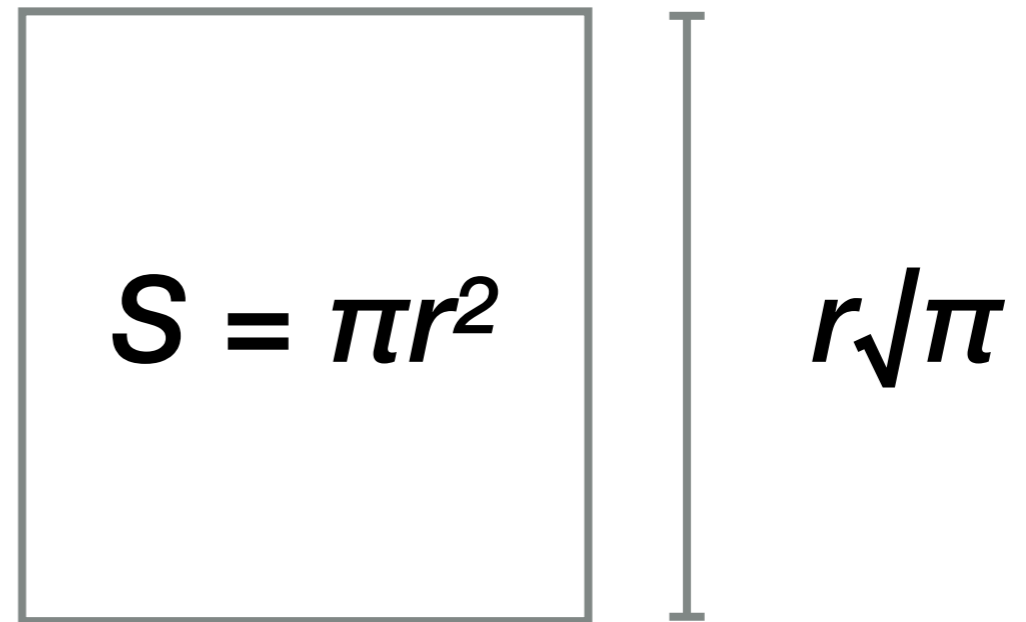
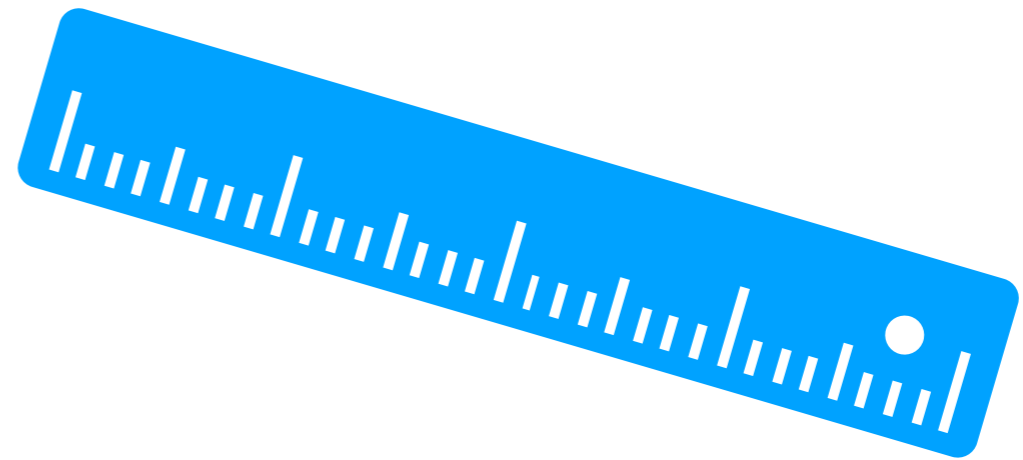
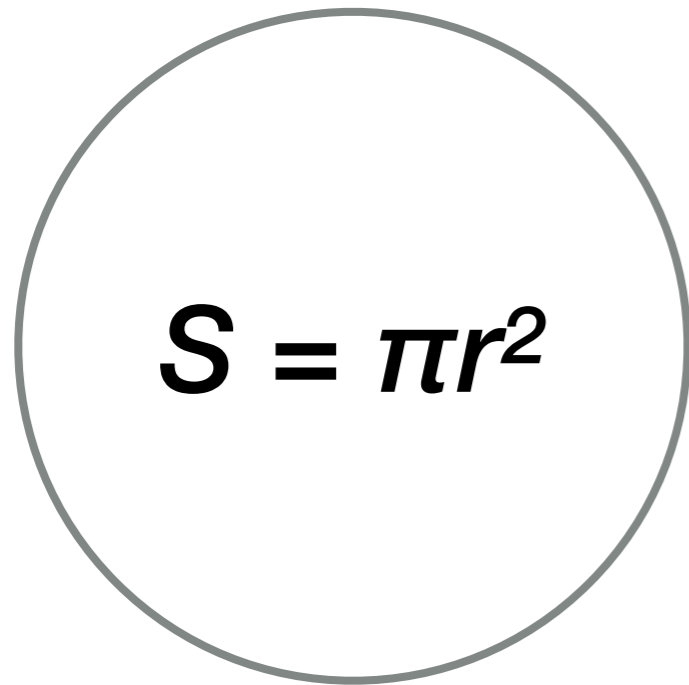


Introduction à l'informatique CM2

Antonio E. Porreca
aeporreca.org/introinfo

Algorithmes !

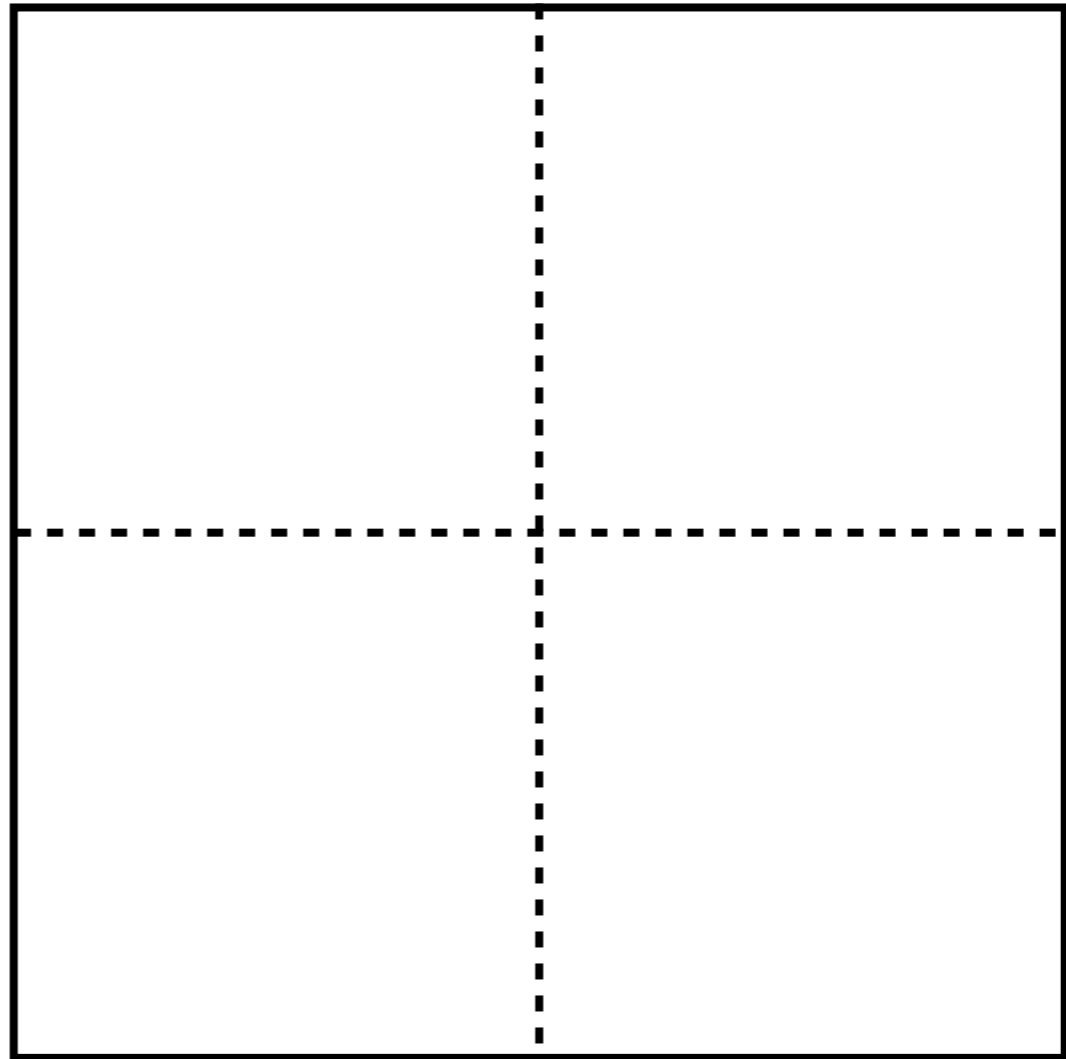
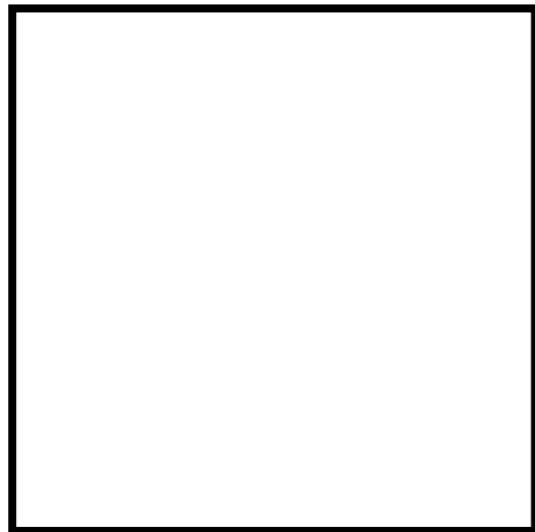
Quadrature du cercle



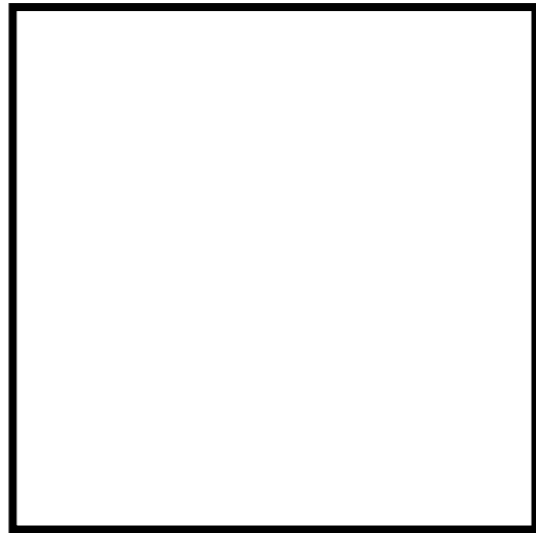
THÉORÈME :
**La quadrature du cercle à la règle
et au compas est impossible**

