Introduction à l’informatique CM1

Antonio E. Porreca
aeporreca.org/introinfo

Équipe enseignante

- Responsable de l'UE : c'est moi
  - Antonio E. Porreca 👉 antonio.porreca@univ-amu.fr
  - L'accent drôle est italien 🤌
- Chargés de TD :
  - Antonio E. Porreca, c'est toujours moi
  - Marius Rolland (sauf en septembre)

La page web :

aeporreca.org/introinfo

Deux UE complémentaires

- Introduction à l’informatique
    - Découvrir l’informatique au travers d’exemples et de mises en situation théoriques et pratiques, en mode « débranché »
- Mise en œuvre informatique
    - Développer des compétences pratiques en algorithmique et programmation, sur ordinateur en langage Python

On va apprendre 😊

- Concevoir le traitement informatisé d’informations de différentes natures
- Modéliser un problème concret sous la forme d’un problème algorithmique connu
- Évaluer l’efficacité et la correction d’une solution algorithmique
- Être familiarisé avec les concepts fondamentaux de complexité et de calculabilité

On ne va pas apprendre ☹️

- Utiliser des logiciels bureautiques
- Installer Linux sur son ordinateur
- Faire du web design
- Jouer aux ou réaliser des jeux vidéos
- Sécurité informatique

Format de l’UE

- 15h de cours magistral (CM) = 10 séances de 1h30
- 15h de travaux dirigés (TD) = 10 séances de 1h30
- Travail personnel
  - prise de notes pendant les cours
  - travail entre les séances pour préparer/terminer les TD
  - lecture des notes de cours
- Tutorat

Evaluation

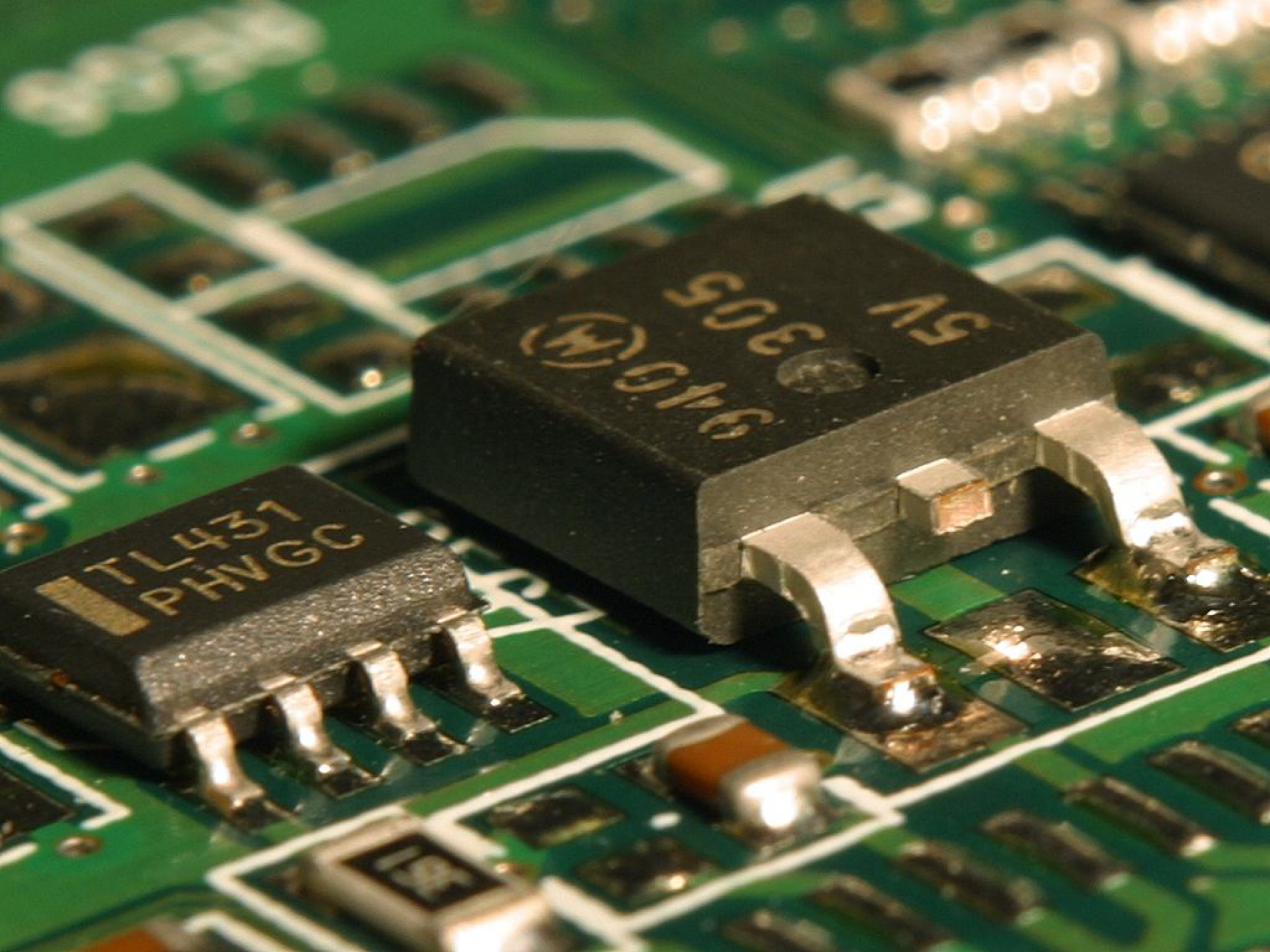
- Évaluation en 1ère session :

$$\max(0{,}3 \times \text{partiel} + 0{,}7 \times \text{ex terminal}, \text{ex terminal})$$

- Partiel le 27 octobre, 1h30, sans documents ni calculatrice
- Examen terminal de 2h, sans document ni calculatrice
- Évaluation en 2ème session (rattrapage, en juin) :

$$1{,}0 \times (\text{note examen terminal})$$

C’est quoi l’informatique ?


TL431
PHVGC

C’est aussi de la science ?

- Peut-être…
- On définit des modèles mathématiques du calcul
- On prouve des résultats
- On peut faire des choix techniques

Objets d’étude de l’informatique

- Le stockage de l’information
- La transmission de l’information
- Le traitement de l’information 👉 algorithmes !
- L’information elle-même

