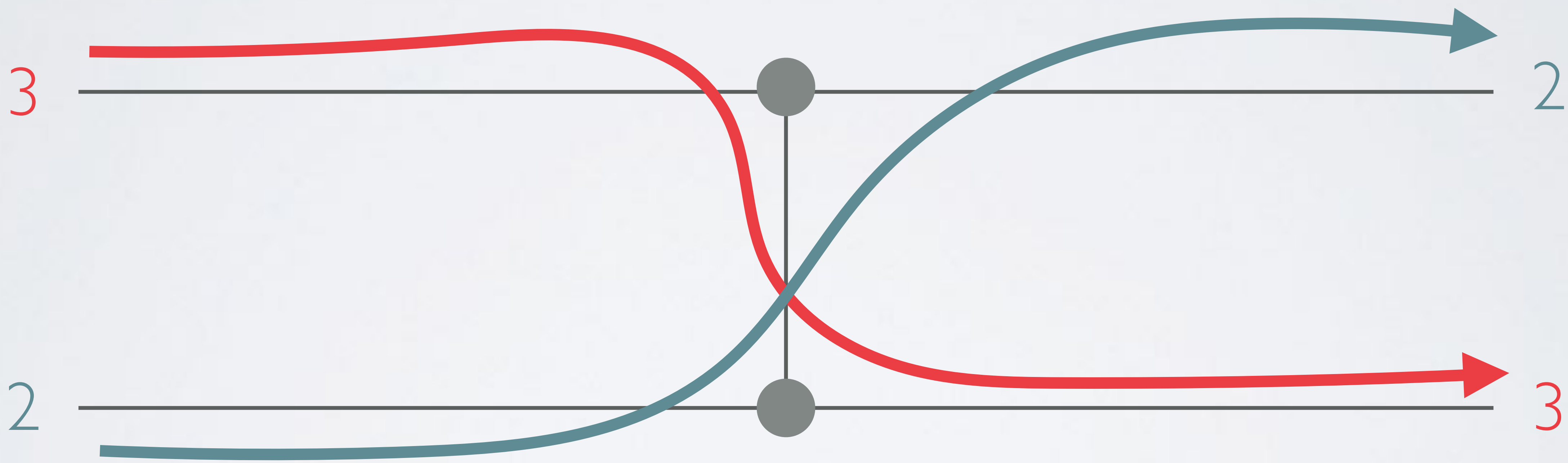
COMBIEN D'ÉTUDIANTS Y A-T-IL DANS LA SALLE ?

- Chaque étudiant commence avec le nombre 1 en tête
- Tant qu'il reste au moins deux étudiants debout :
    - Chaque étudiant encore debout cherche du regard un autre étudiant debout
    - Les deux étudiants s'échangent le nombre qu'ils ont en tête
    - L'un des deux étudiants s'assoit
    - L'autre additionne les deux nombres qu'il mémorise
- Le dernier étudiant debout crie le nombre qu'il a en tête