–Ferdinand von Lindemann, 1882

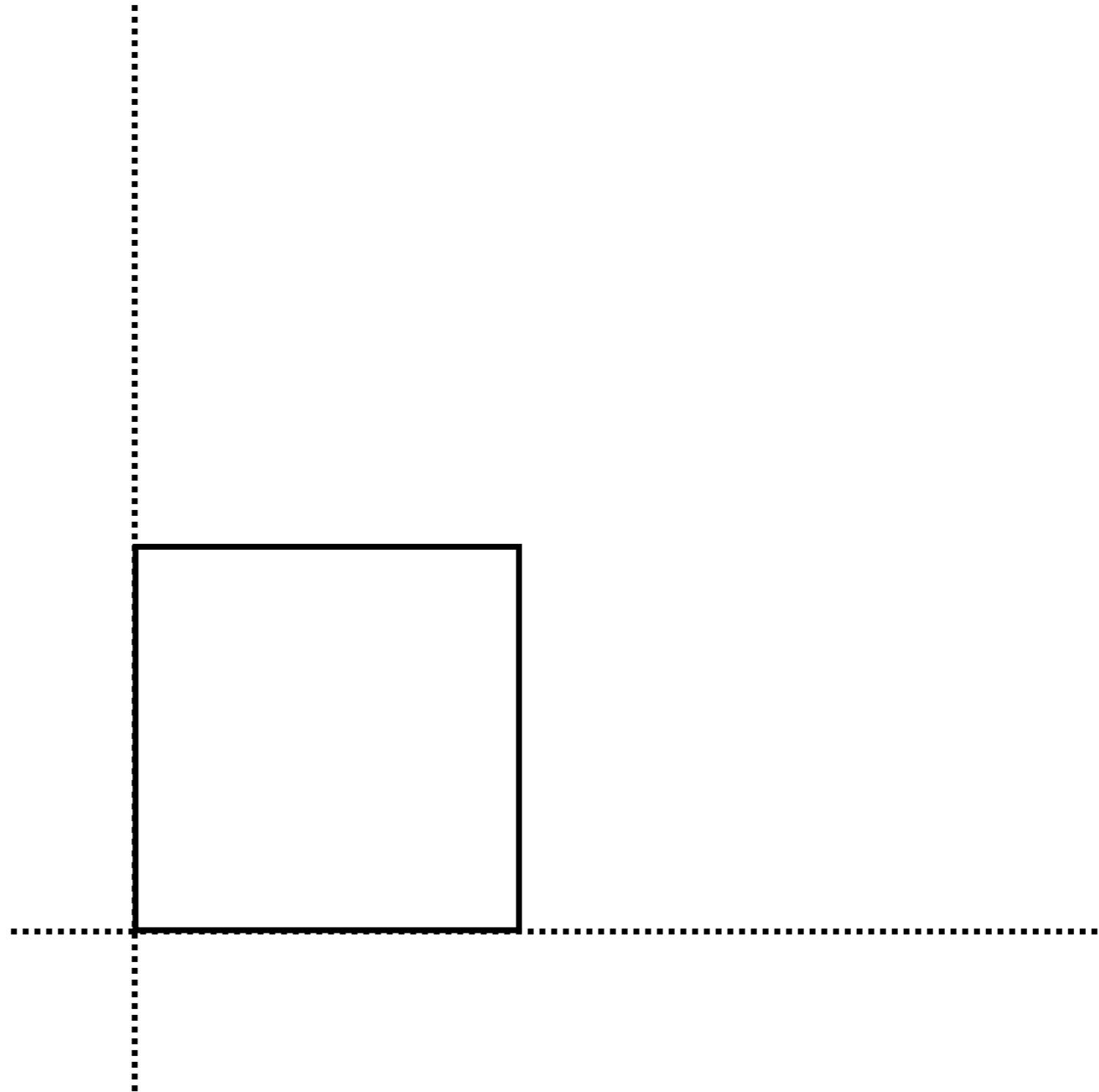
Quadruplement du carré



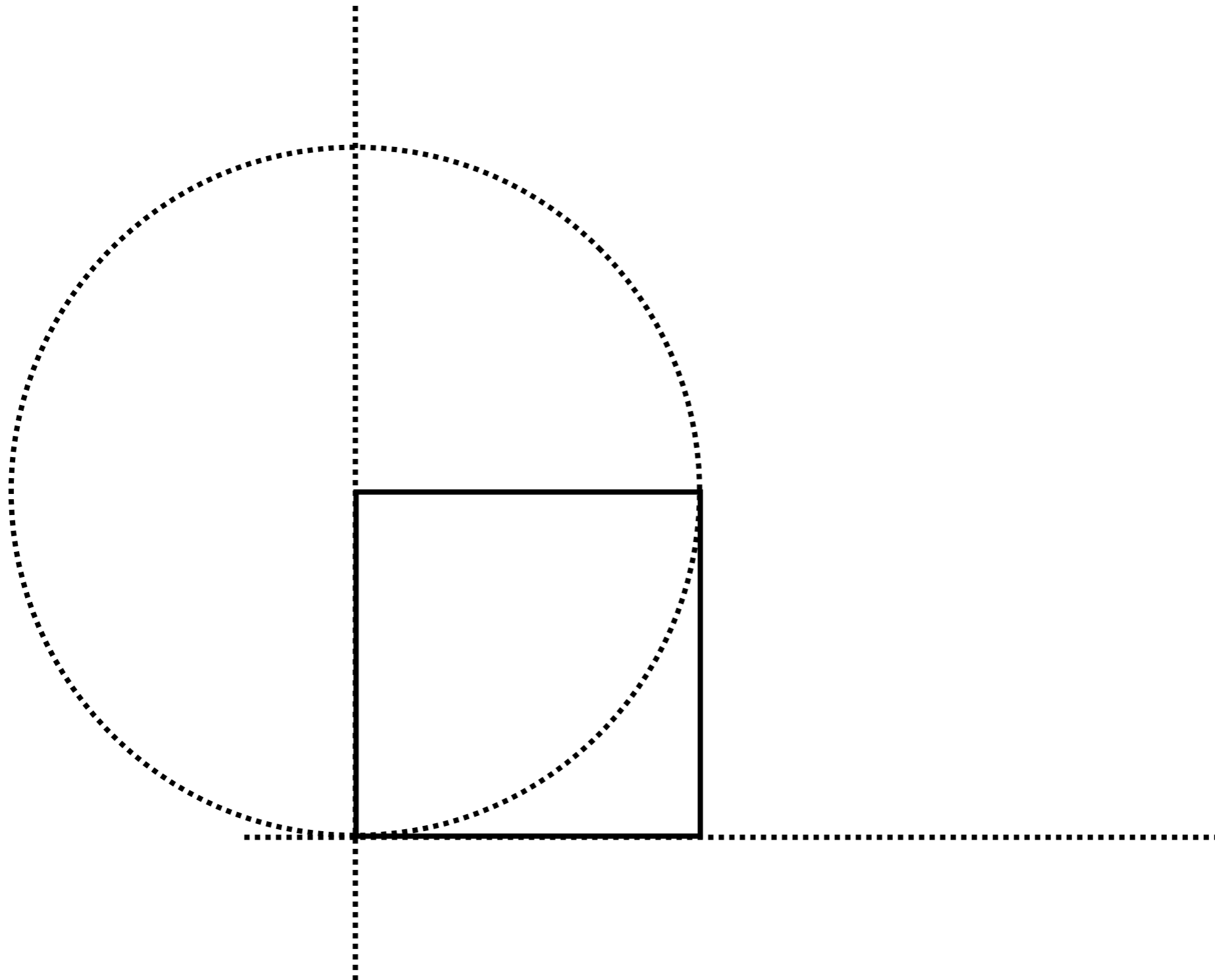
Quadruplement du carré



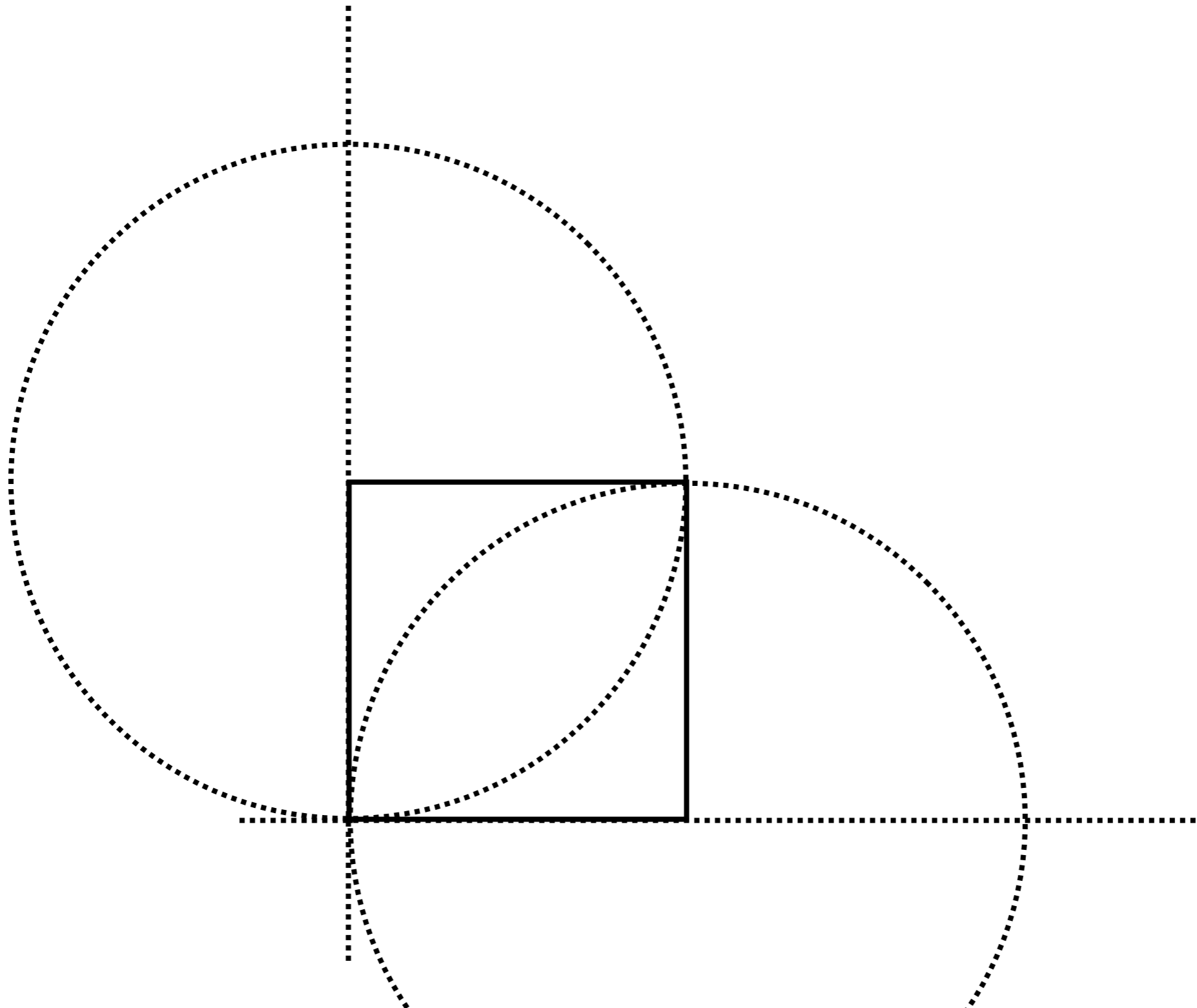
Quadruplement du carré



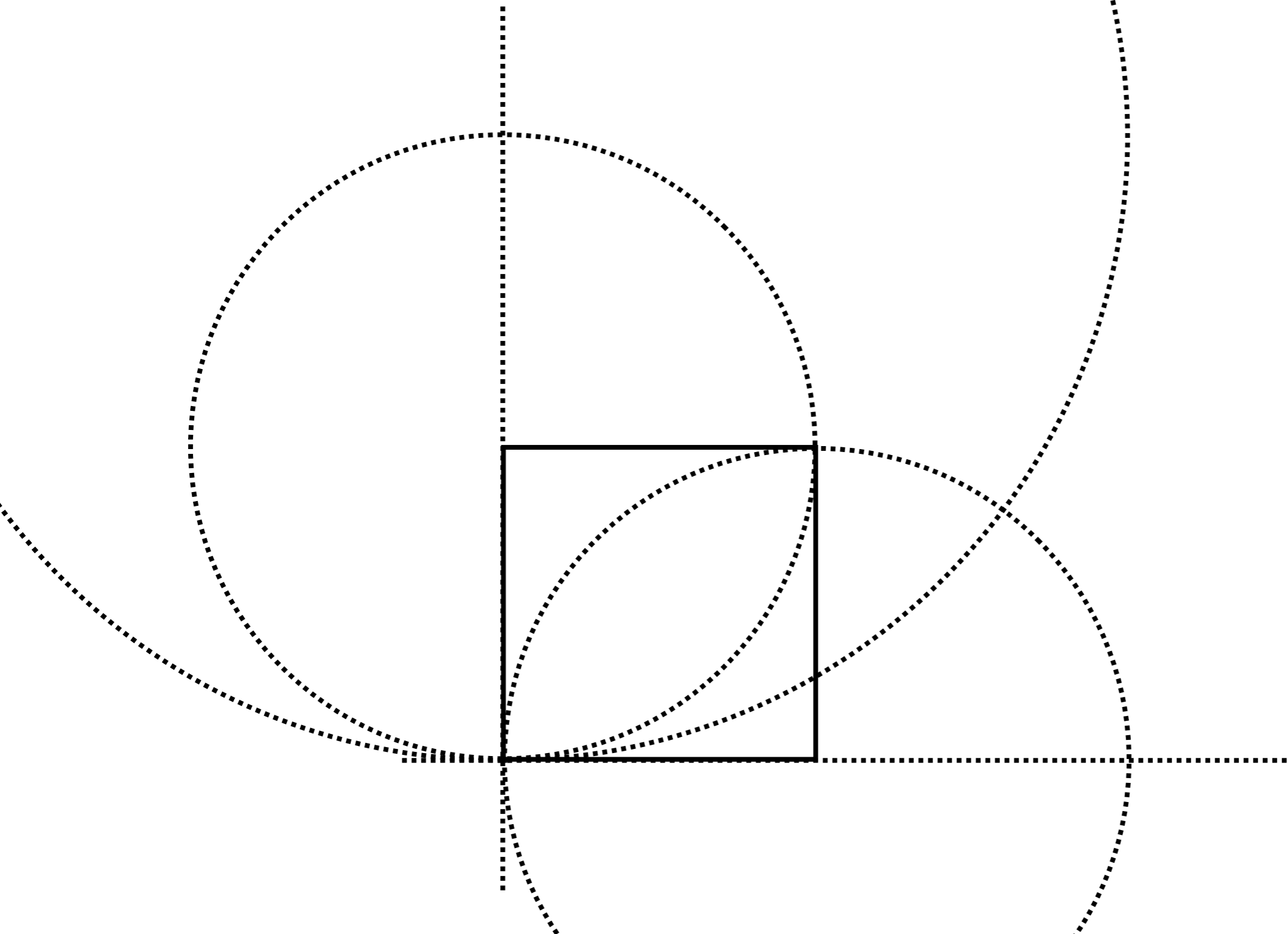
Quadruplement du carré



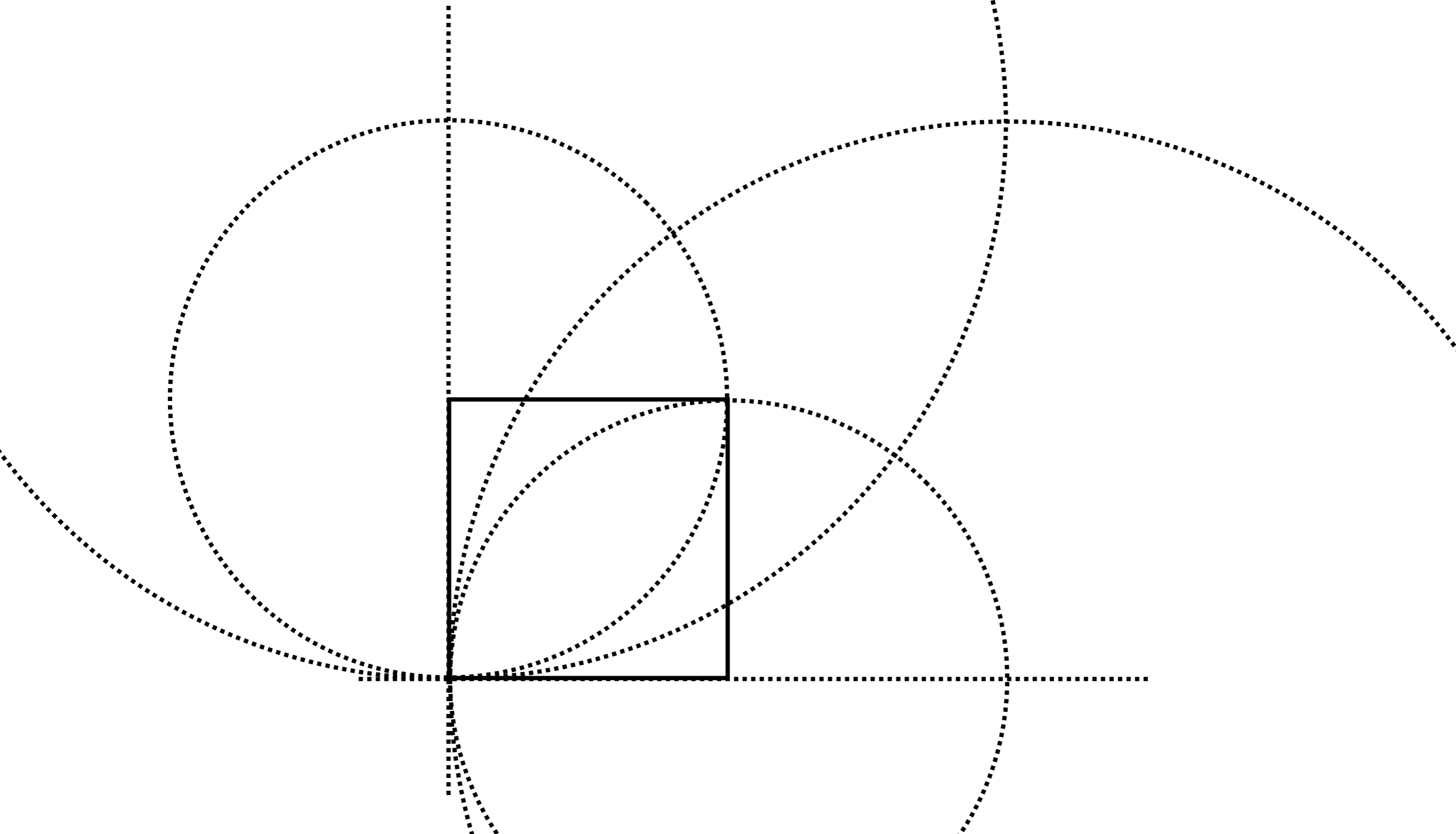
Quadruplement du carré



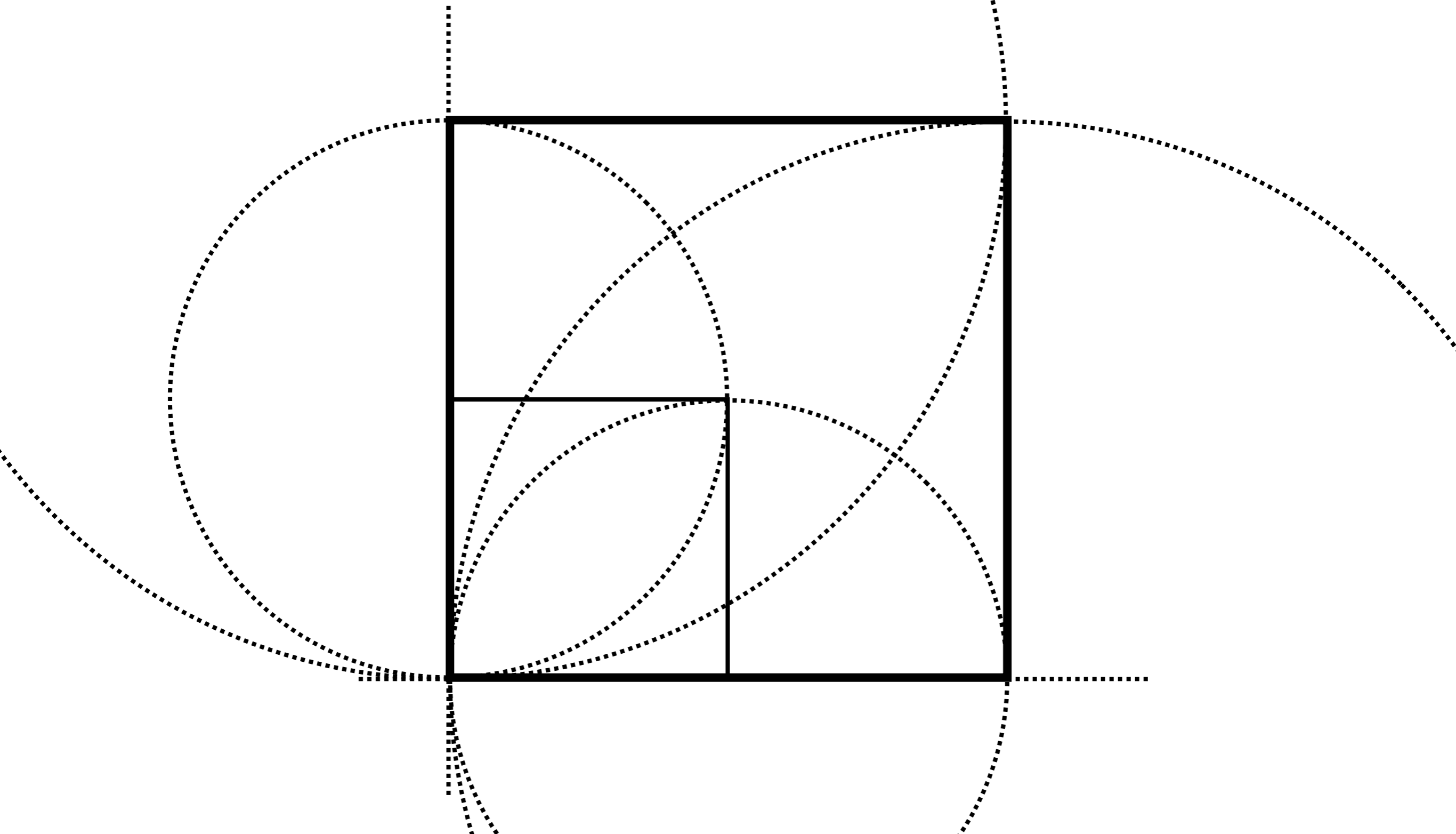
Quadruplement du carré



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

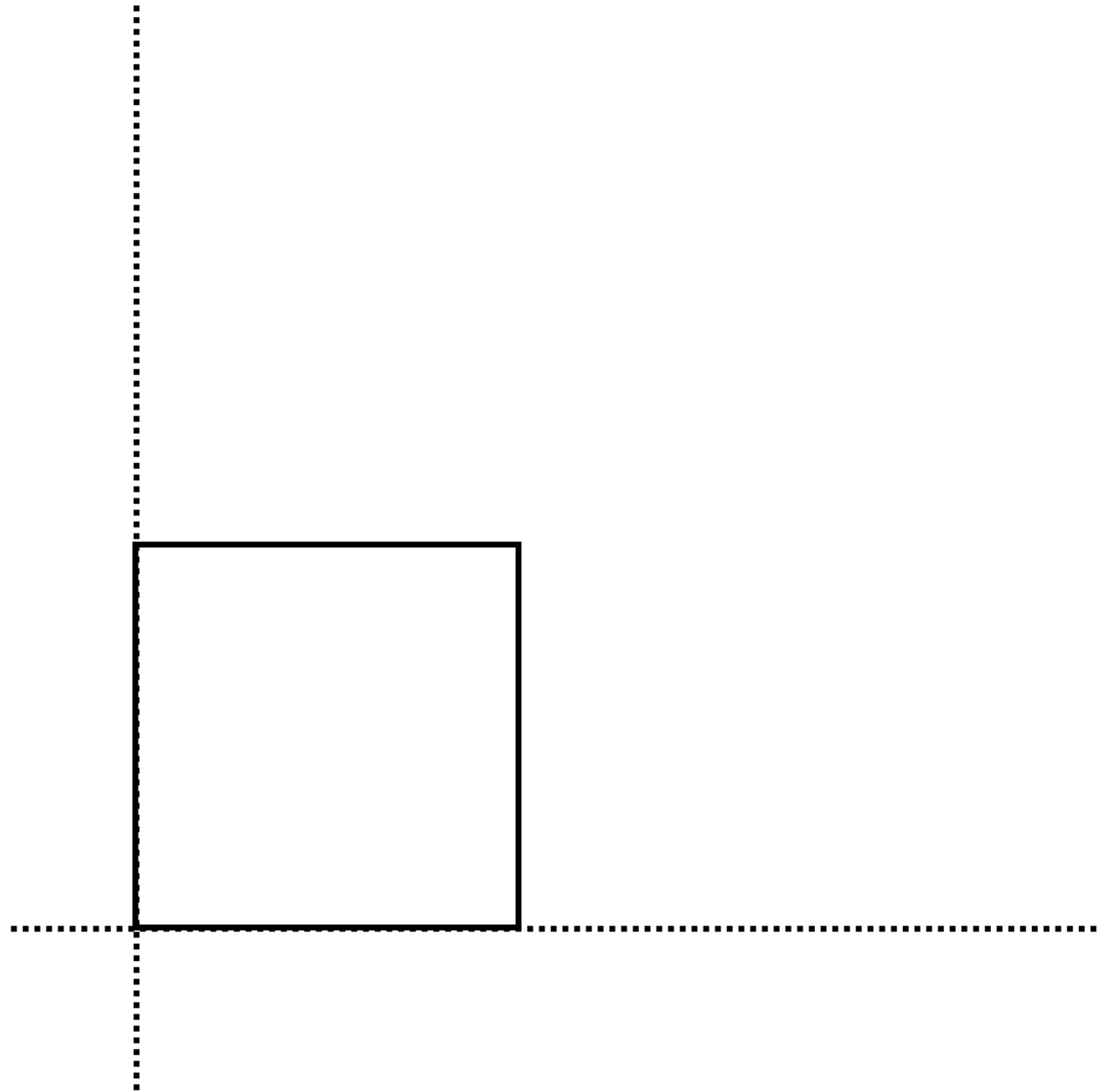