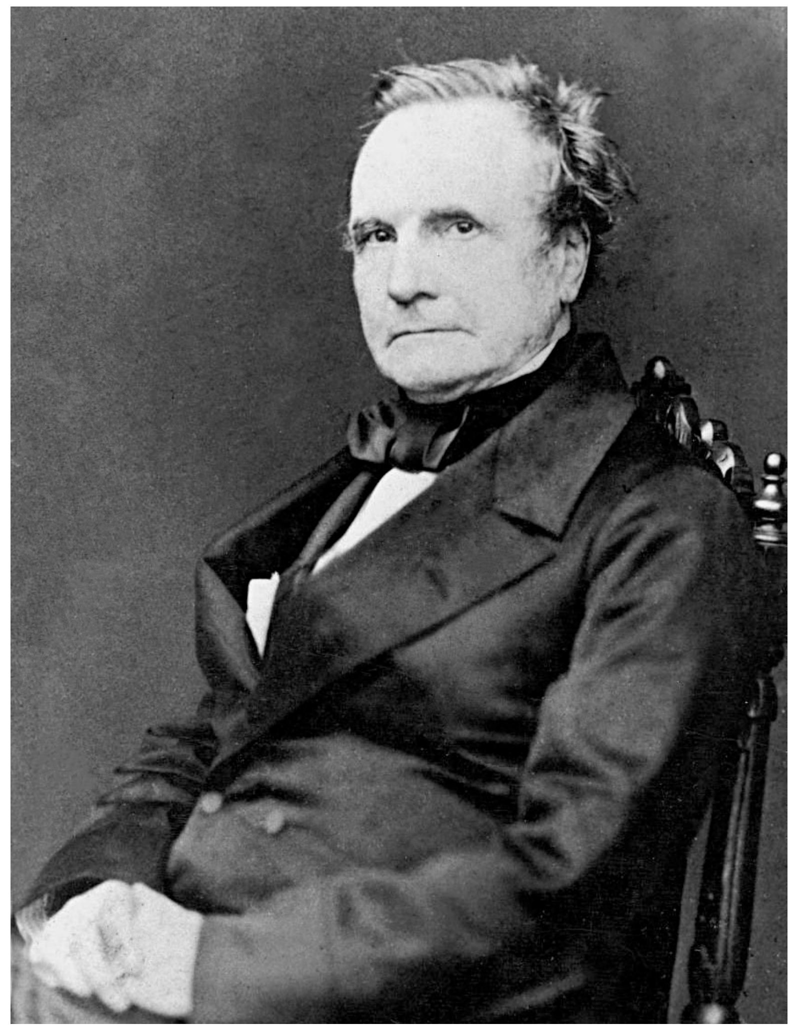
Quelques personnages

Quelques personnages

Muḥammad ibn Mūsā al-Khwārizmī

Quelques personnages

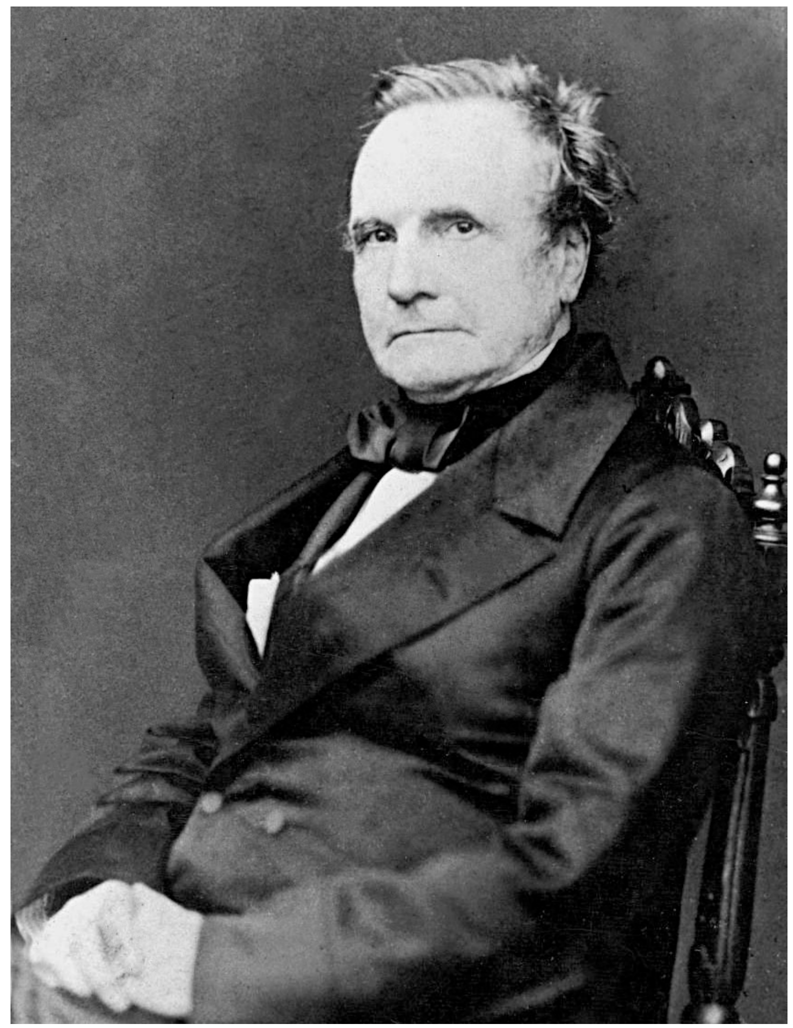
Muḥammad ibn Mūsā
al-Khwārizmī



Charles Babbage

Quelques personnages

Muḥammad ibn Mūsā al-Khwārizmī



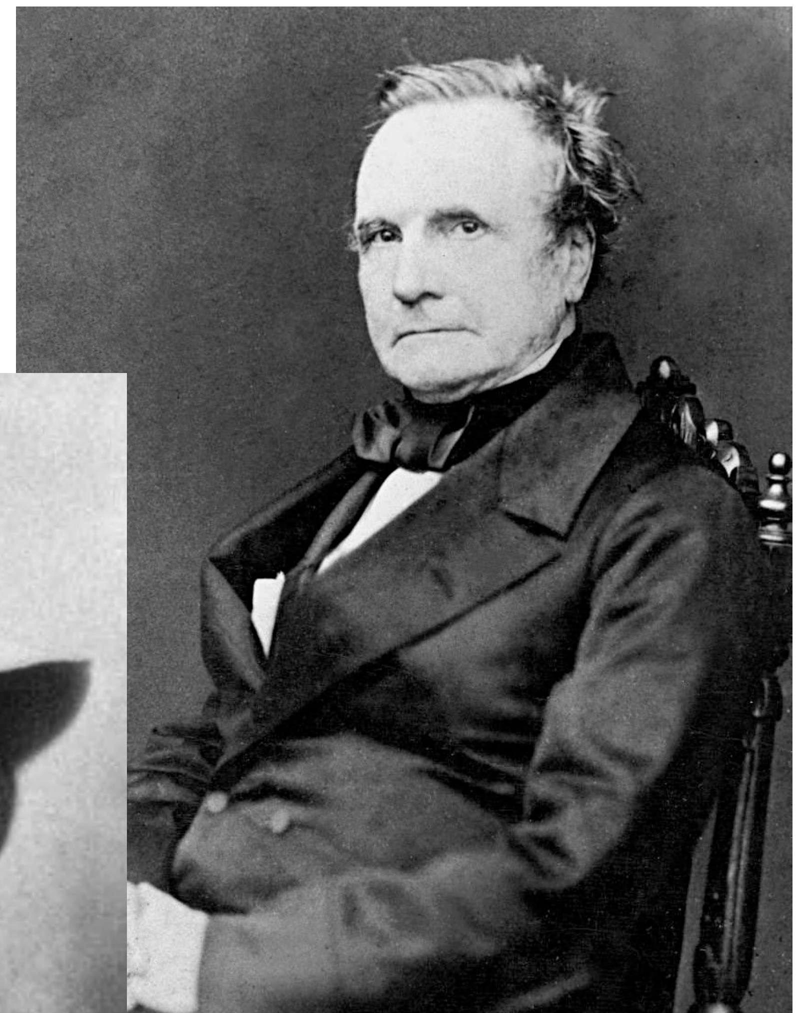
Charles Babbage

Ada Lovelace

Quelques personnages

Muḥ

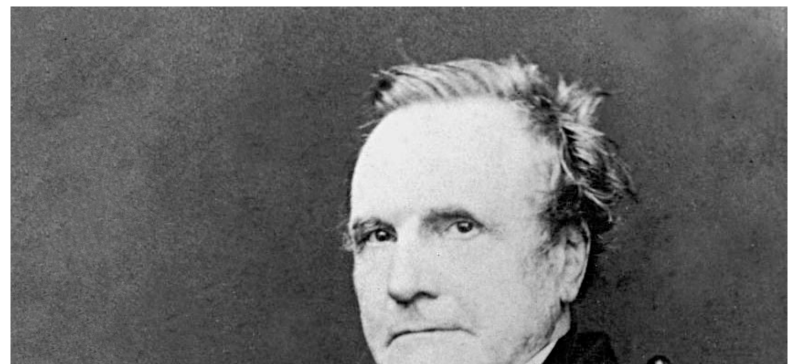
David Hilbert



Charles Babbage

Ada Lovelace