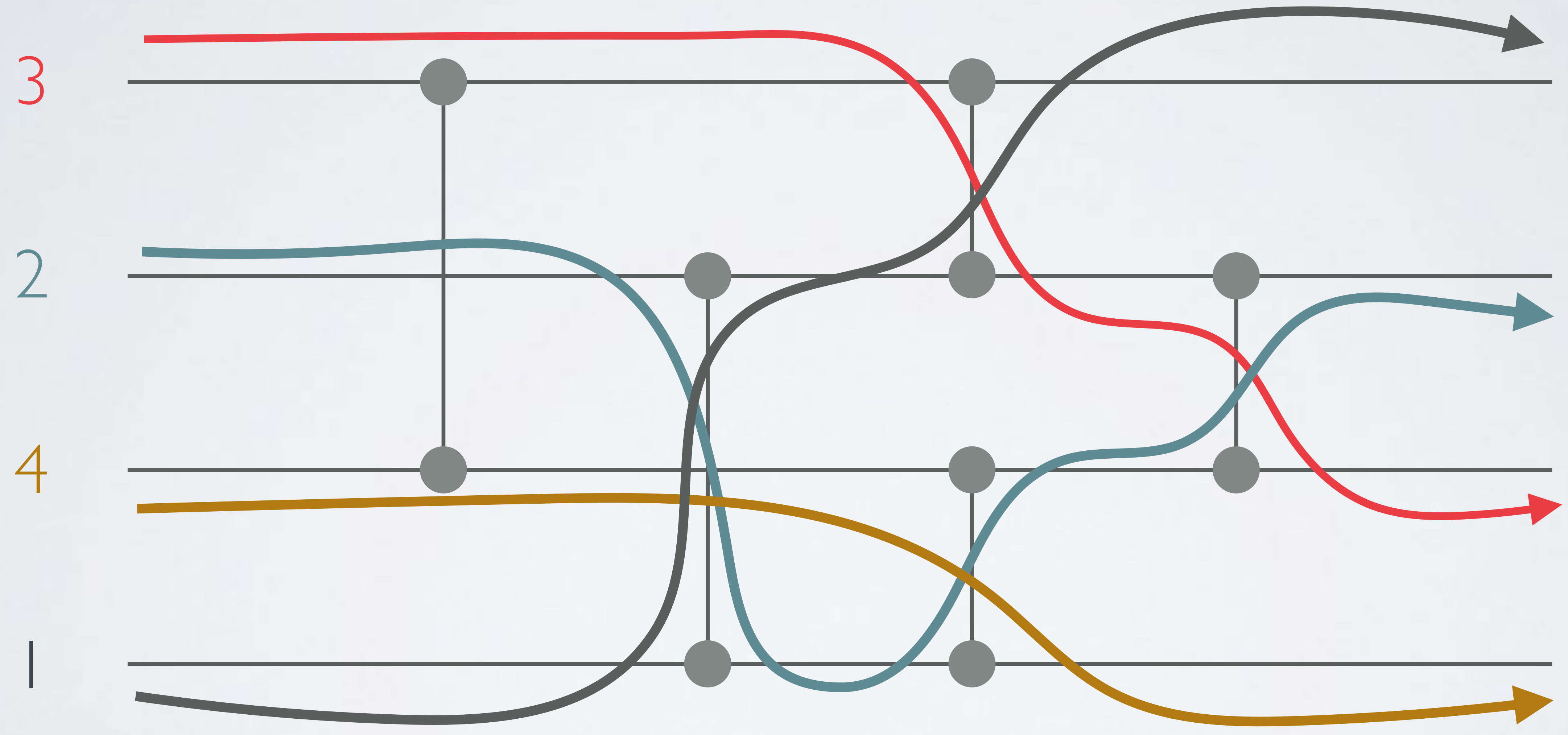
EFFICACITÉ DU COMPTAGE



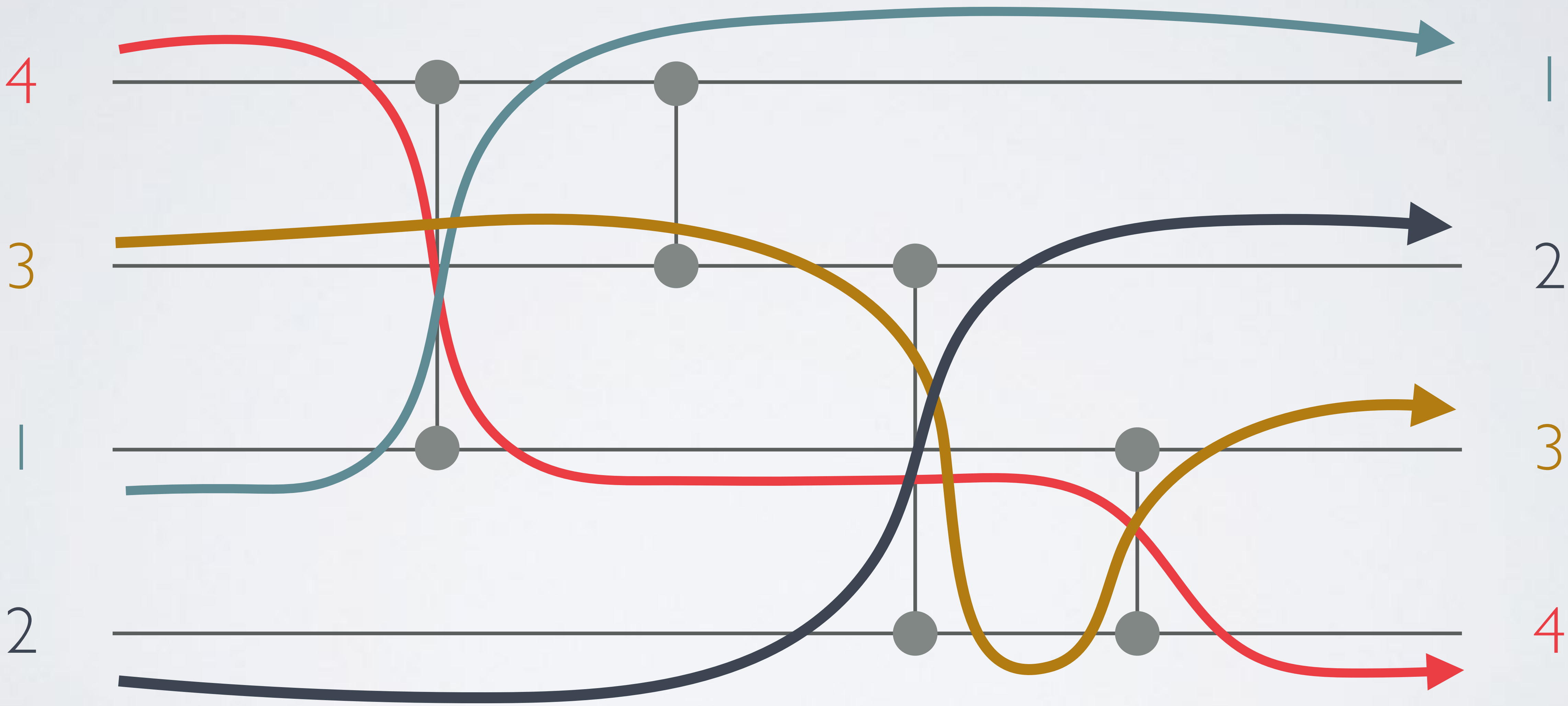