- Les Babyloniens avaient déjà des algorithmes pour faire de l'arithmétique et de l'algèbre (-3000)
- On a dû attendre Alan M. Turing (1936) pour une formalisation satisfaisante de la notion d'algorithme
- Maintenant on sait qu'il existe des problèmes « bien formés » qui n'ont pas d'algorithme

Un « petit » problème sans solution algorithmique

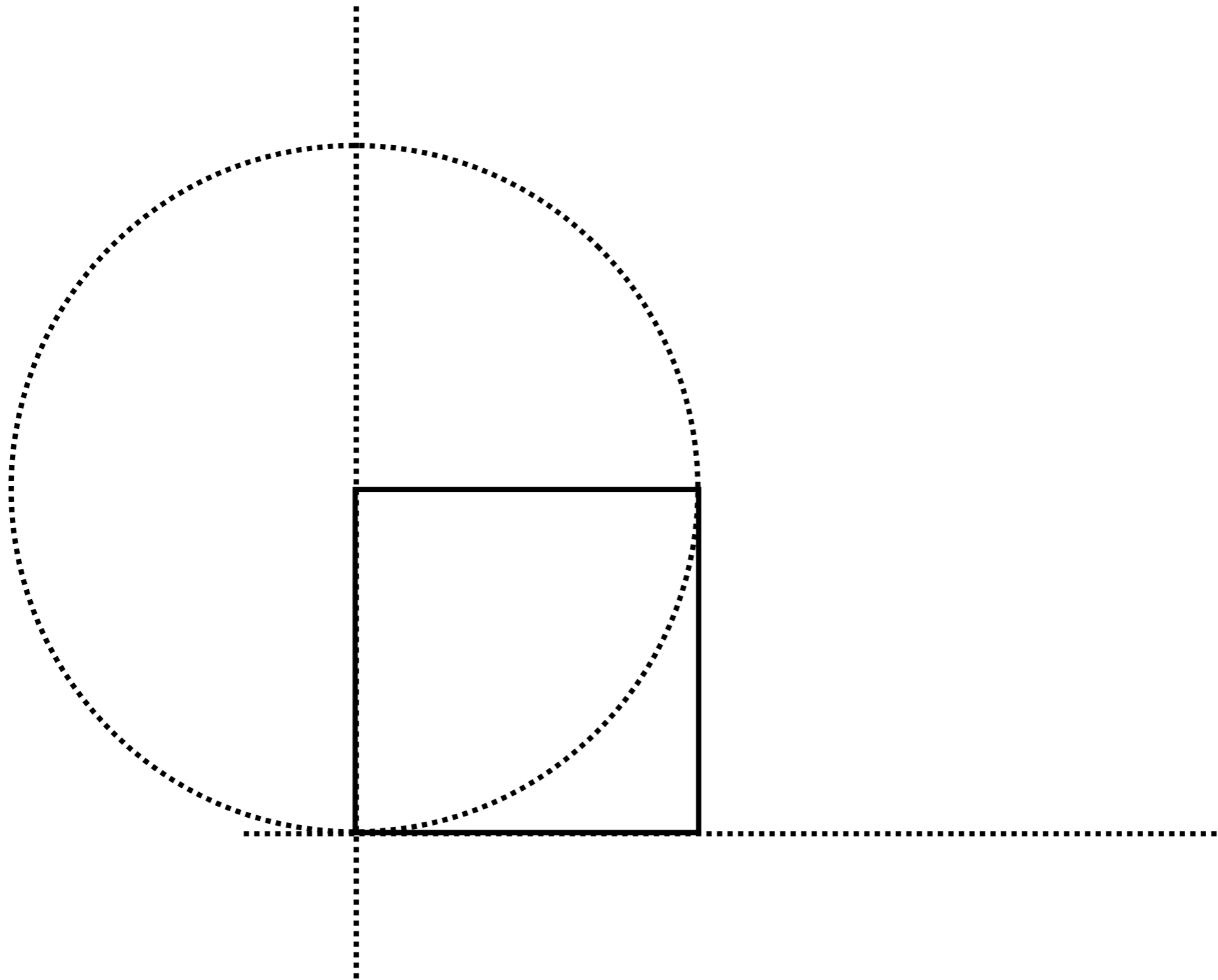
- Entrée : une proposition arithmétique φ formalisée
- Par exemple : $(\forall n > 2)(\nexists x, y, z \neq 0)(x^n + y^n = z^n)$
- Sortie : **oui** si on peut prouver φ , **no** si on ne peut pas
- Ce problème n'a pas d'algorithme !