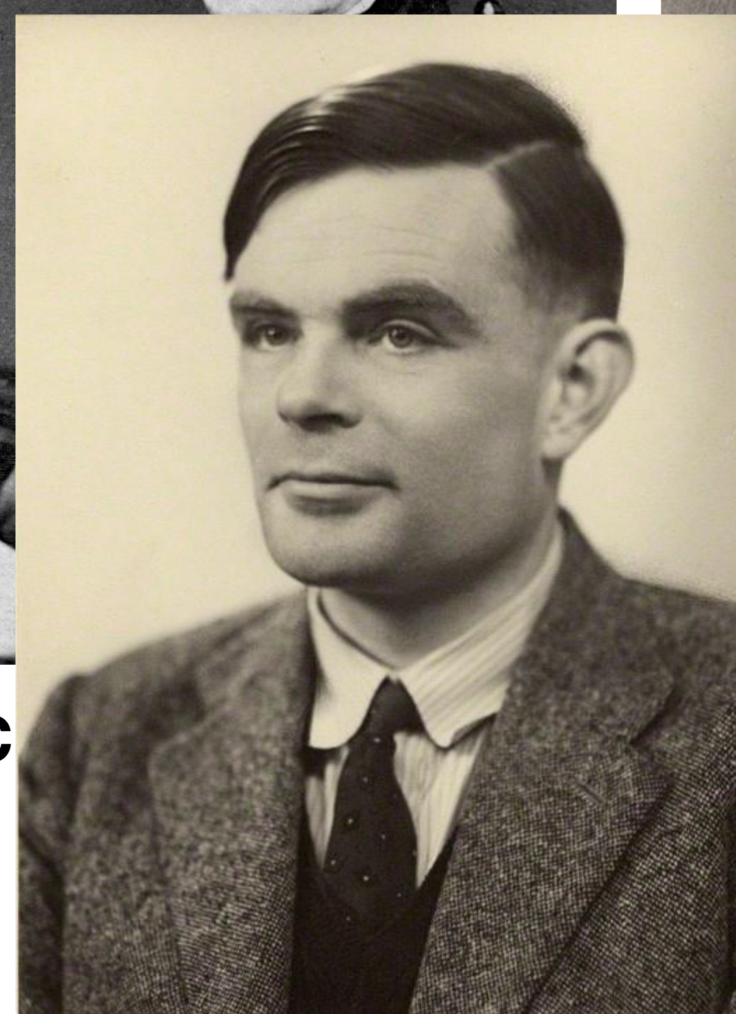
Quelques personnages



Muḥ

C

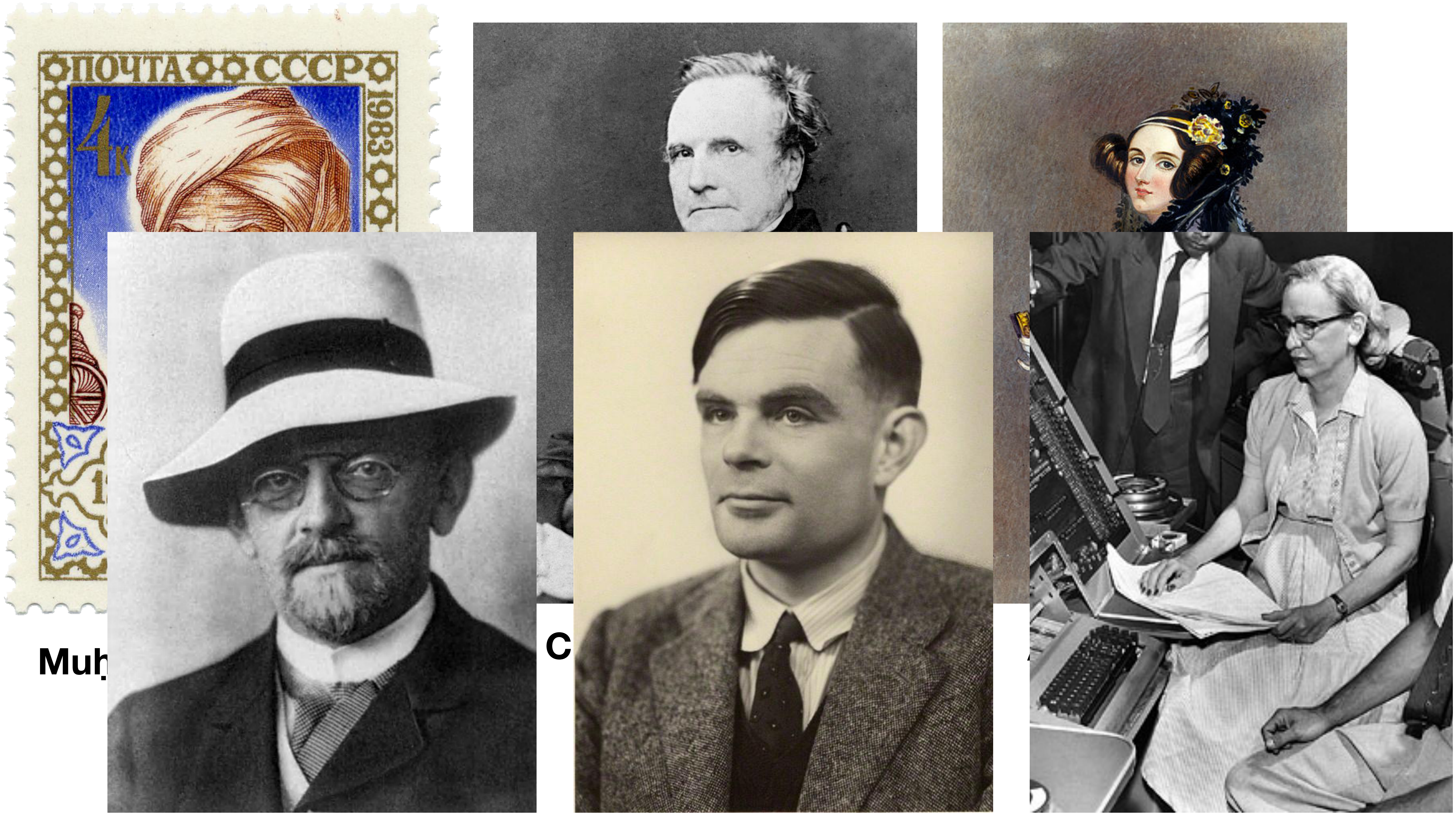
Ada Lovelace



David Hilbert

Alan M. Turing

Quelques personnages



Muḥ

C

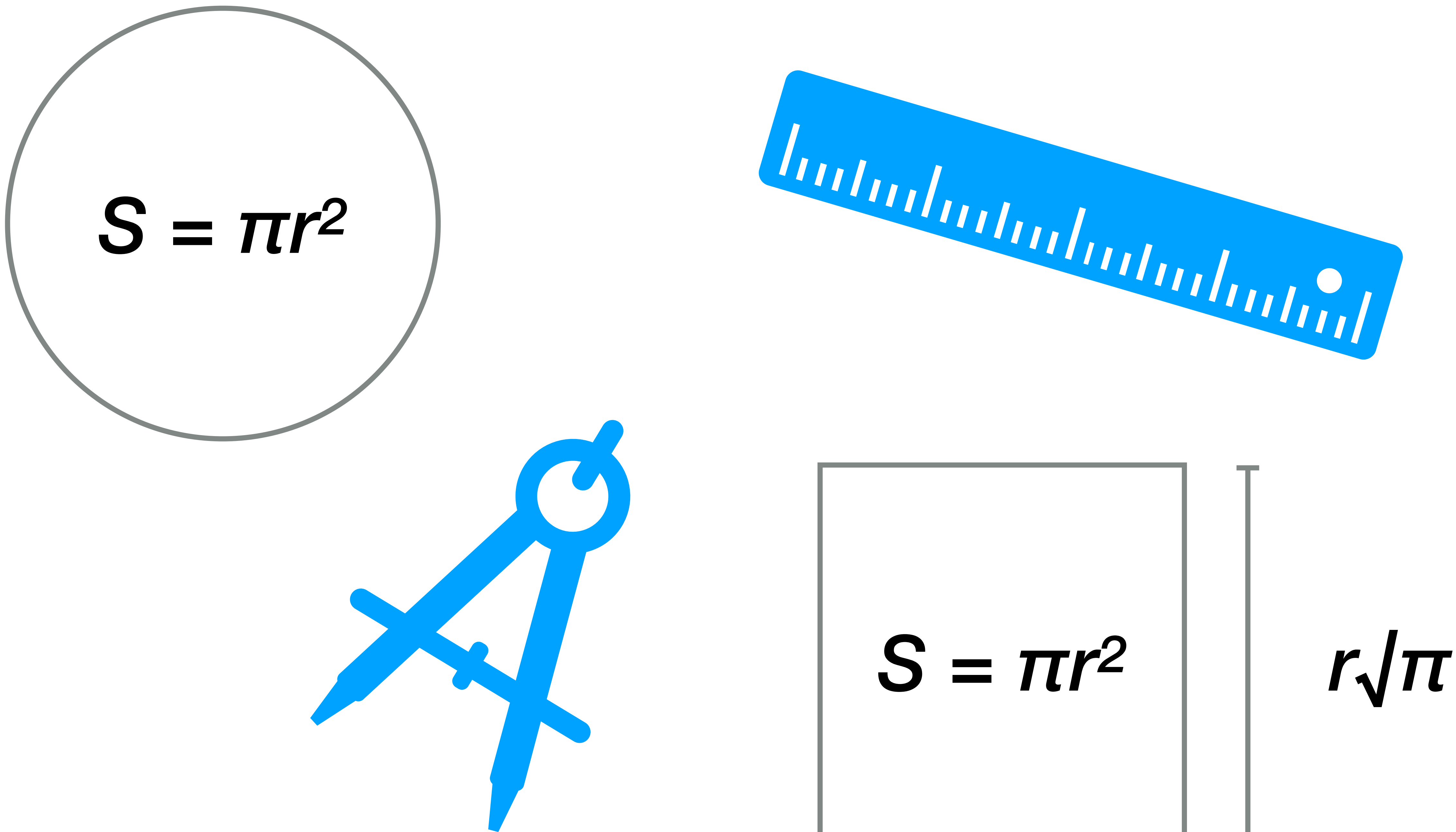
David Hilbert

Alan M. Turing

Grace Hopper

Algorithmes !

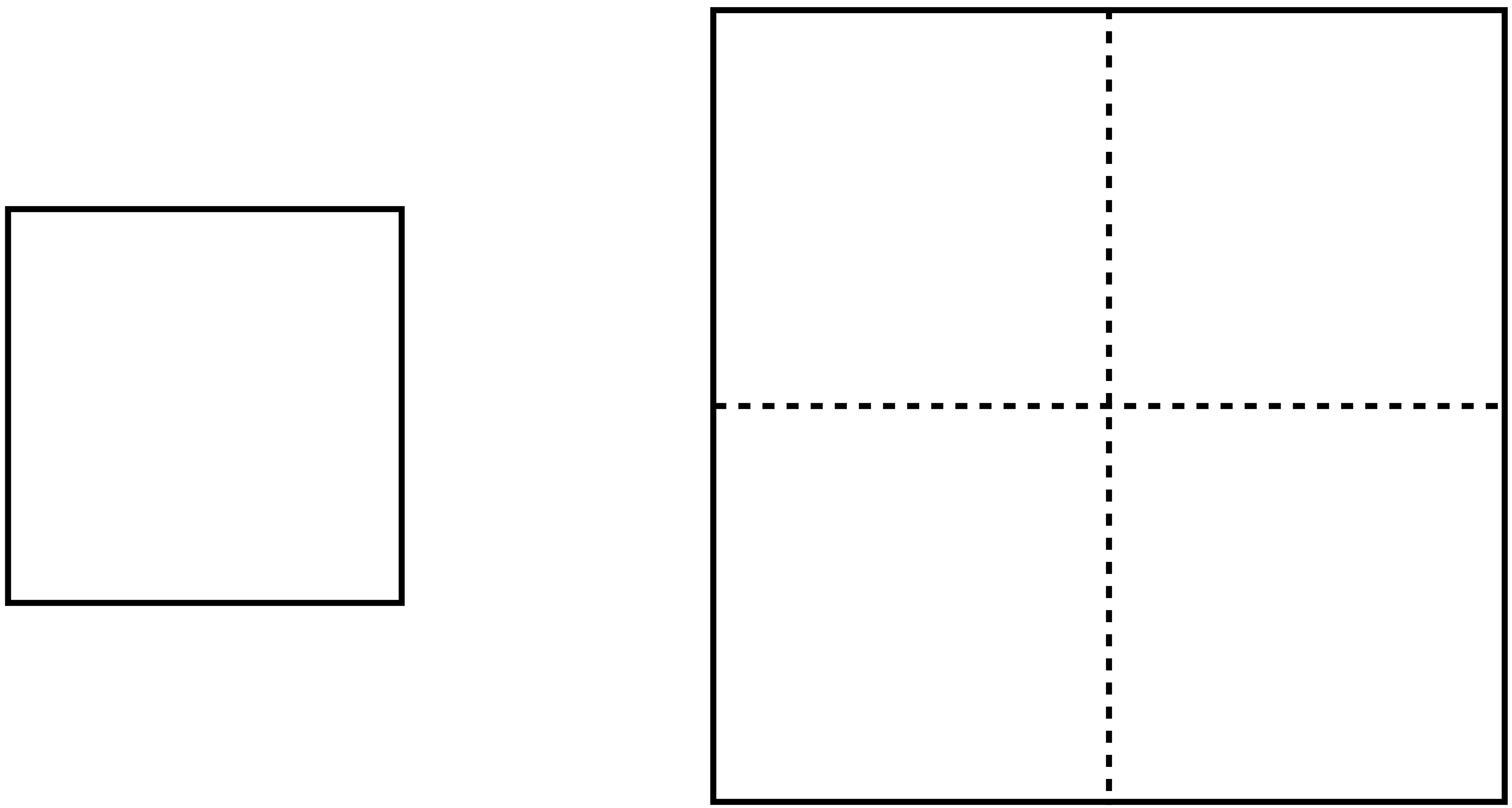
Quadrature du cercle



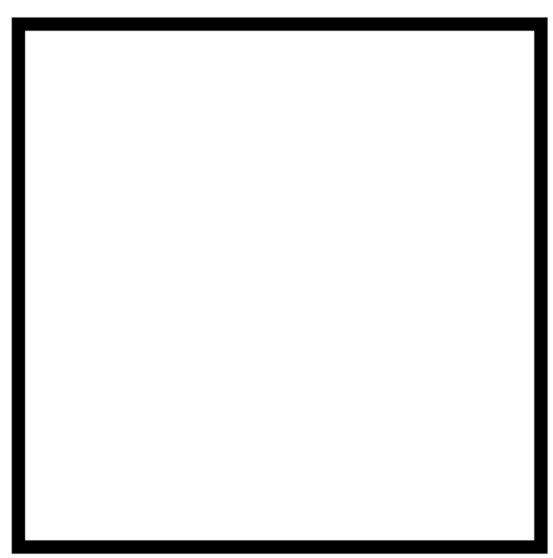
THÉORÈME :
La quadrature du cercle à la règle
et au compas est impossible