RÉSEAUX DE TRI



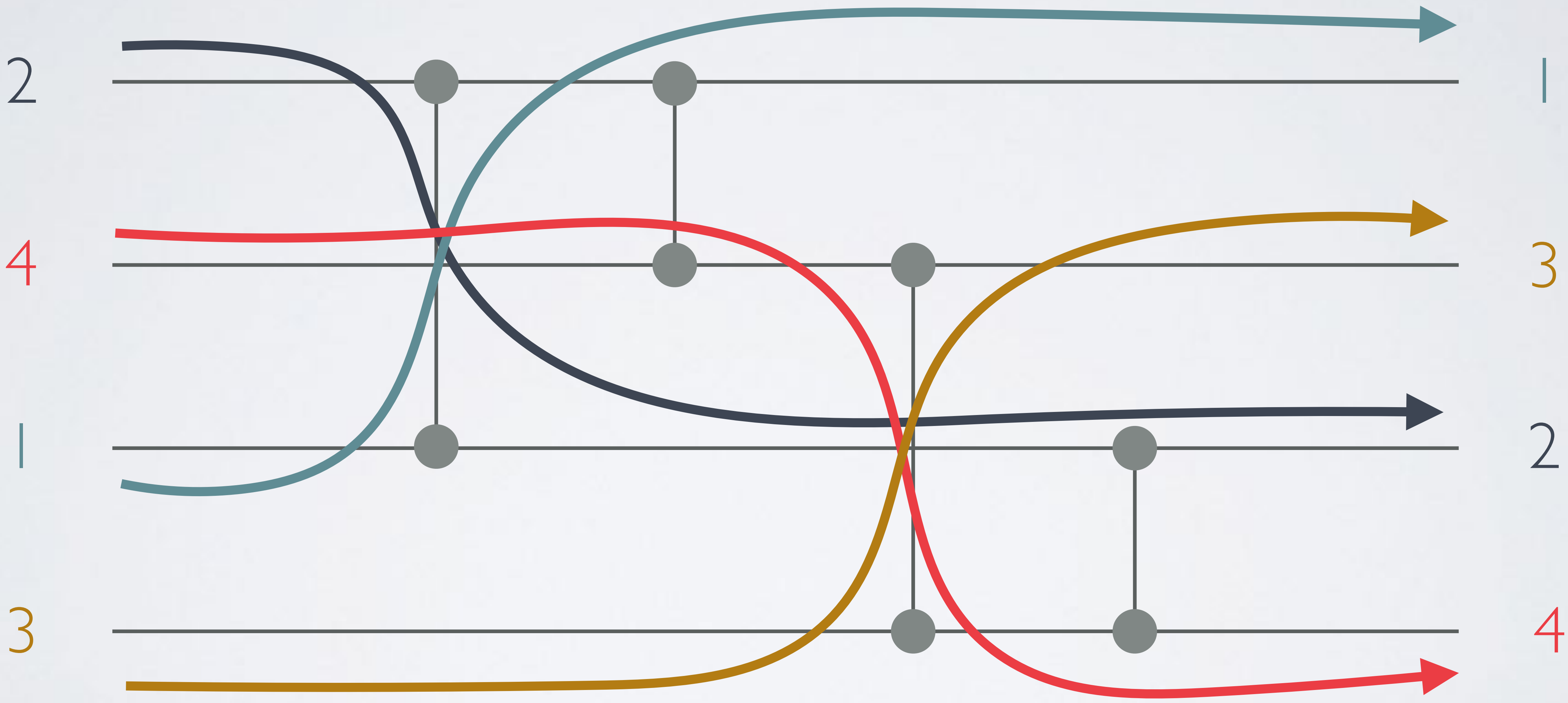
UN EXEMPLE