Efficacité des algorithmes

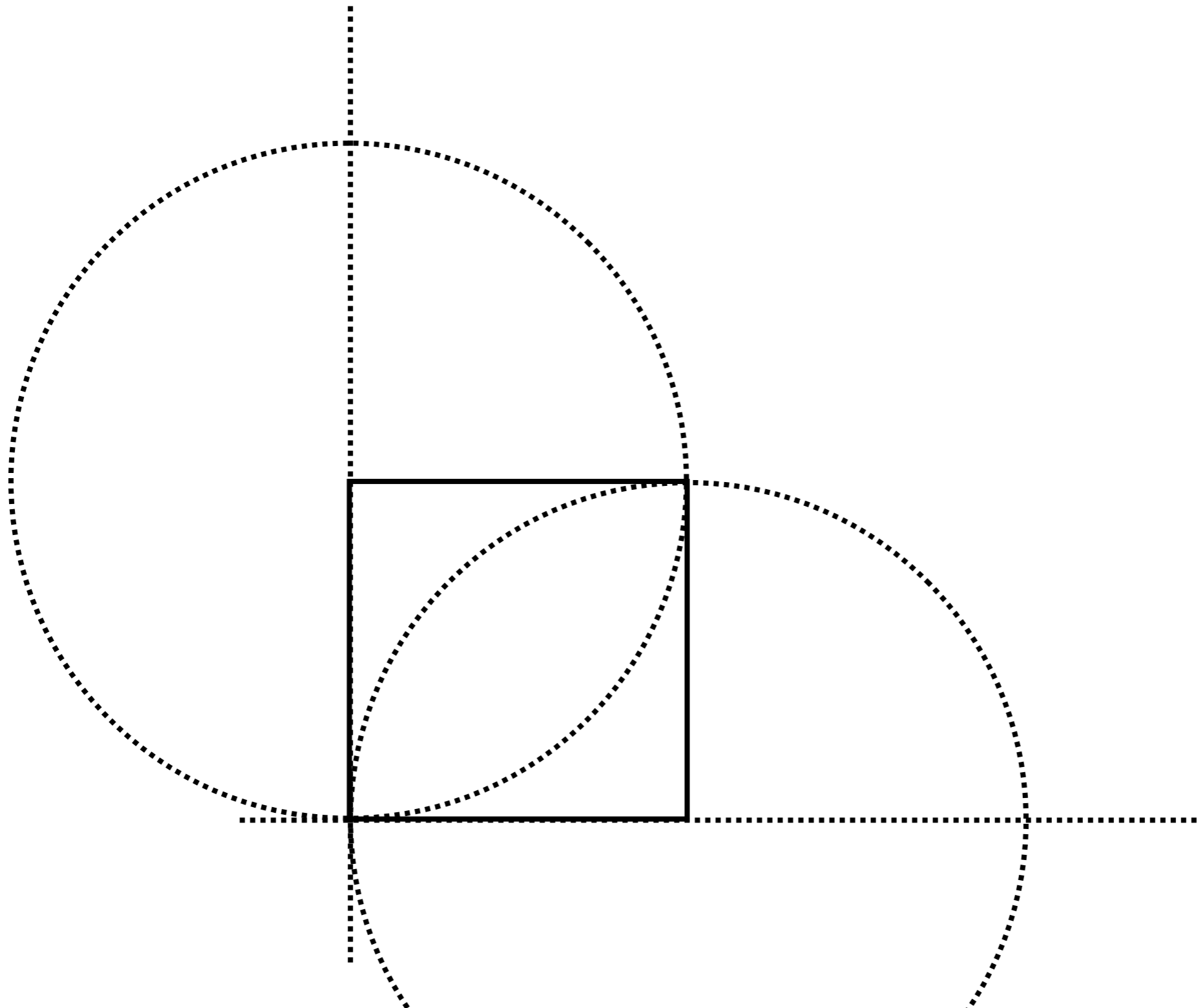
Quadruplement du carré



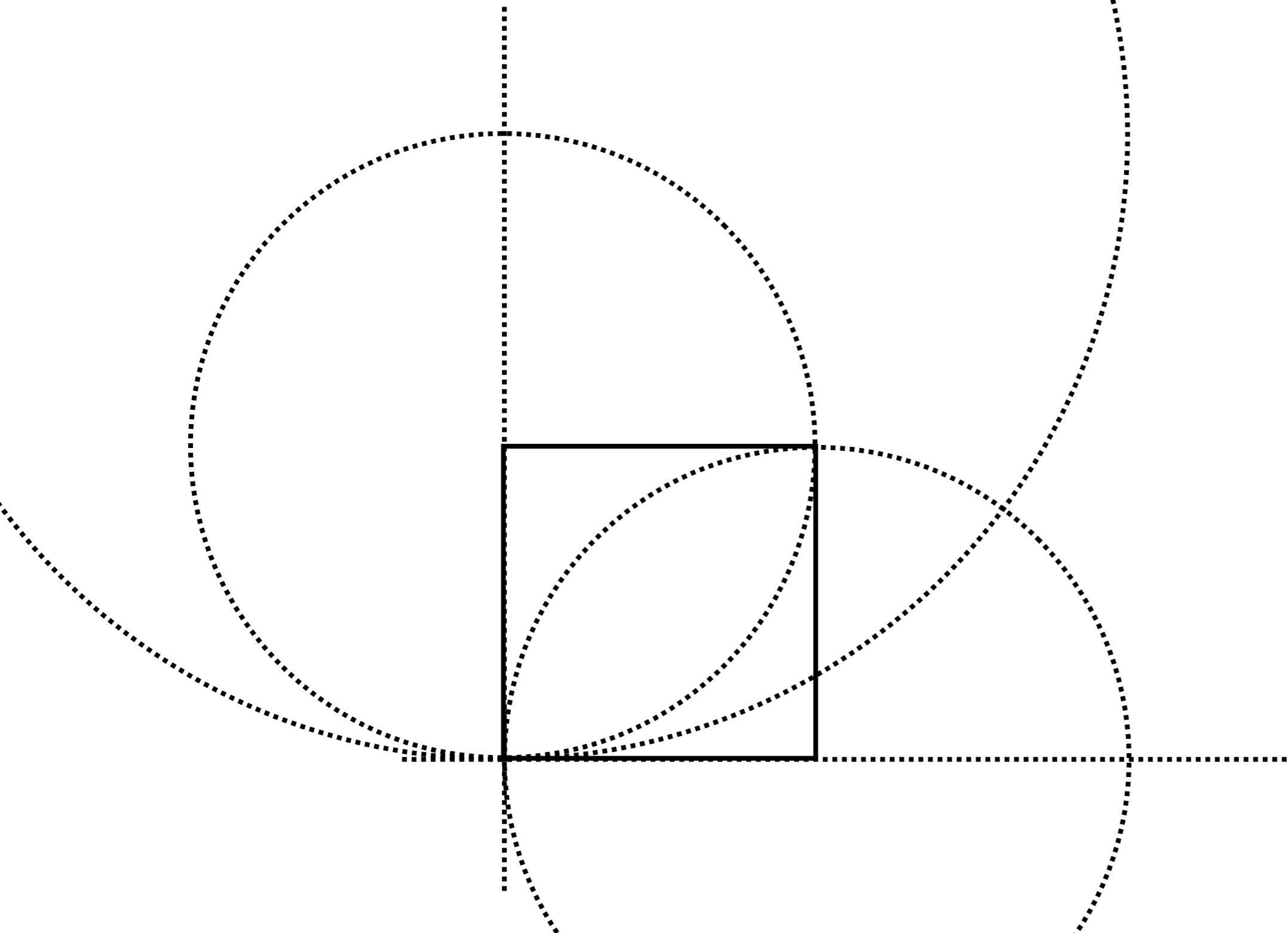
Quadruplement du carré



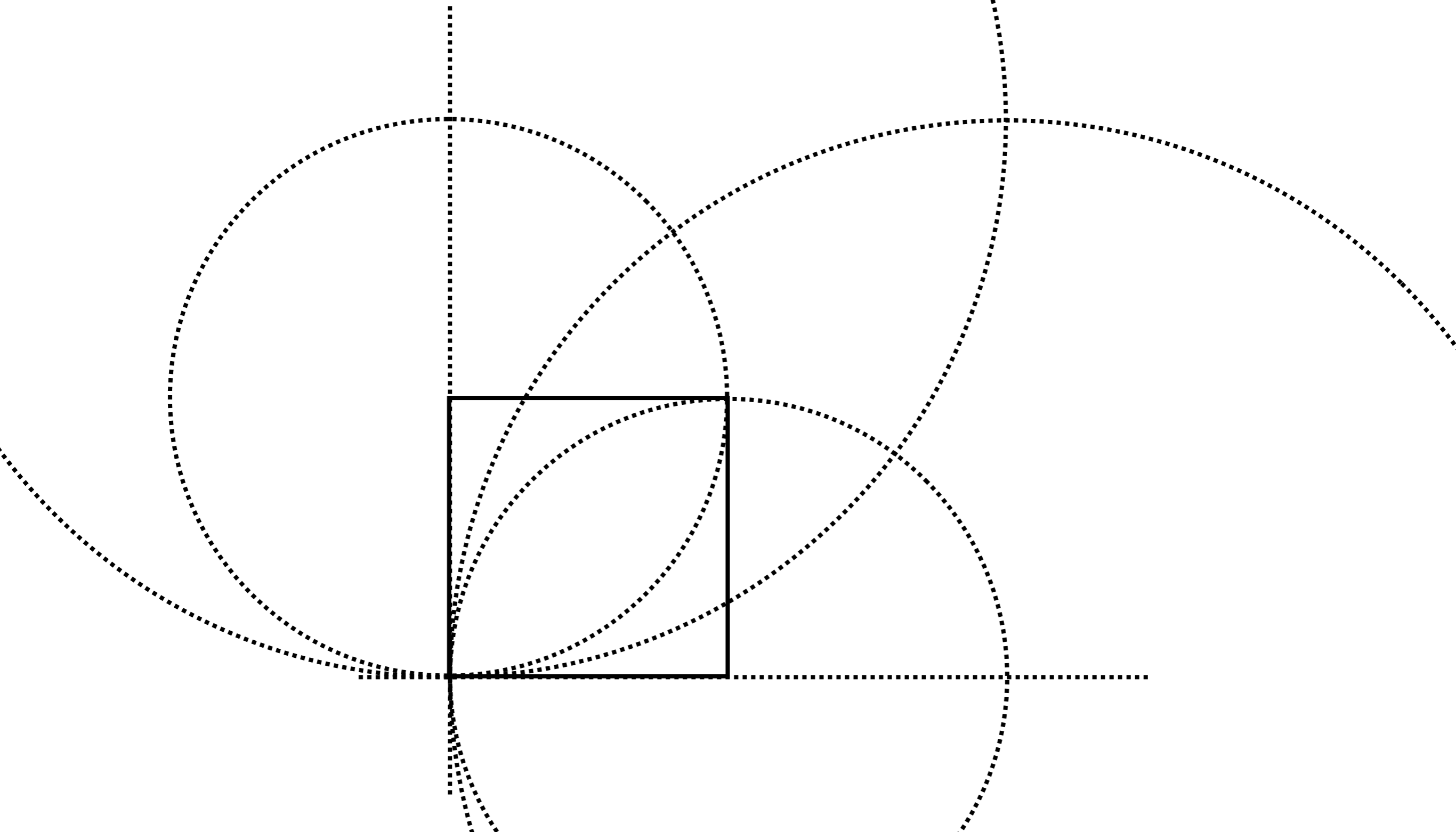
Quadruplement du carré



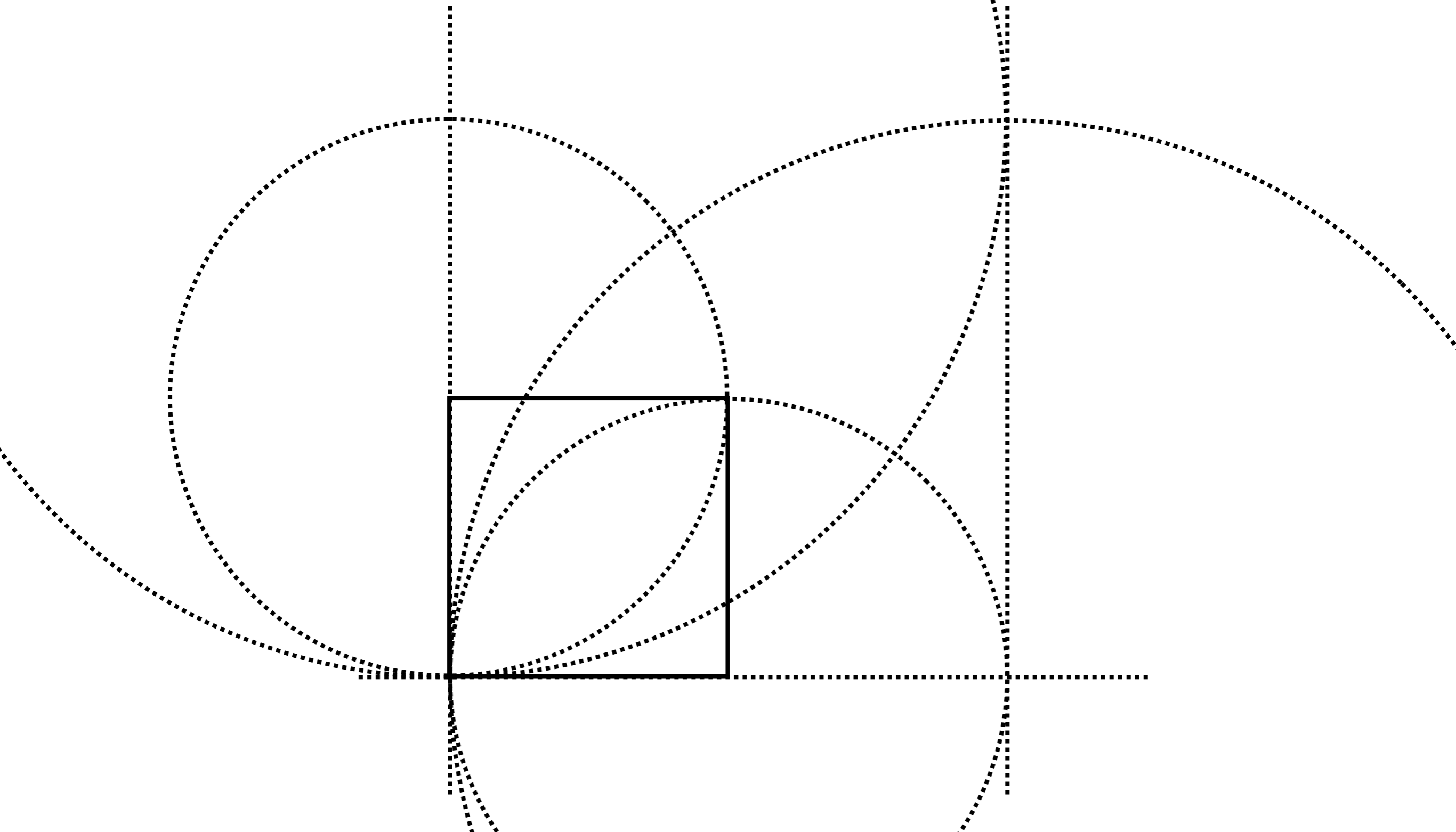
Quadruplement du carré



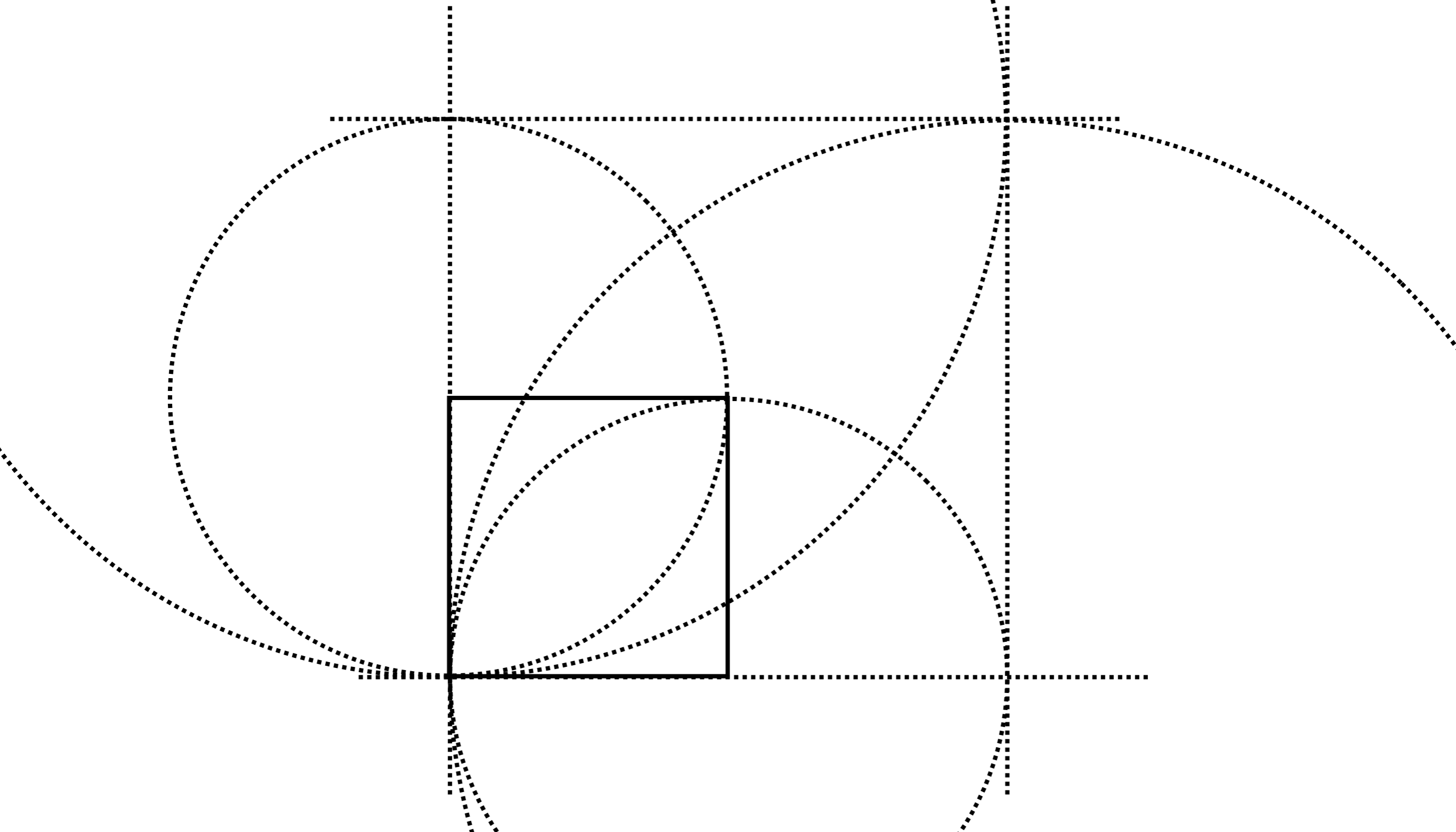
Quadruplement du carré



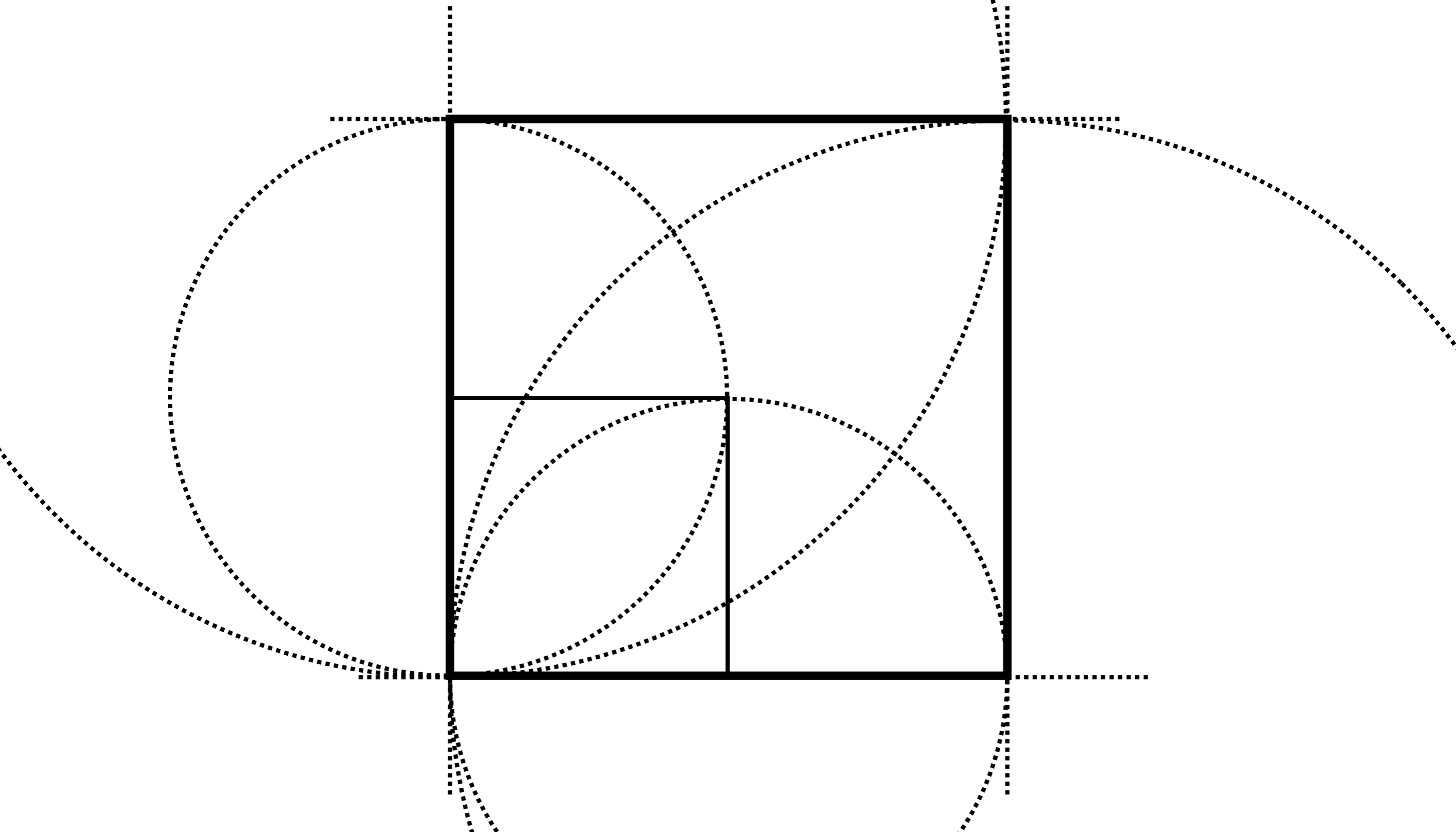
Quadruplement du carré



Quadruplement du carré



Quadruplement du carré