–Ferdinand von Lindenmann, 1882

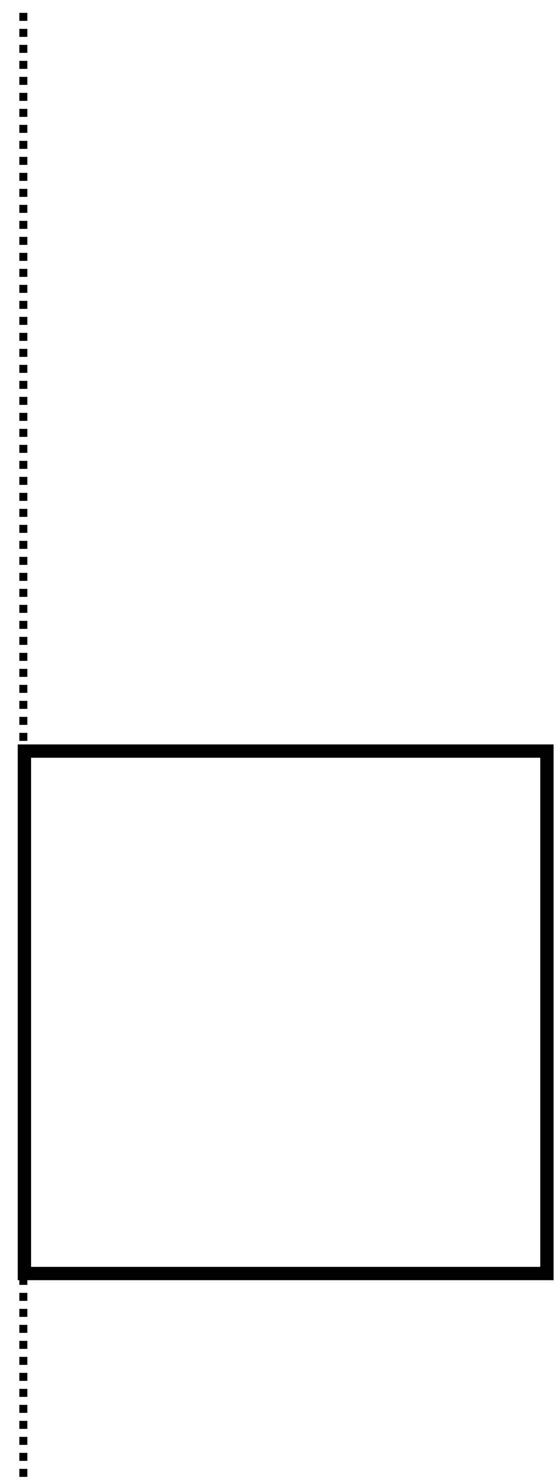
Quadruplement du carré



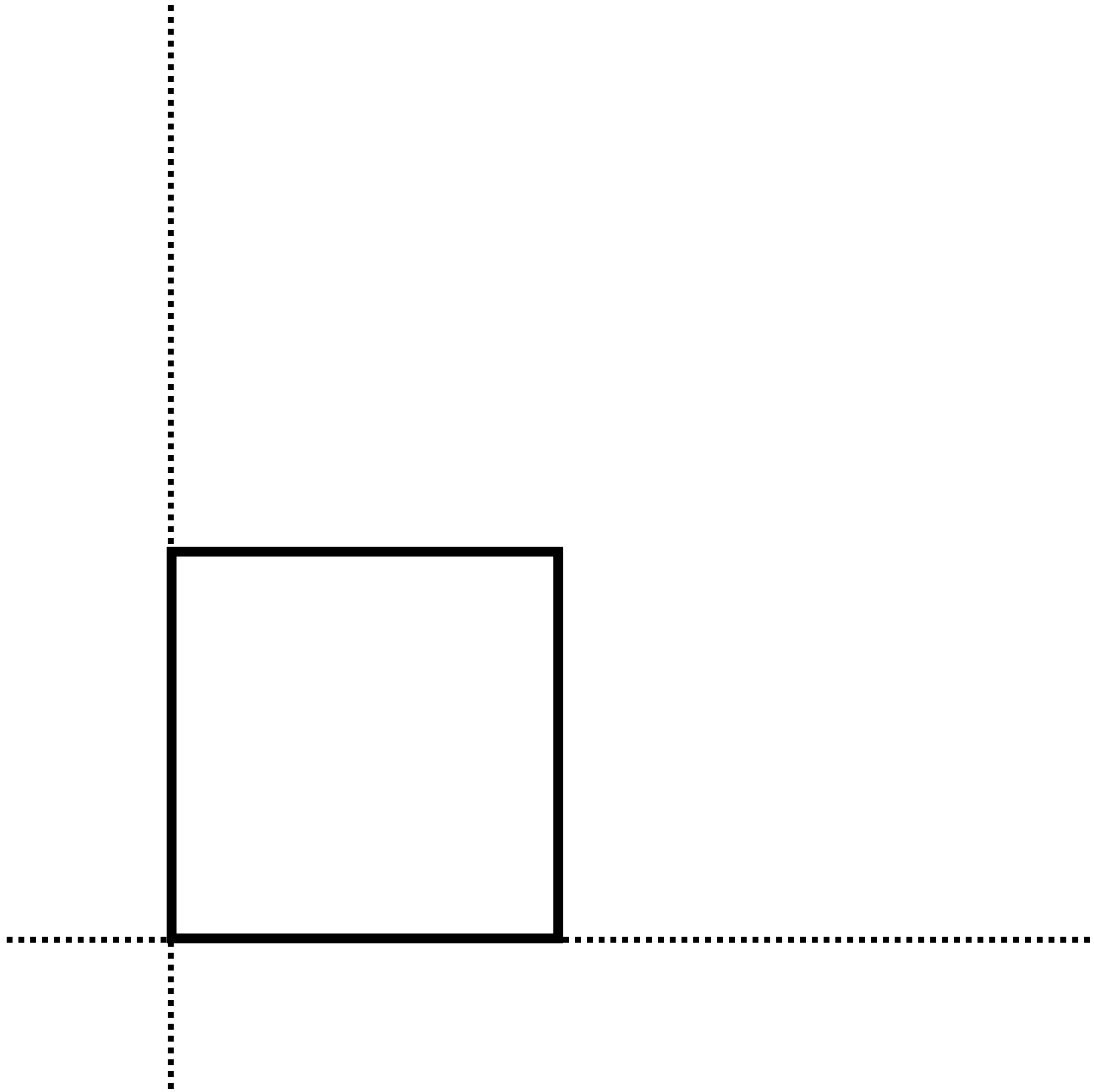
Quadruplement du carré



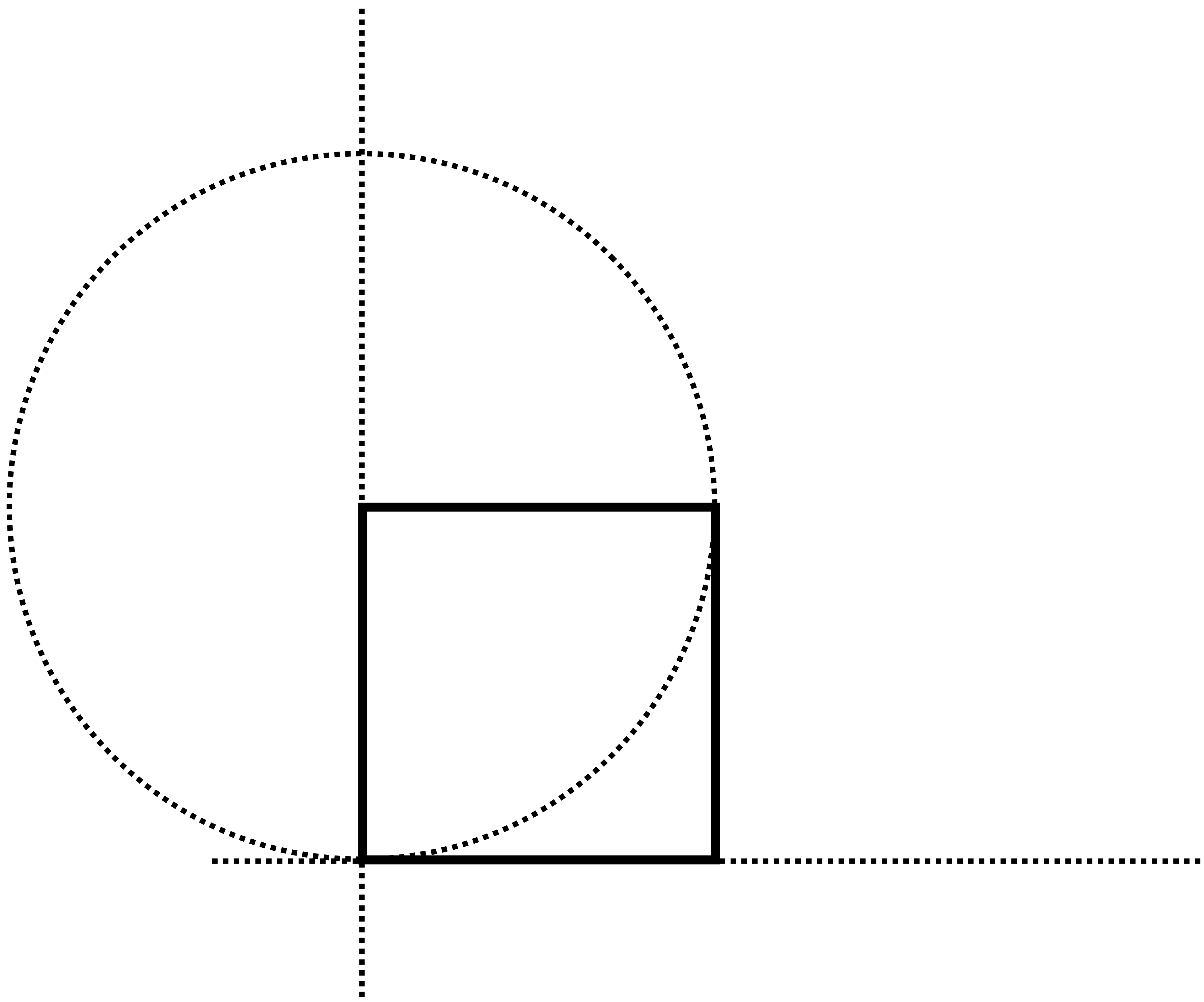
Quadruplement du carré



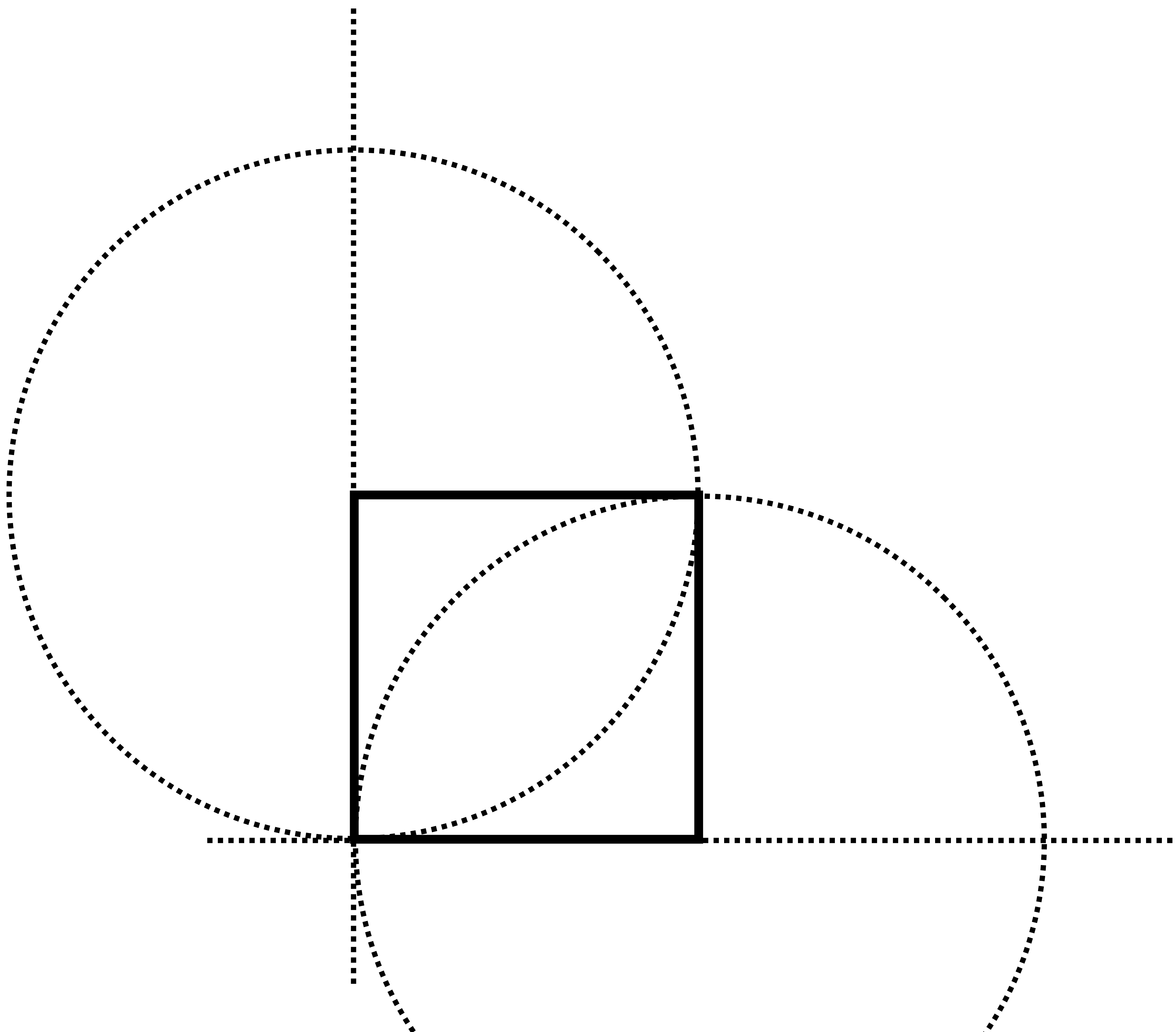