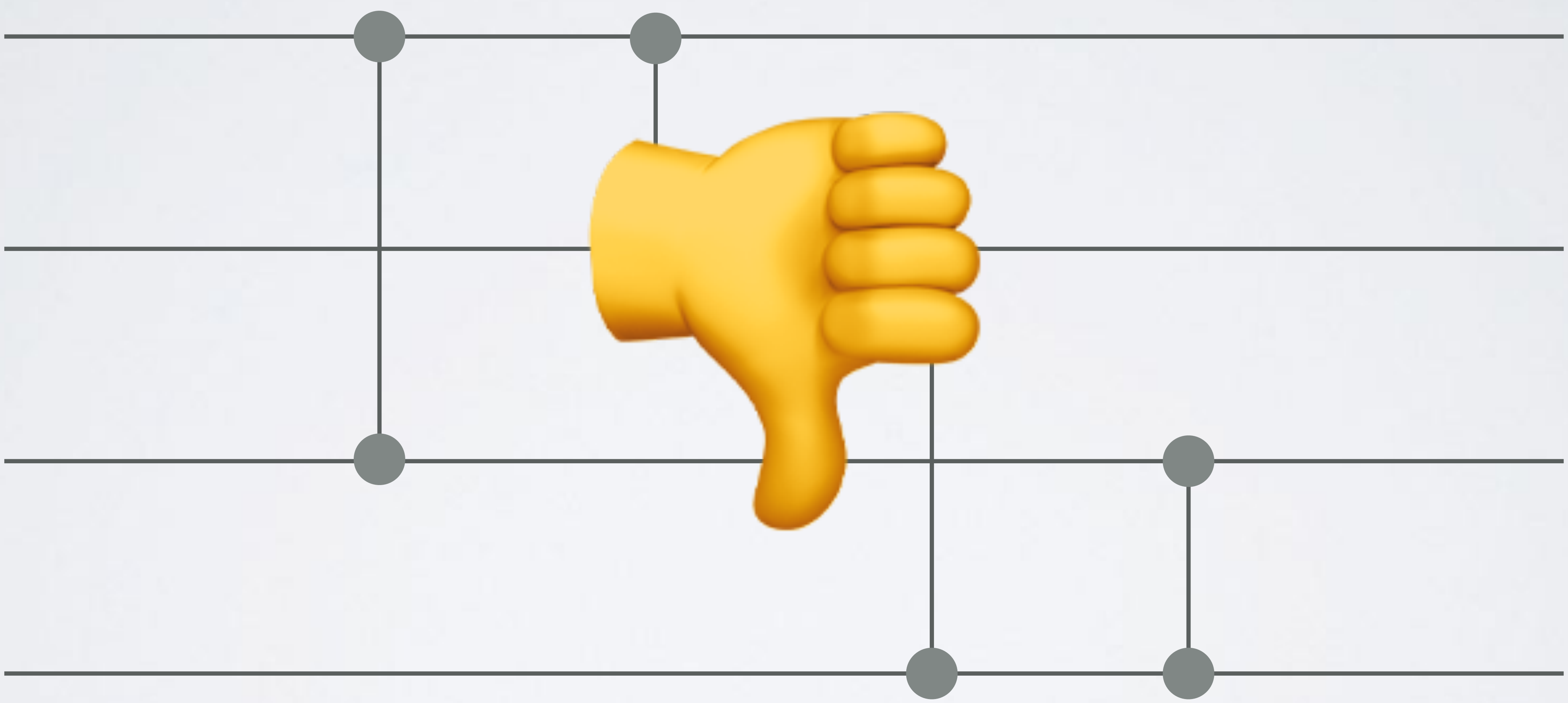
UN AUTRE RÉSEAU



AVEC D'AUTRES DONNÉES



PAS UN RÉSEAU DE TRI !