Effacité 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) utilisée 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 $n \rightarrow t(n)$ opérations (exemple : $t(n) = n^2$ ou $t(n) = n + 5$)
- Est-ce qu'on peut faire la même chose en moins de $t(n)$ 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

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

approximativement
3 opérations

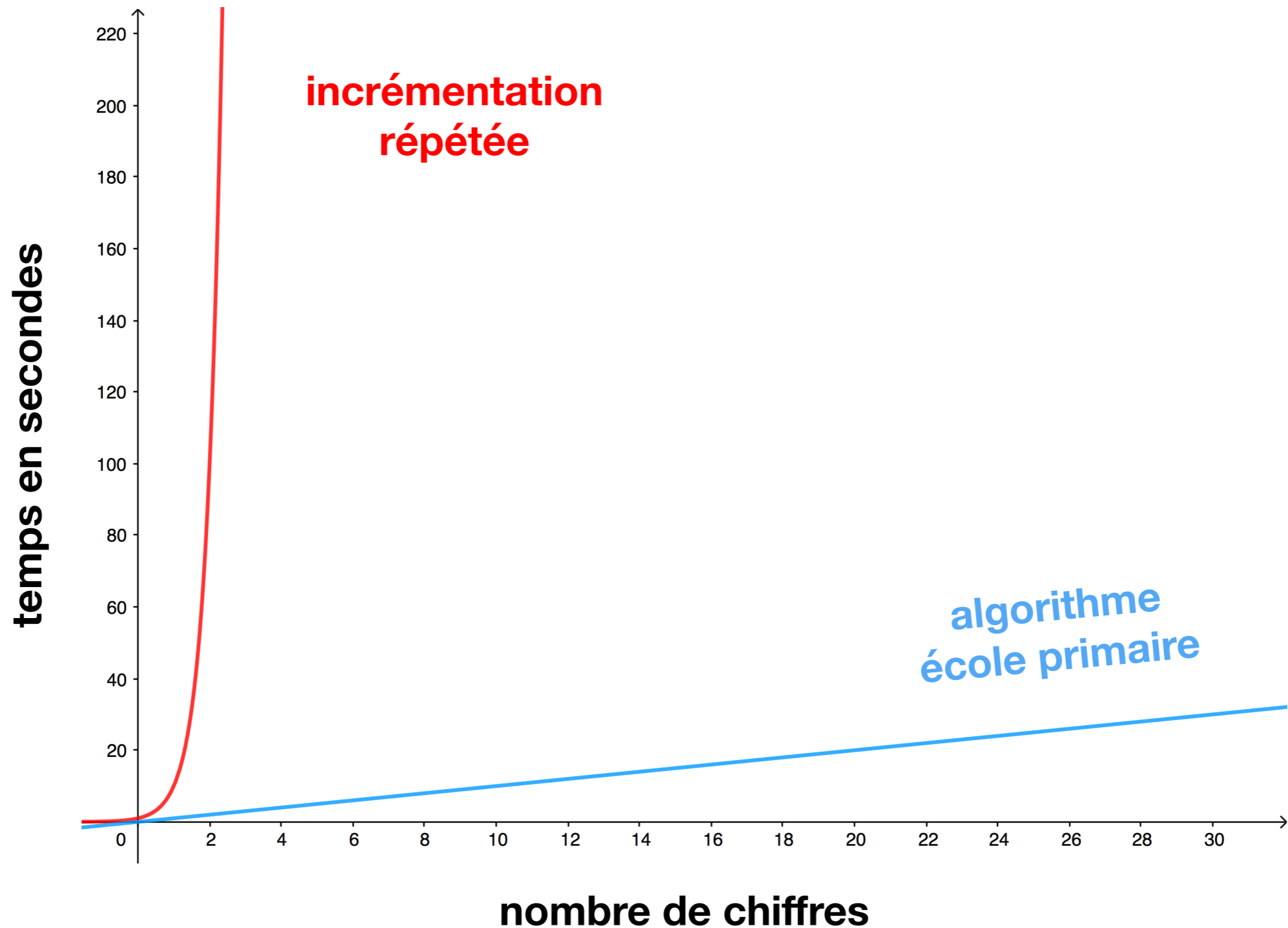
234
235
236
237
...
513
514
515

au moins
281 opérations

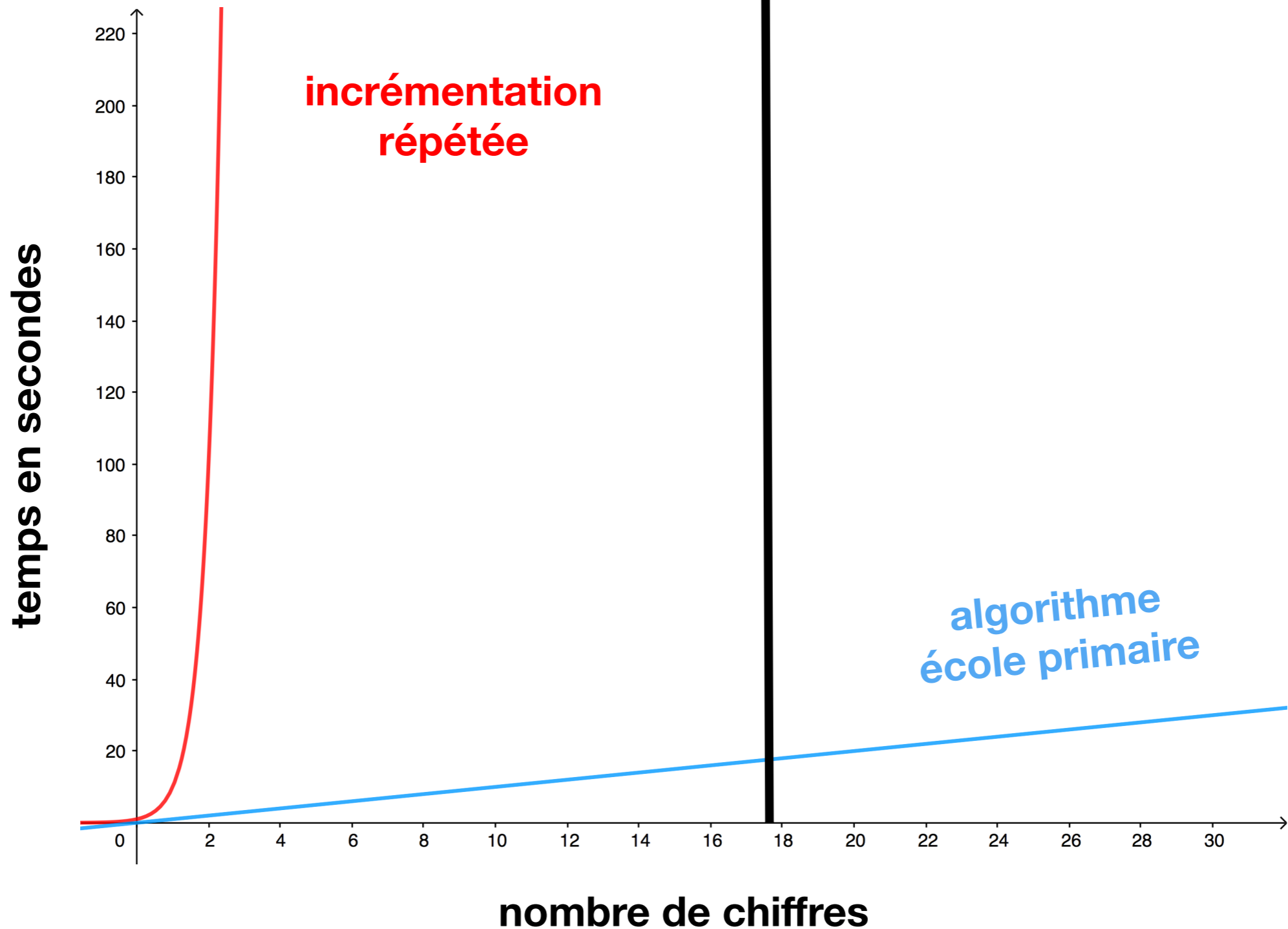
Efficacité des algorithmes pour l'addition