Quadruplement du carré



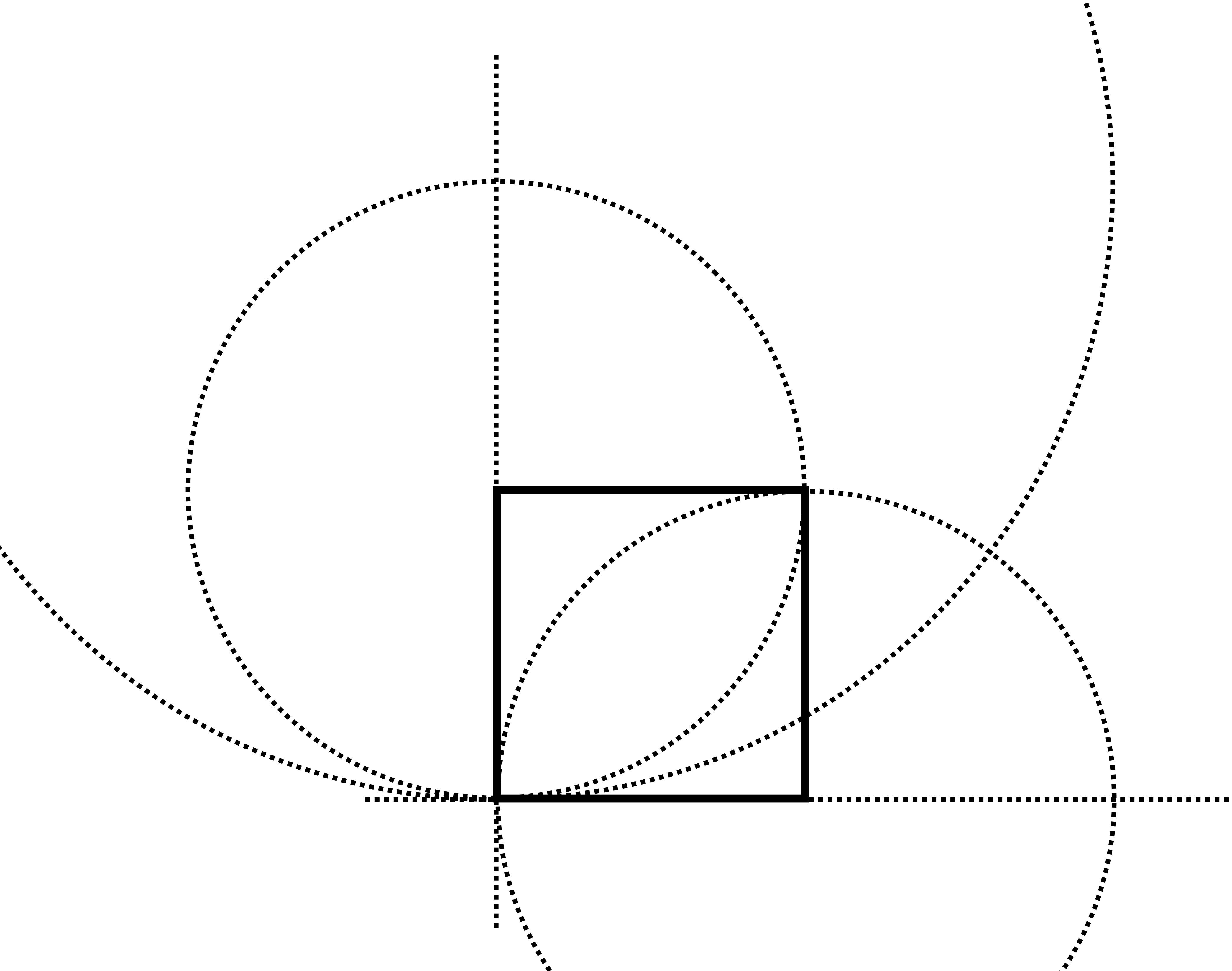
Quadruplement du carré



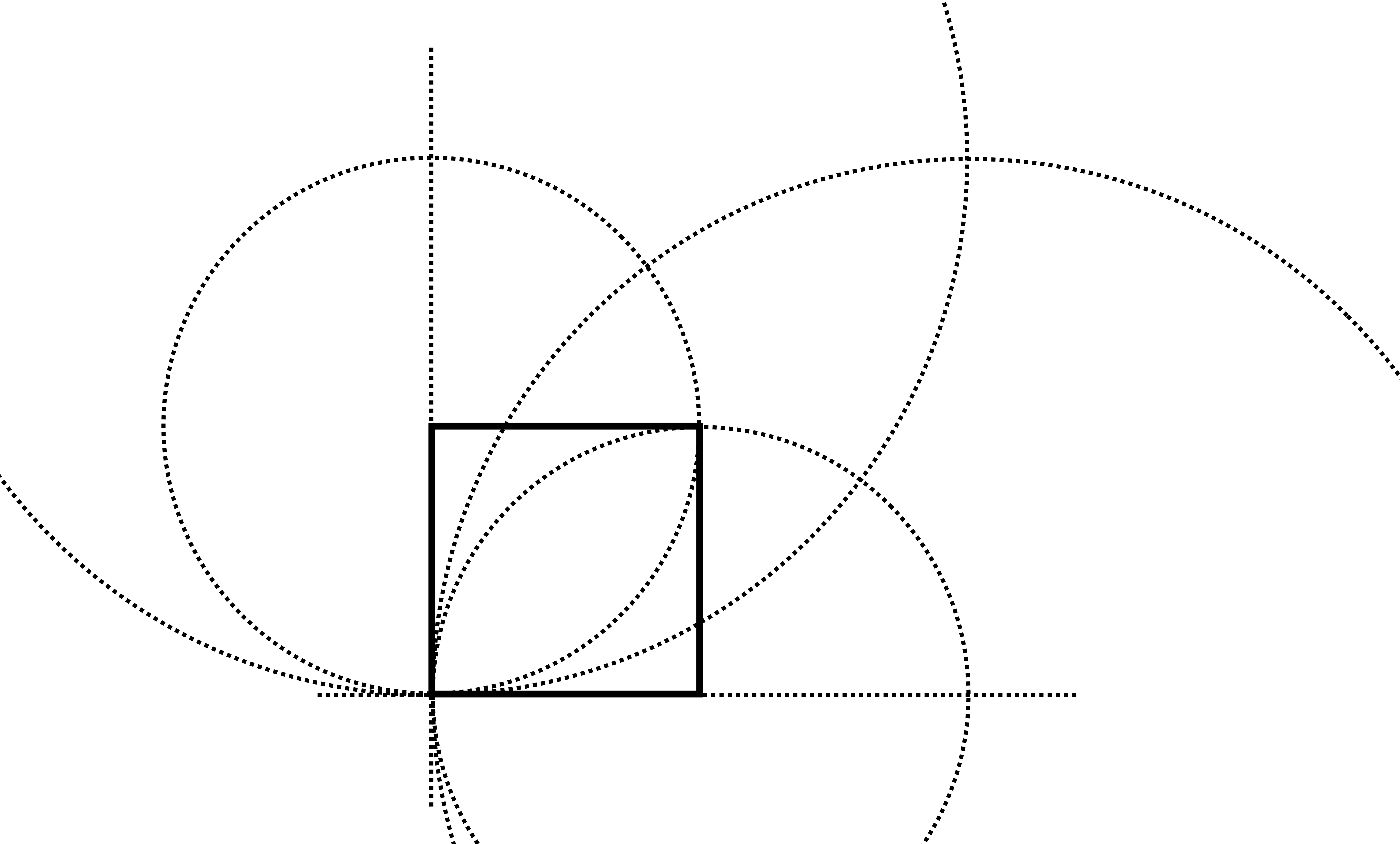
Quadruplement du carré



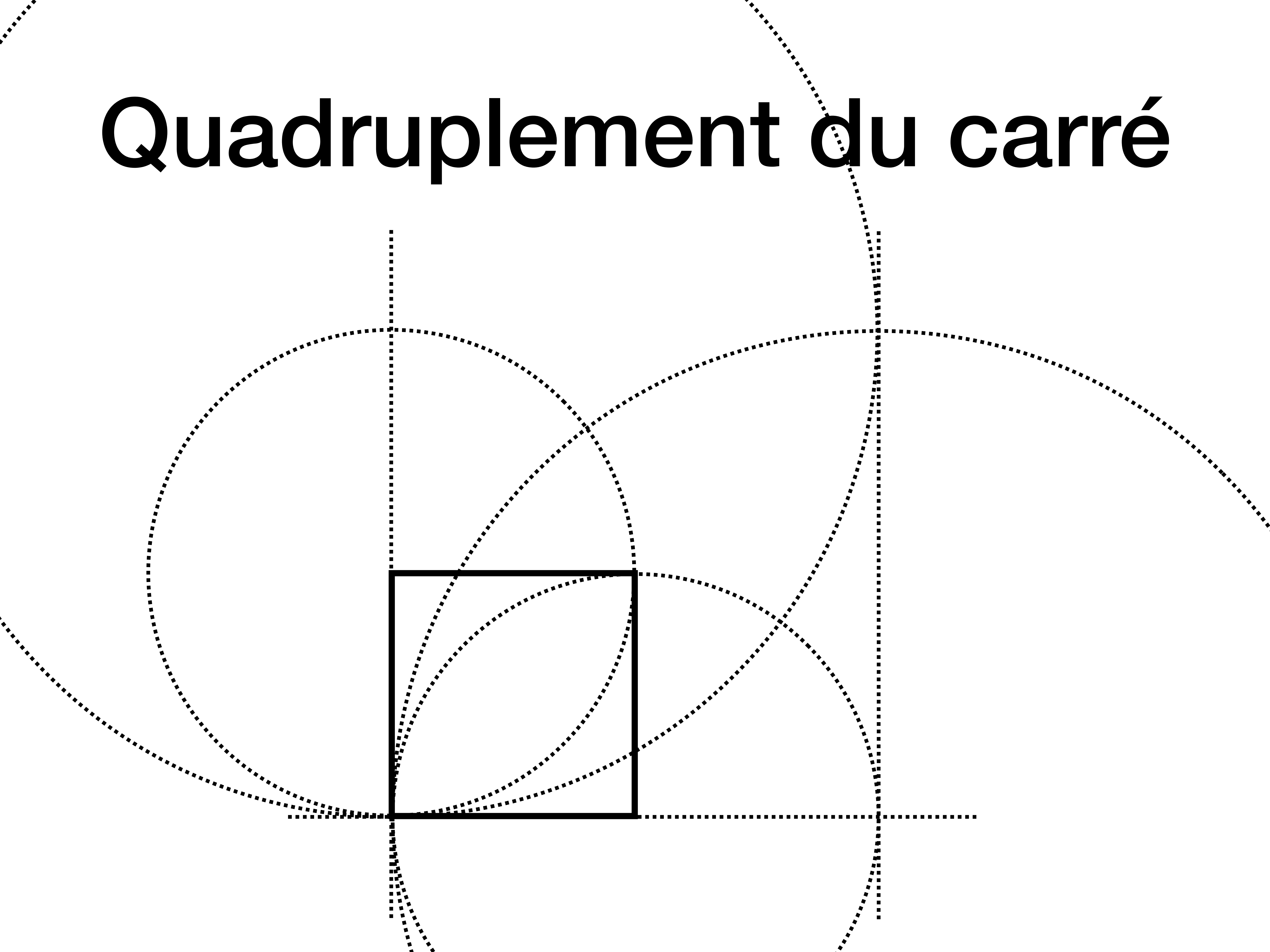
Quadruplement du carré



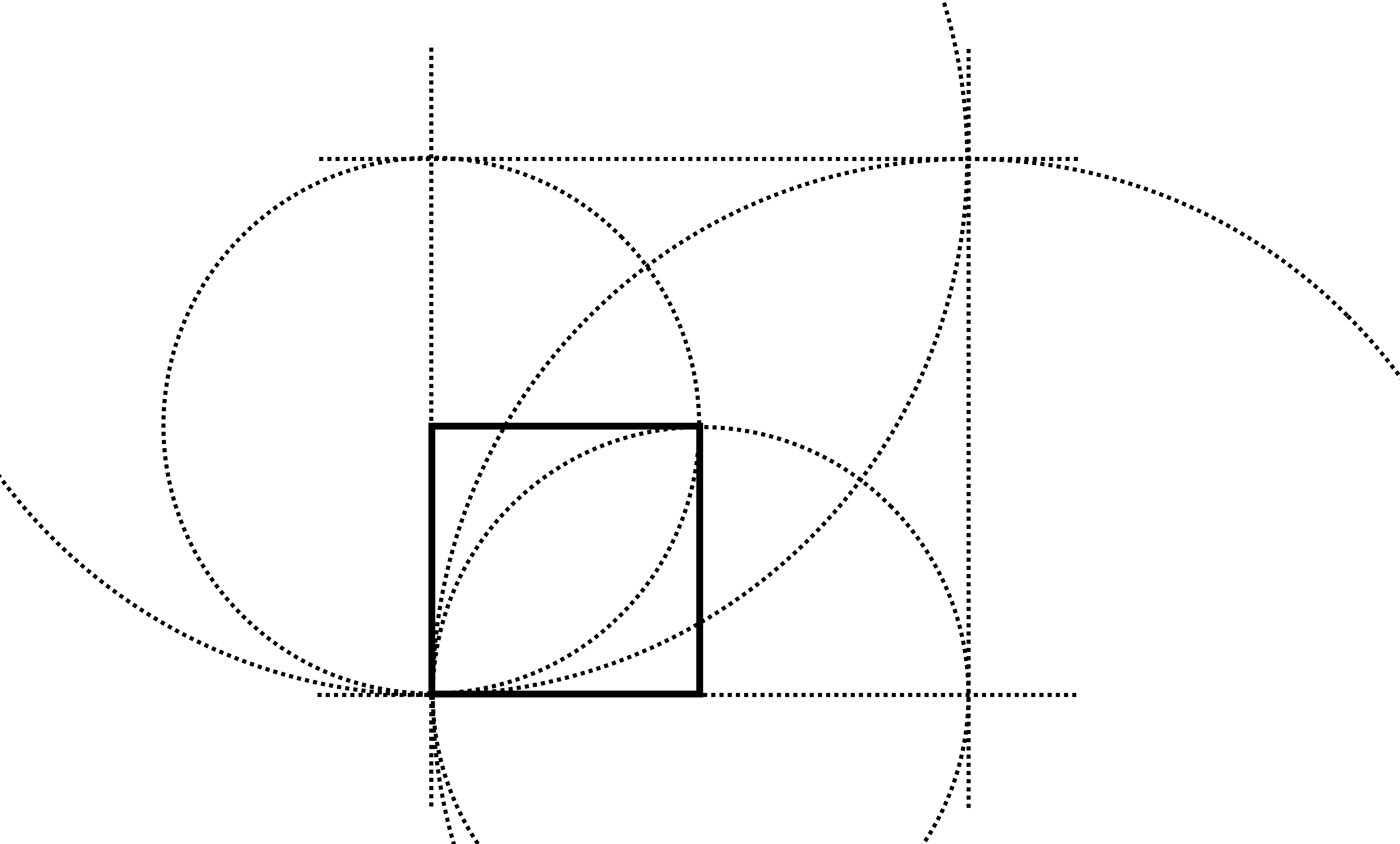