VÉRIFIER SI UN RÉSEAU EST DE TRI

- Essayer avec toutes les entrées de n entiers naturels : mais il y en a infinies !
- Peu importent les valeurs, seul l'ordre compte : tester avec toutes les permutations de $1, \ldots, n$
- Mais il y en a $n!$ (n factoriel), qui est même pire que 10^n

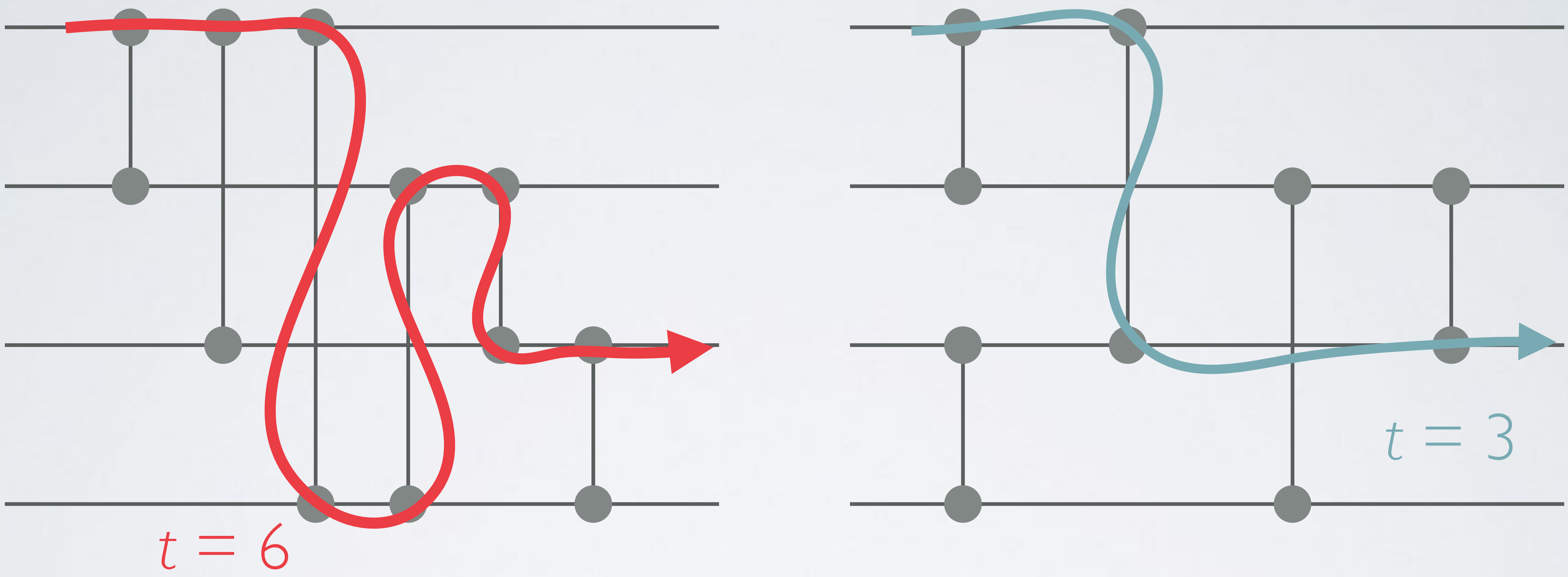
THÉORÈME DU 0-1

- Si le réseau est correcte pour toutes les entrées qui consistent de 0s et 1s, alors il est correcte pour toute entrée

- Mais il y a 2^n entrées de 0s et 1s… mieux que $n!$ et10^n, mais c'est quand même trop

- Parfois on préfère utiliser des maths un peu plus sophistiquées pour gagner du temps

EFFICACITÉ DES RÉSEAUX DE TRI



Deux réseaux corrects,
lequel préférez-vous ?

RÉSOUDRE UN PROBLÈME

- On cherche un algorithme
- On le décrit précisément, de manière non ambigüe
- On prouve qu'il est correct
- On vérifie qu'il est efficace (idéalement, on choisit l'algorithme optimal)
- On le met en œuvre
- On le teste