- 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

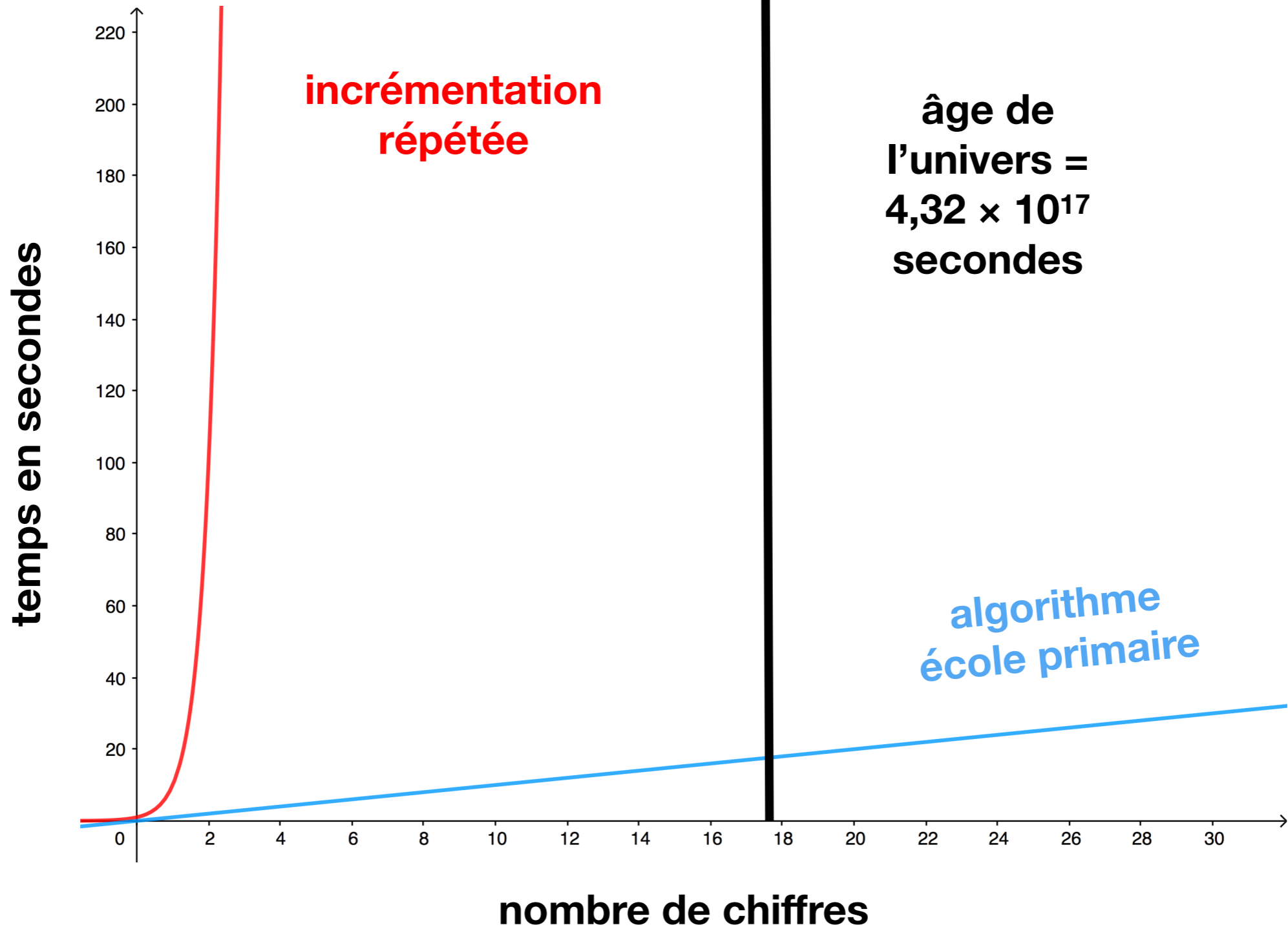
En termes de secondes



En termes de secondes



En termes de secondes

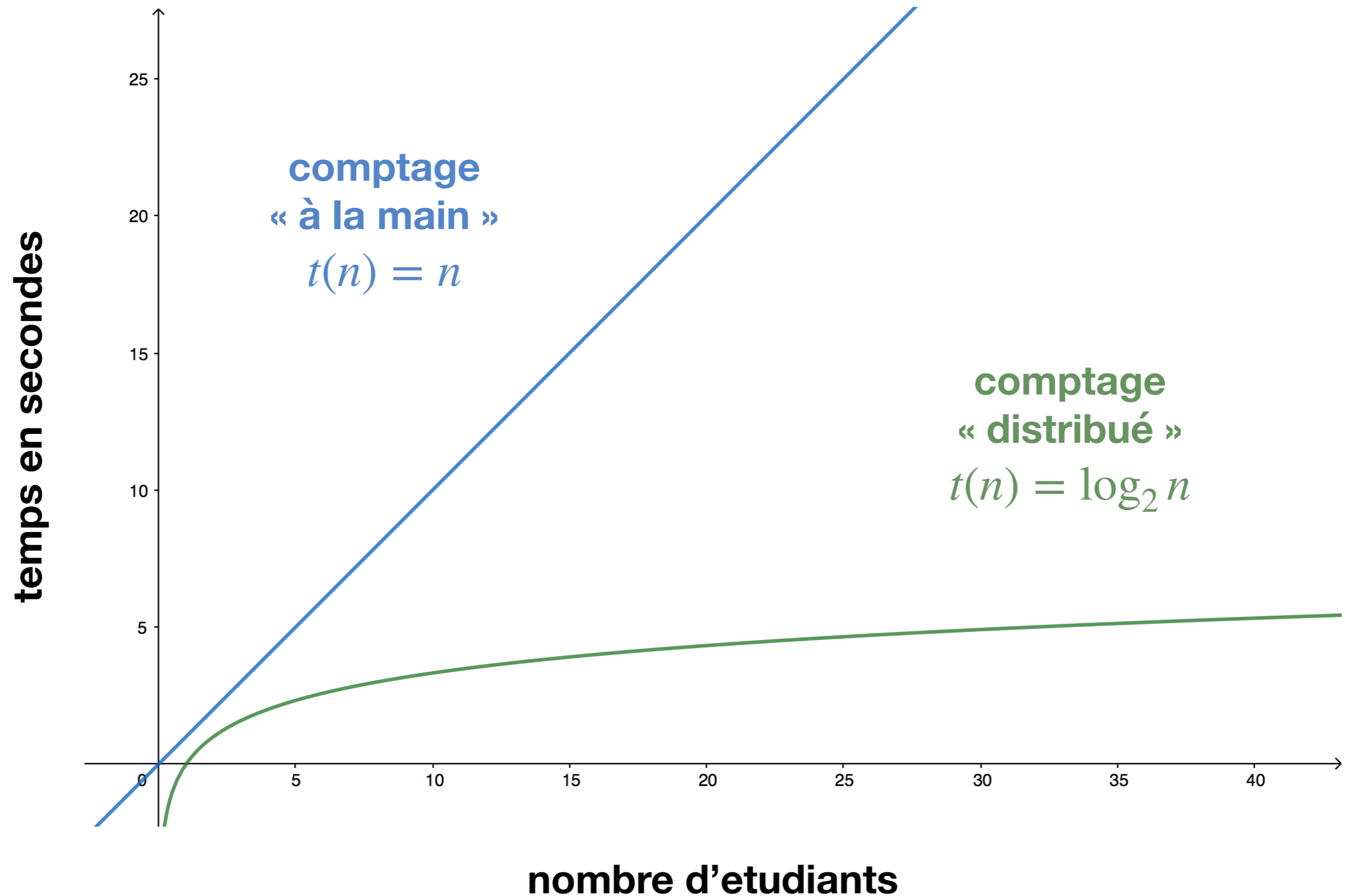


**Combien d'étudiants
y a-t-il dans la salle ?**

Combien d'étudiants y a-t-il dans la salle ?

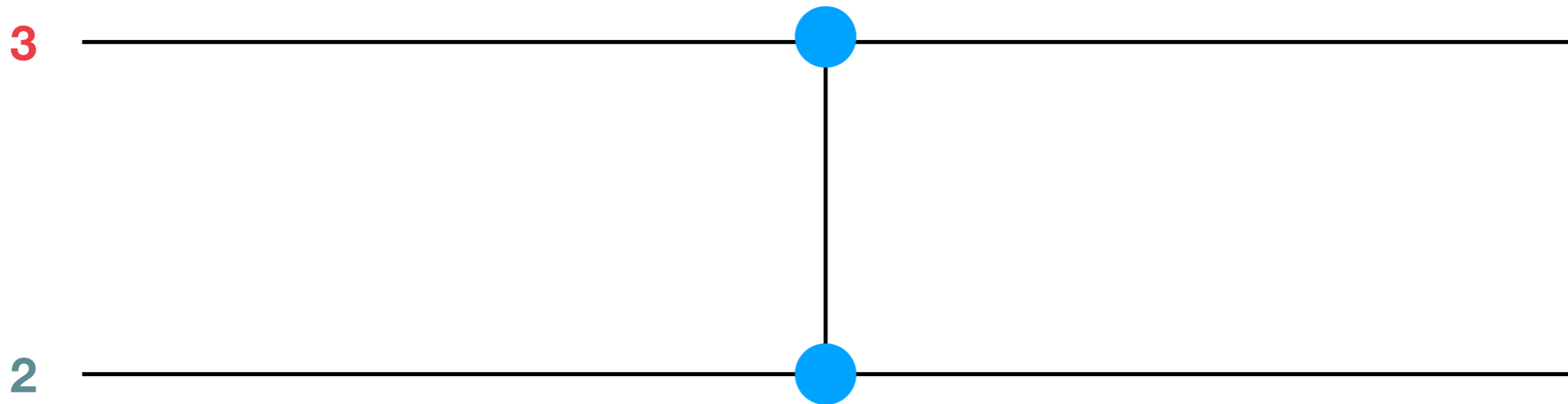
- Chaque étudiant commence avec le nombre 1 en tête
- **Tant qu'il** reste au moins deux étudiants debout :
 - Chaque étudiant encore debout cherche du regard un autre étudiant debout
 - Les deux étudiants s'échangent le nombre qu'ils ont en tête
 - L'un des deux étudiants s'assoit
 - L'autre additionne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