Quadruplement du carré



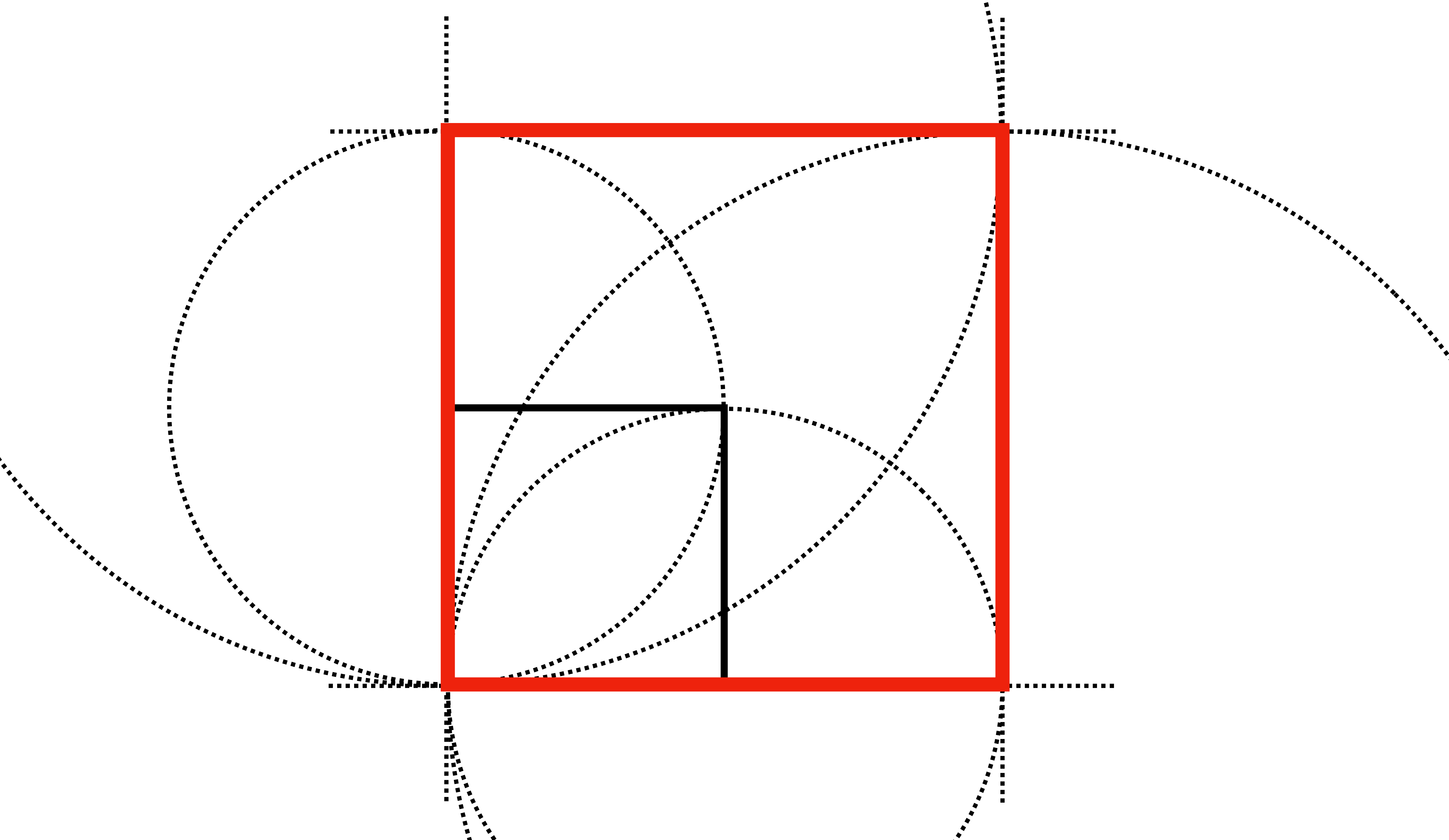
Quadruplement du carré



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

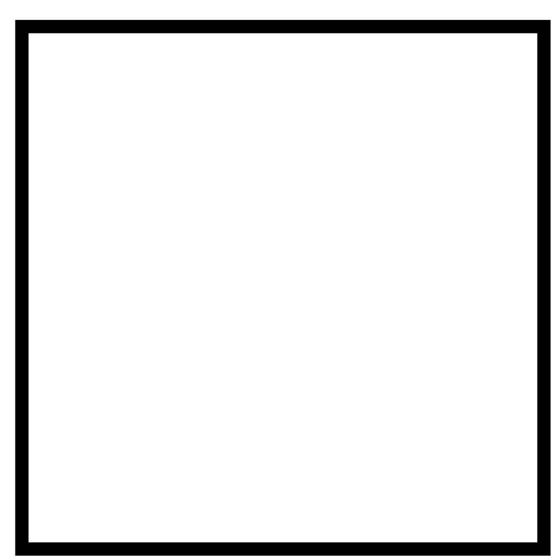
- Maintenant on sait qu'il existe des problèmes
  « bien formés » qui n'ont pas d'algorithme

Un « petit » problème sans solution algorithmique

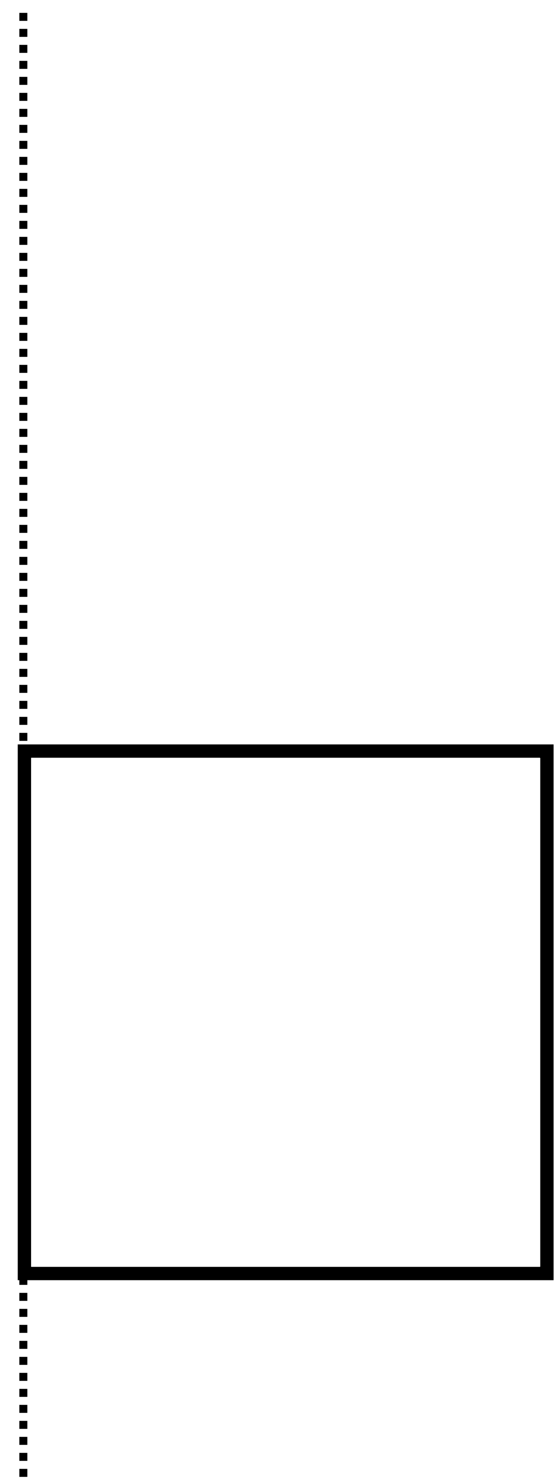
- Entrée : une proposition arithmétique φ formalisée
- Par exemple : $(\forall n > 2)(\nexists x, y, z \neq 0)(x^n + y^n = z^n)$
- Sortie : **oui** si on peut prouver φ, **no** si on ne peut pas
- Ce problème n'a pas d'algorithme !