Efficacité du comptage

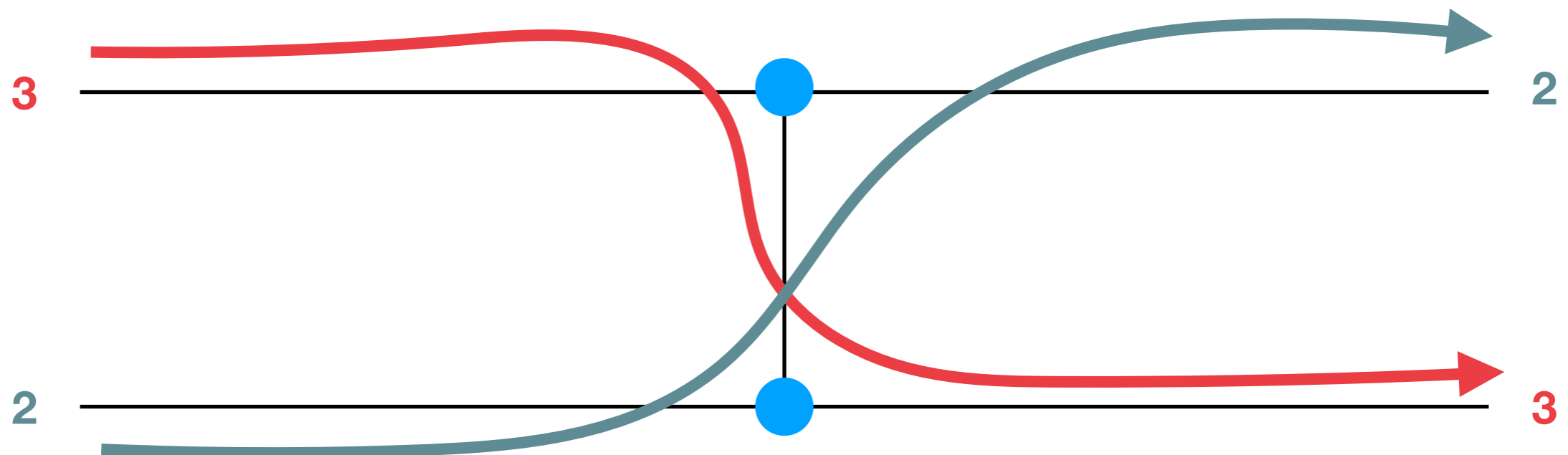


Une formalisation des algorithmes de tri : les réseaux de tri

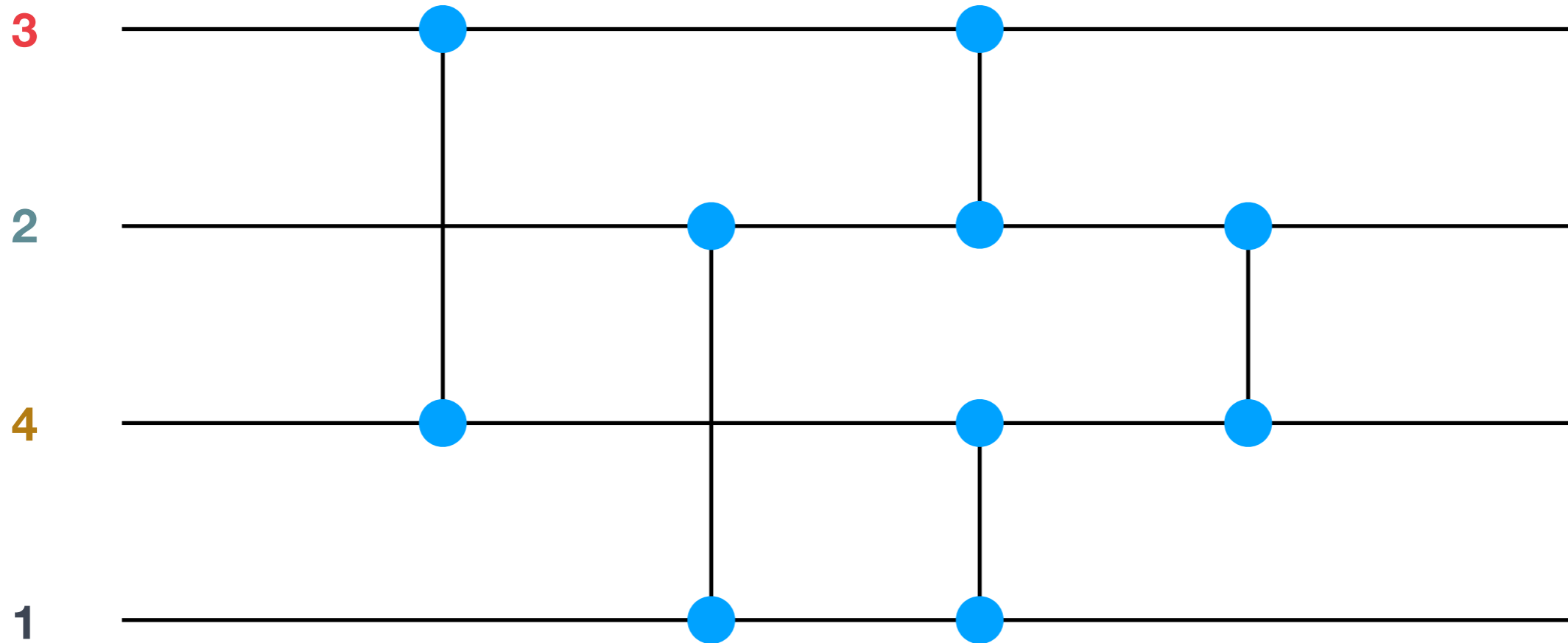
Réseaux de tri



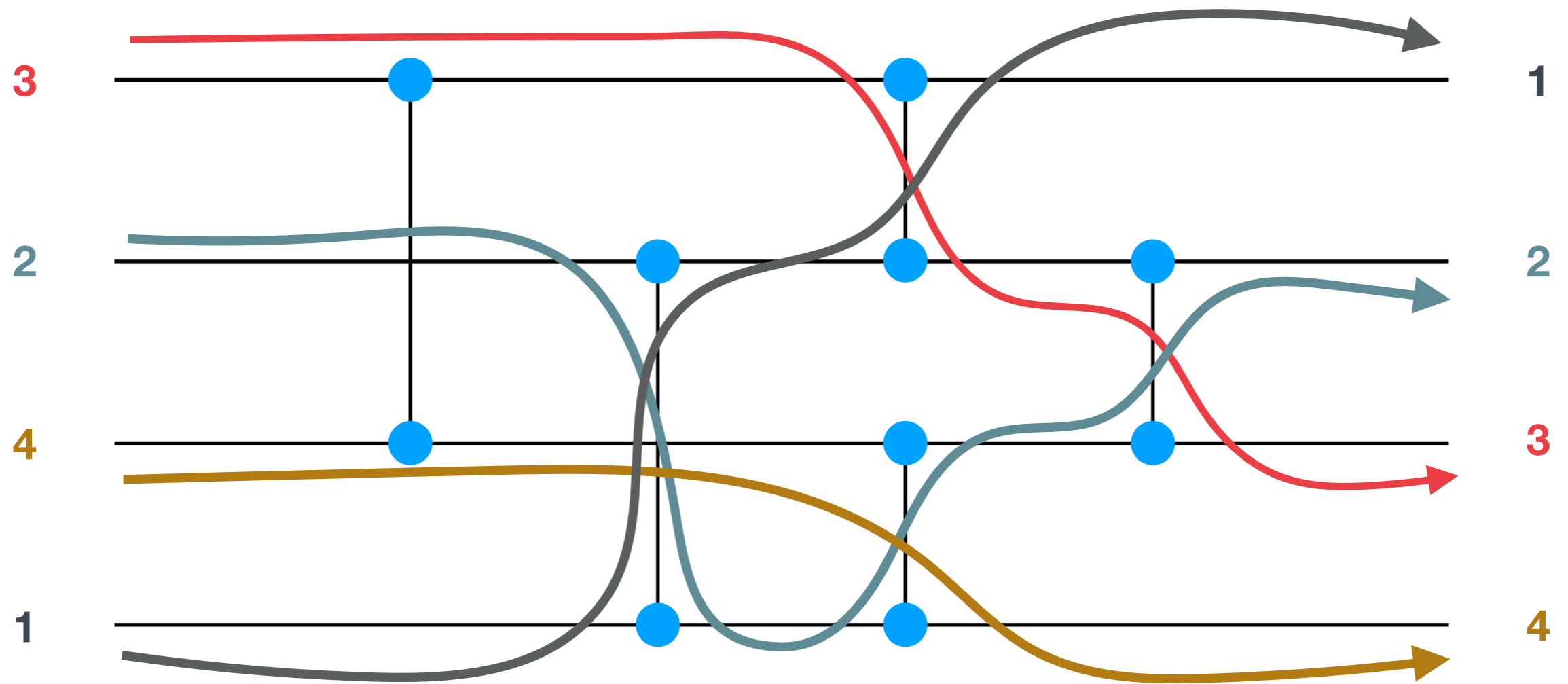
Réseaux de tri



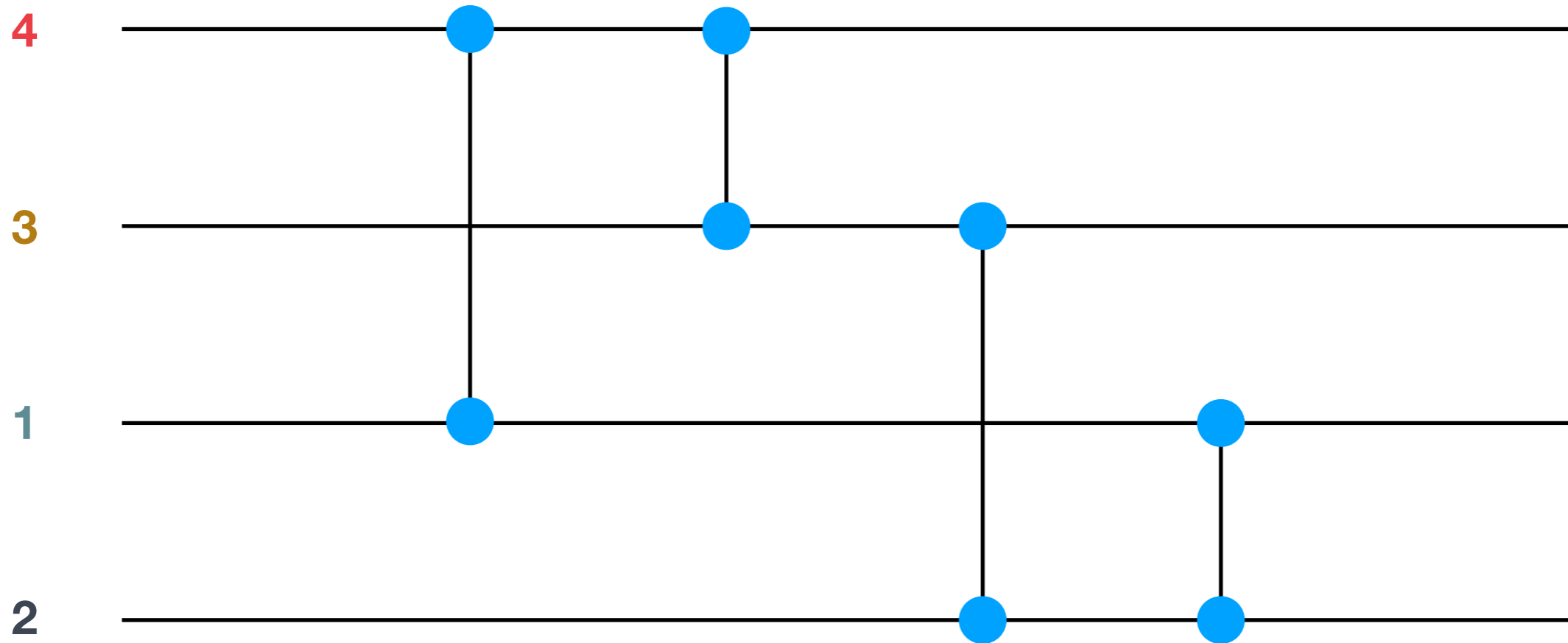
Un exemple



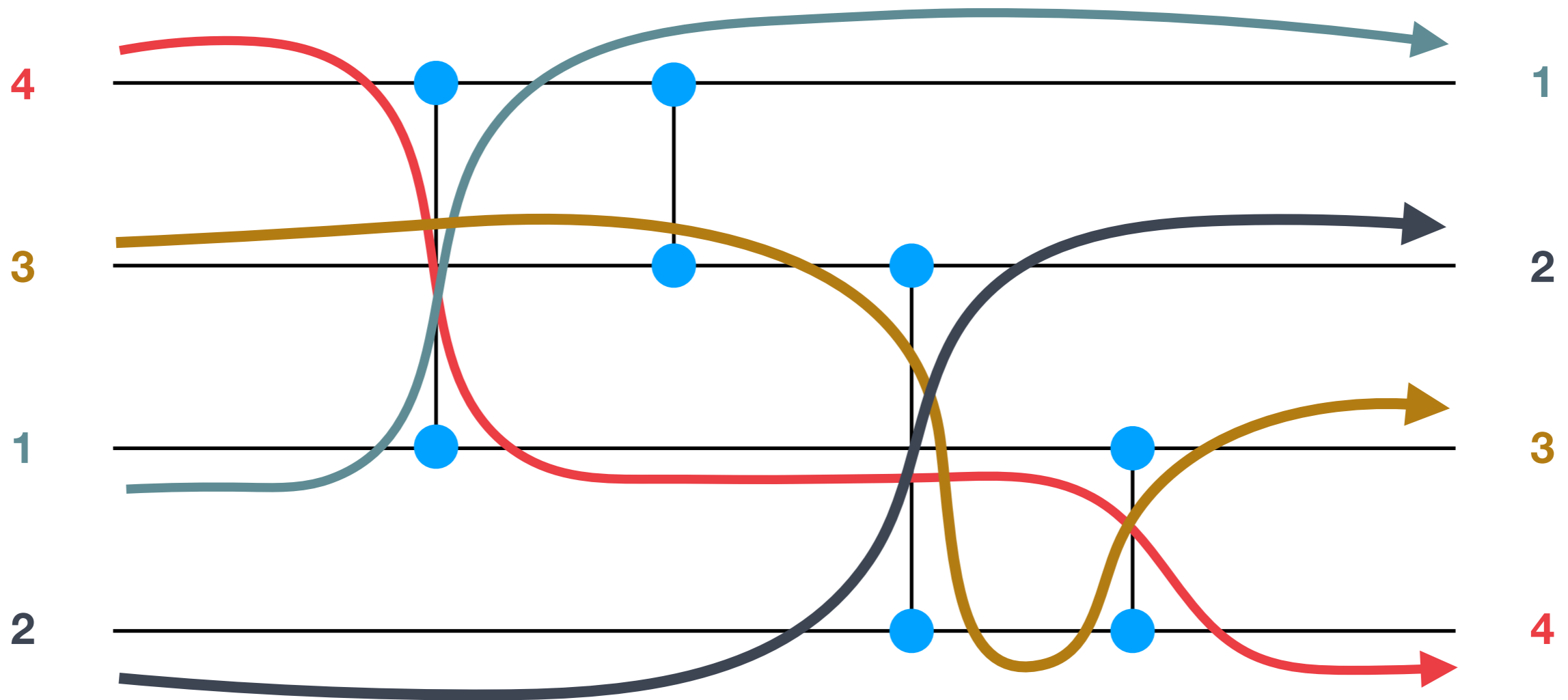
Un exemple



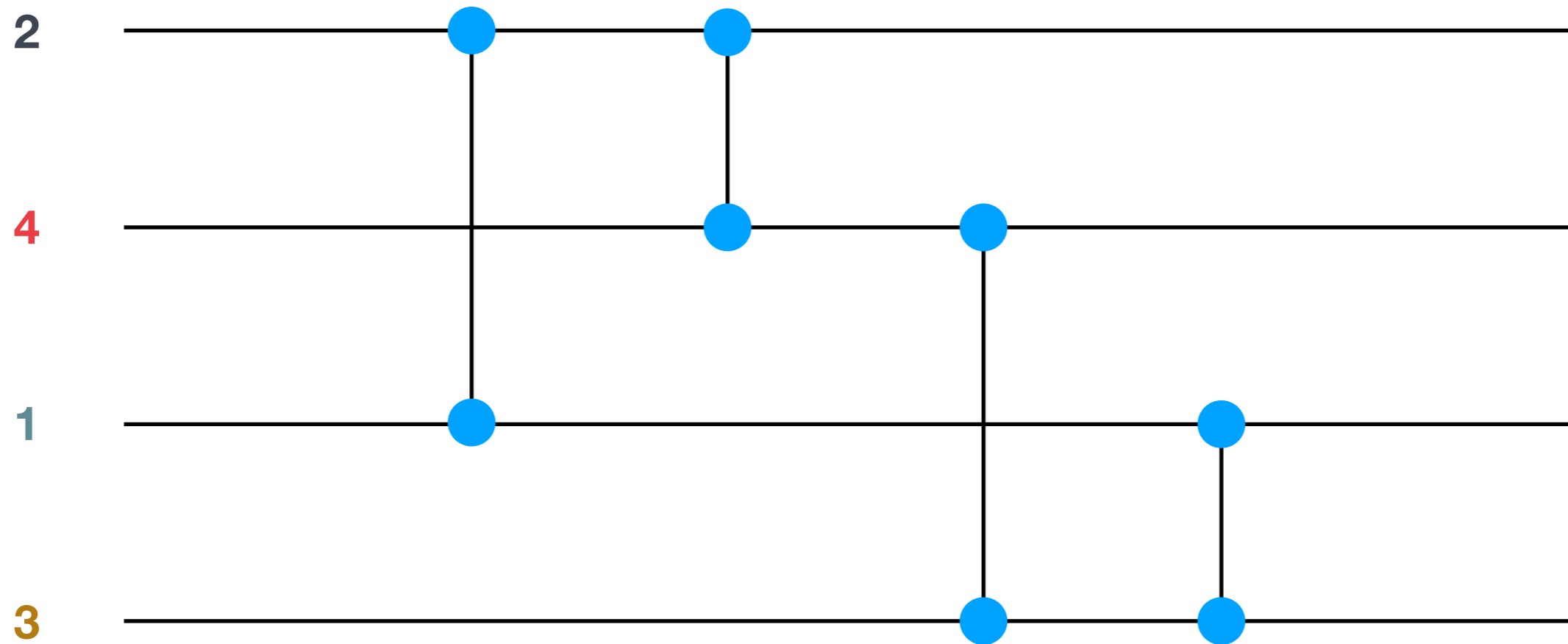
Un autre réseau



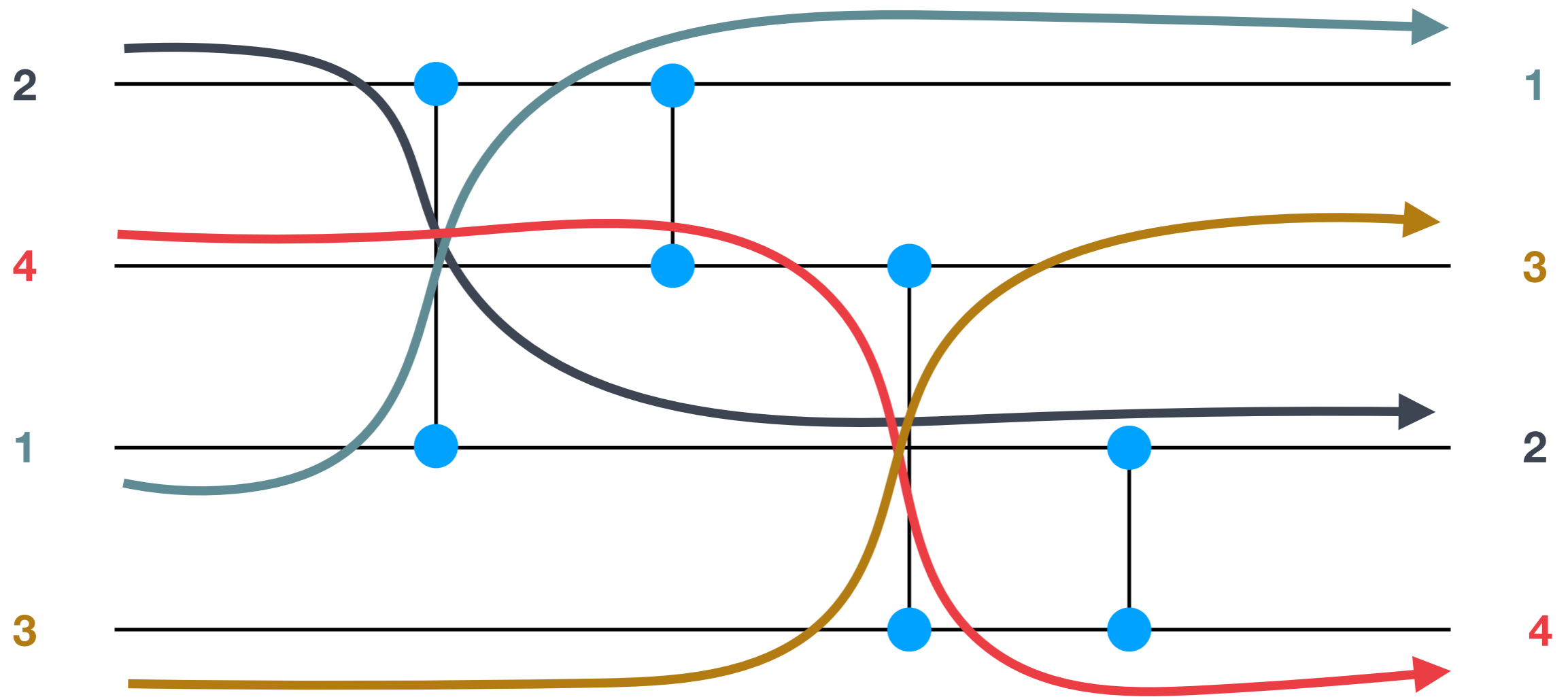
Un autre réseau



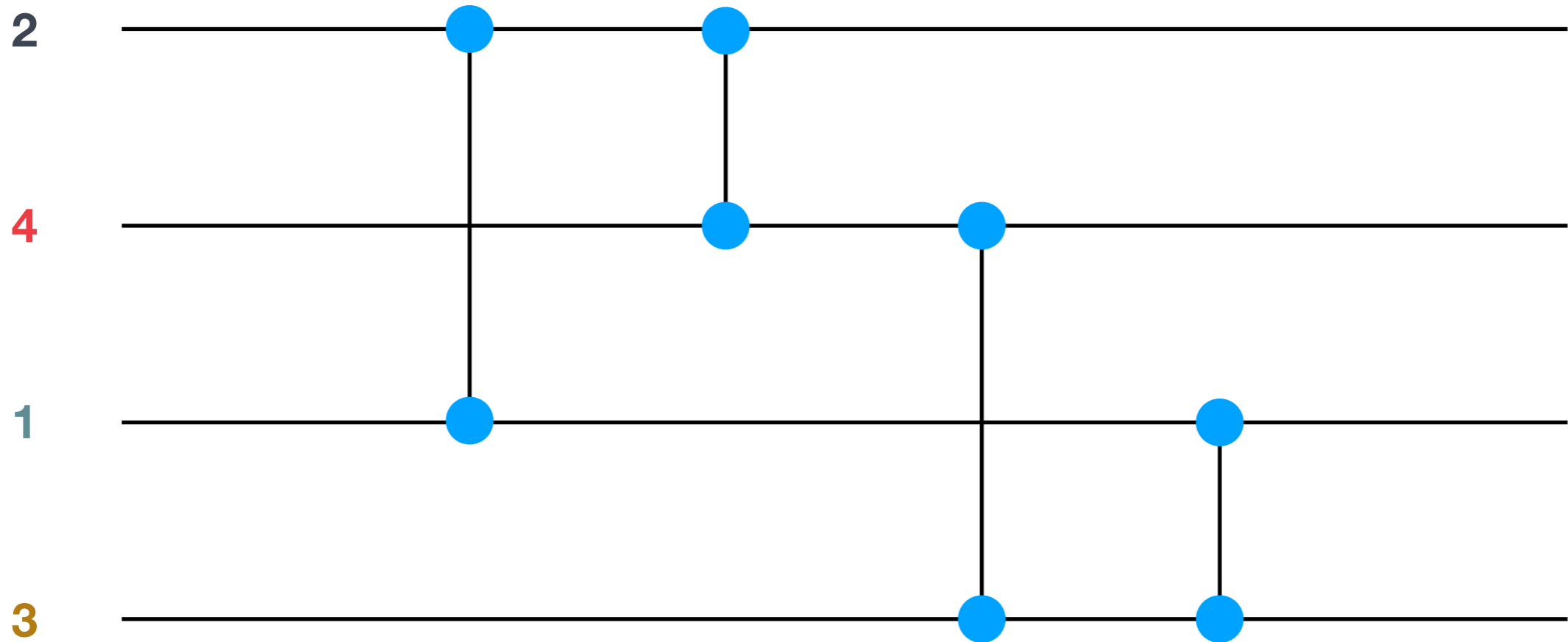
Avec d'autres données



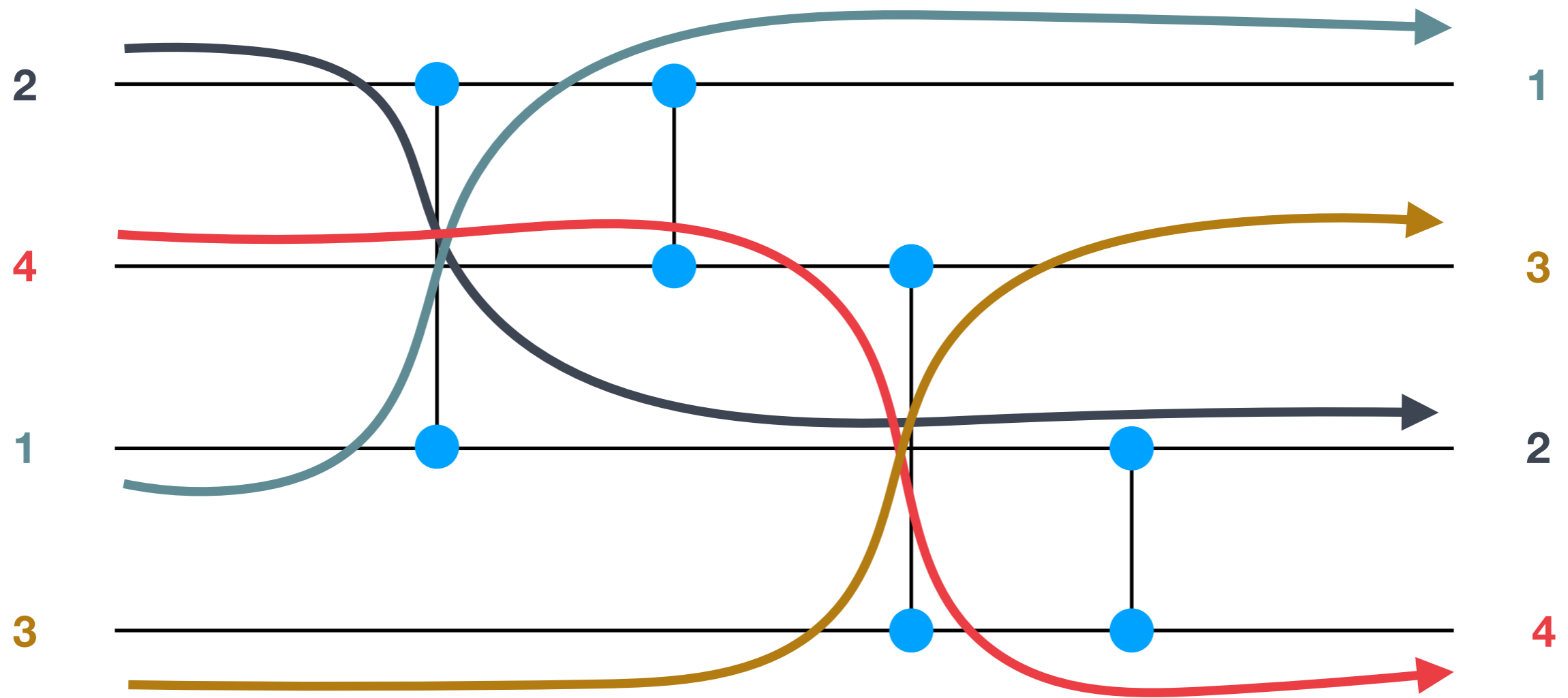
Avec d'autres données



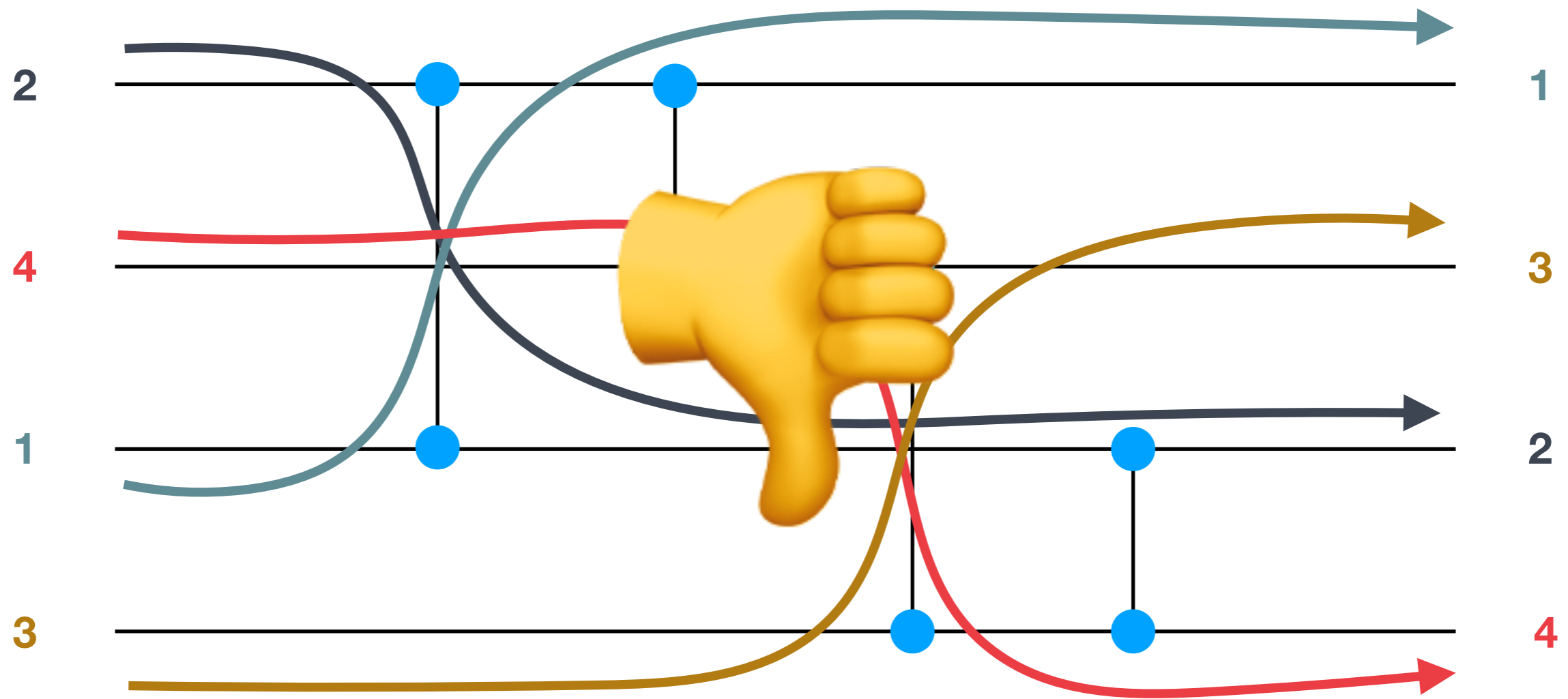
Pas un réseau de tri !



Pas un réseau de tri !



Pas un réseau de tri !



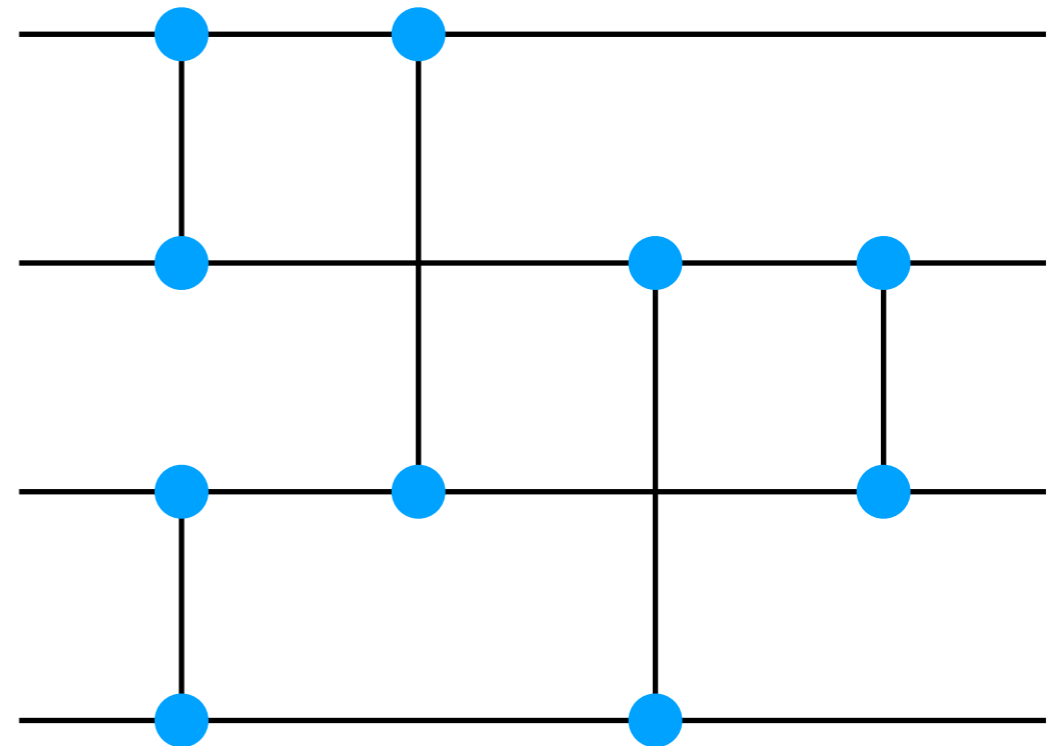
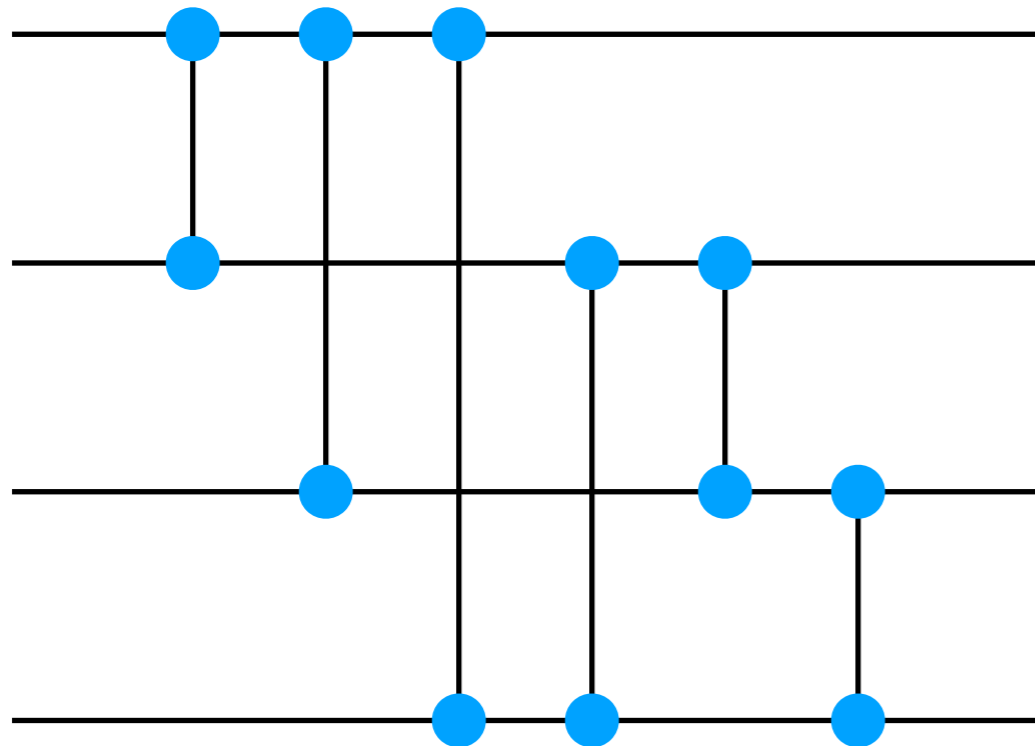
Vérifier si un réseau est de tri

- Essayer avec toutes les entrées de n entiers naturels : mais il y en a infinies !
- Peu important les valeurs, seul l'ordre compte : tester avec toutes les permutations de $1, \dots, n$
- 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 $n!$ et 10^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 ambiguë
- On prouve qu'il est correct
- On vérifie qu'il est efficace (idéalement, on choisit l'algorithme optimal)
- On le met en œuvre (pas dans cette UE)
- On le teste (pas dans cette UE)