Efficacité des algorithmes

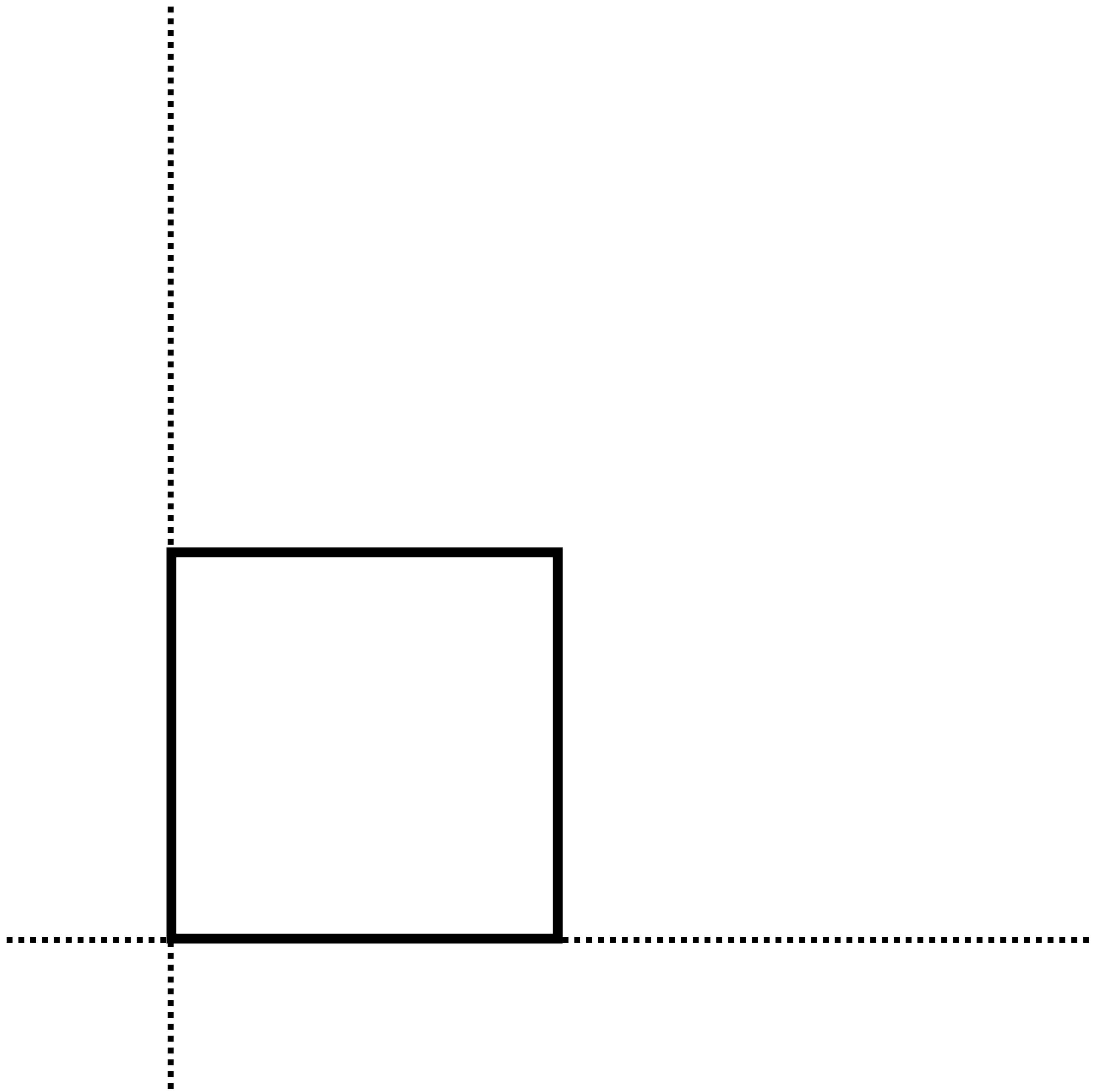
Quadruplement du carré



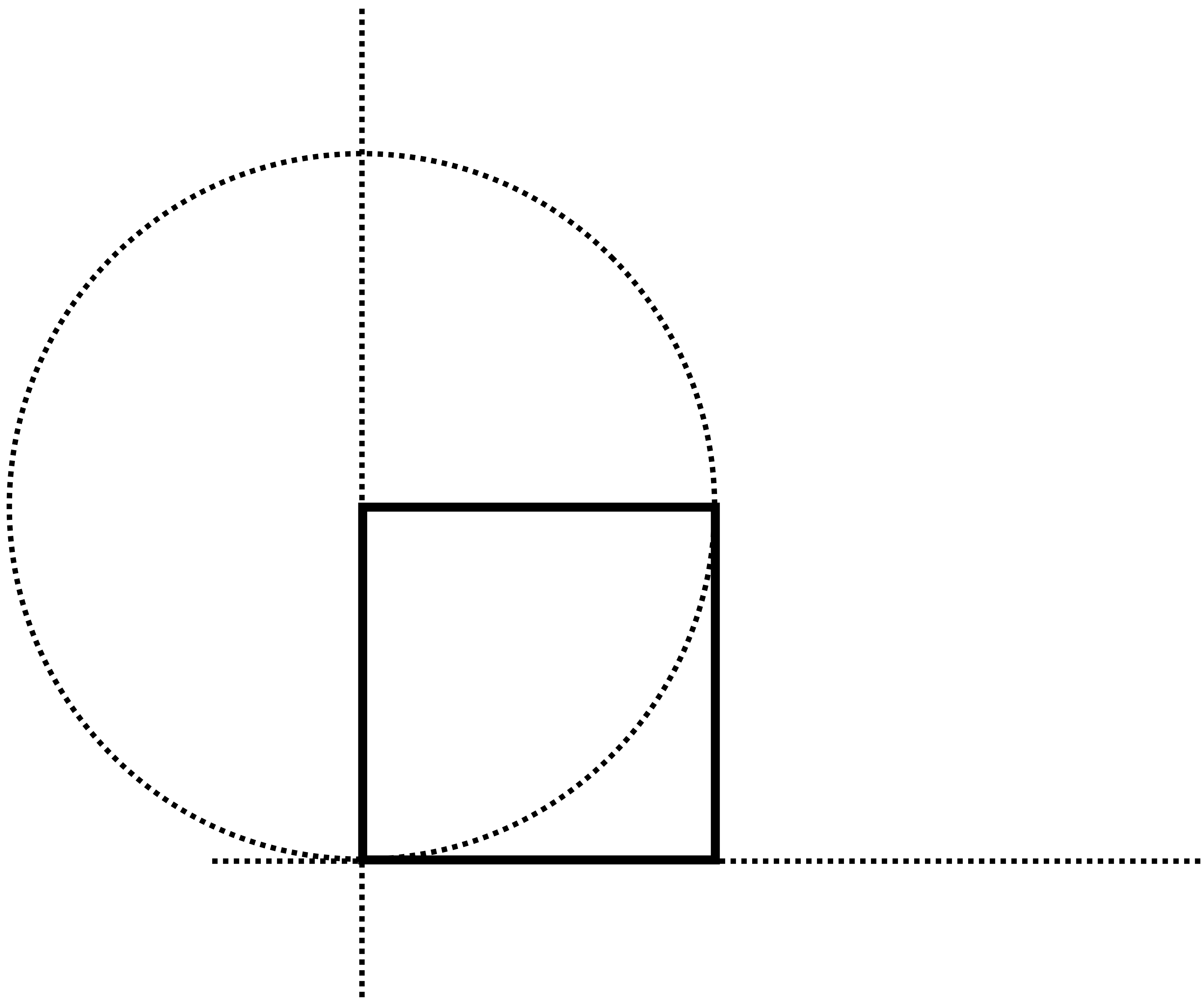
Quadruplement du carré



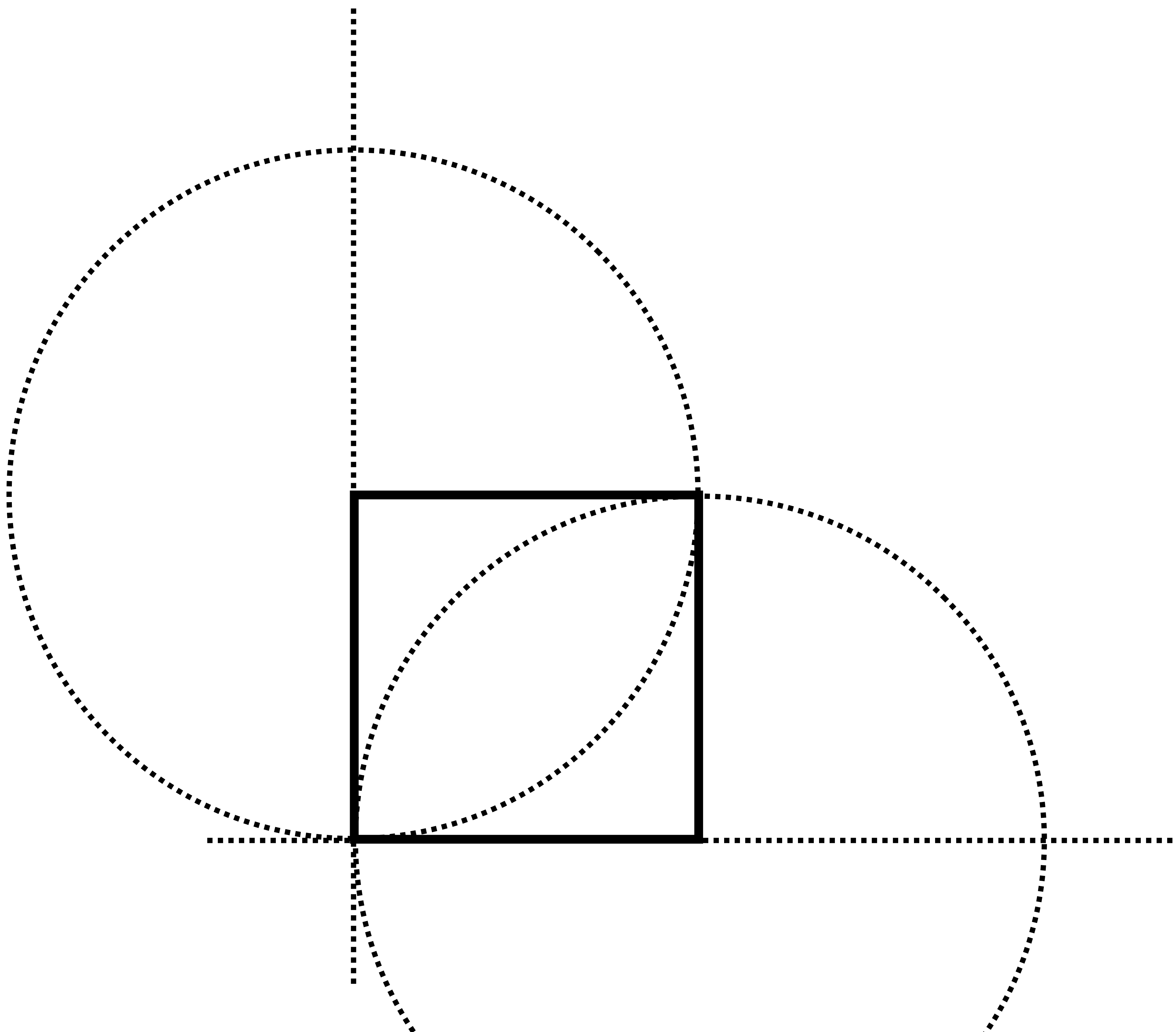
Quadruplement du carré



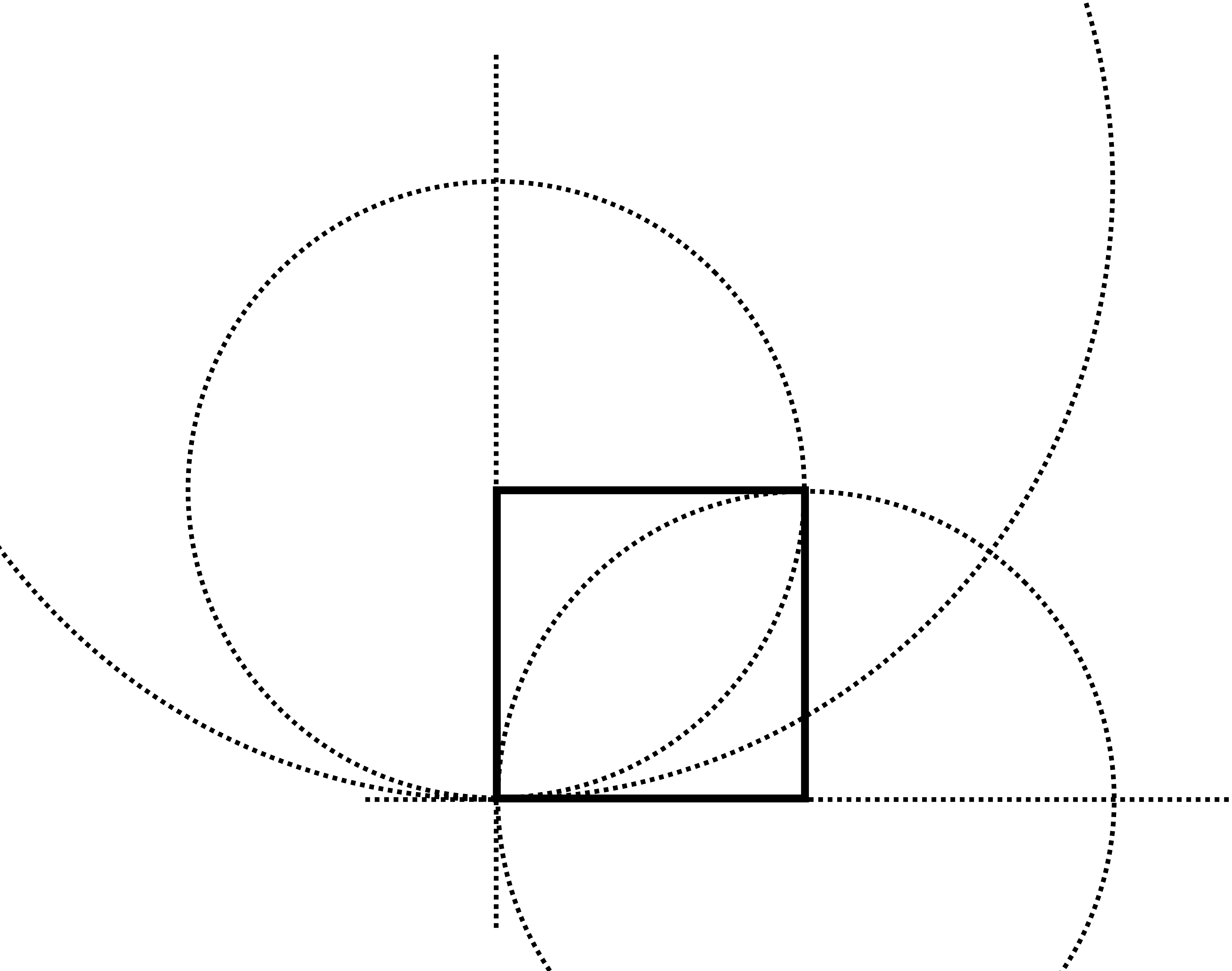
Quadruplement du carré



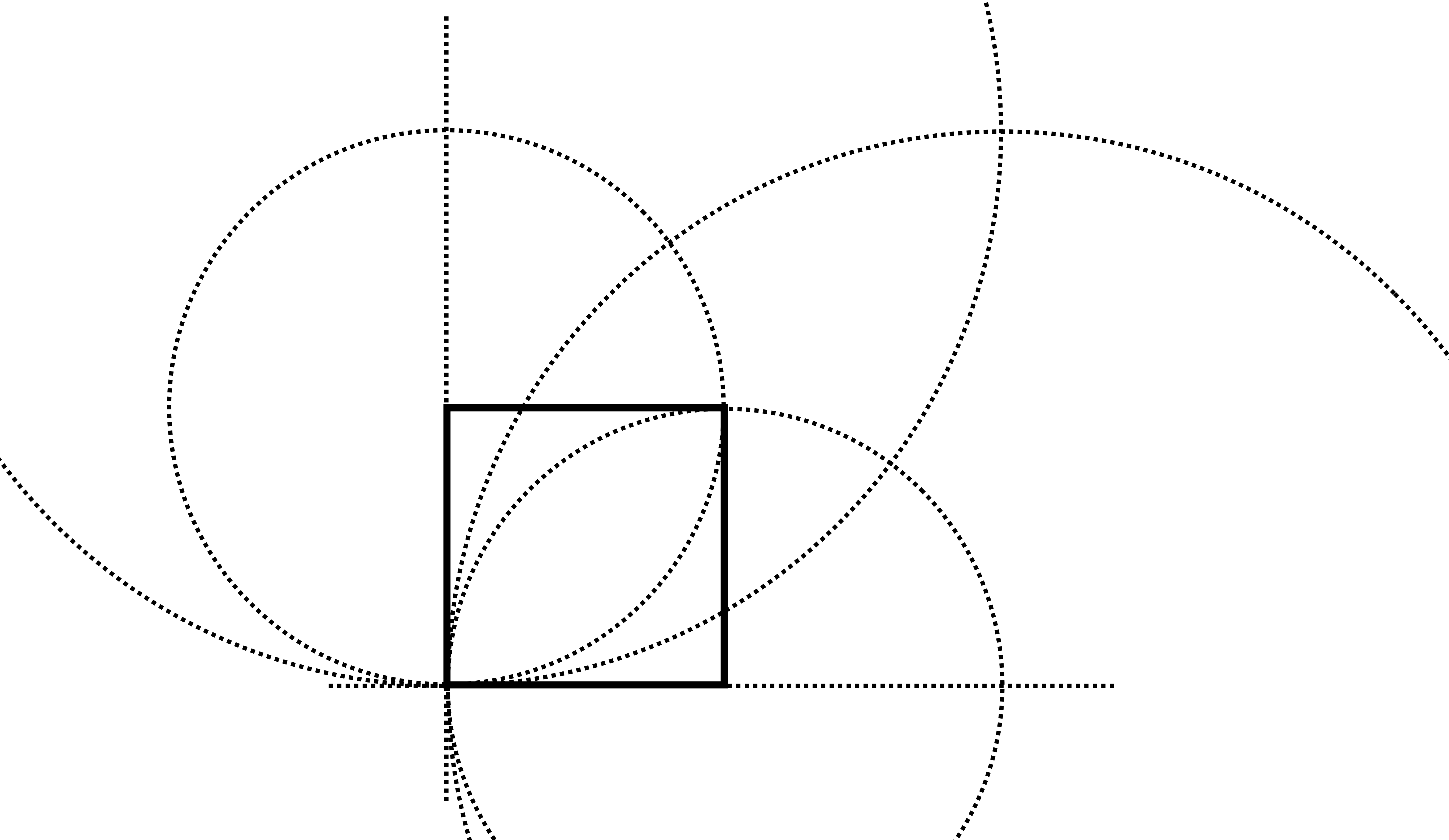
Quadruplement du carré



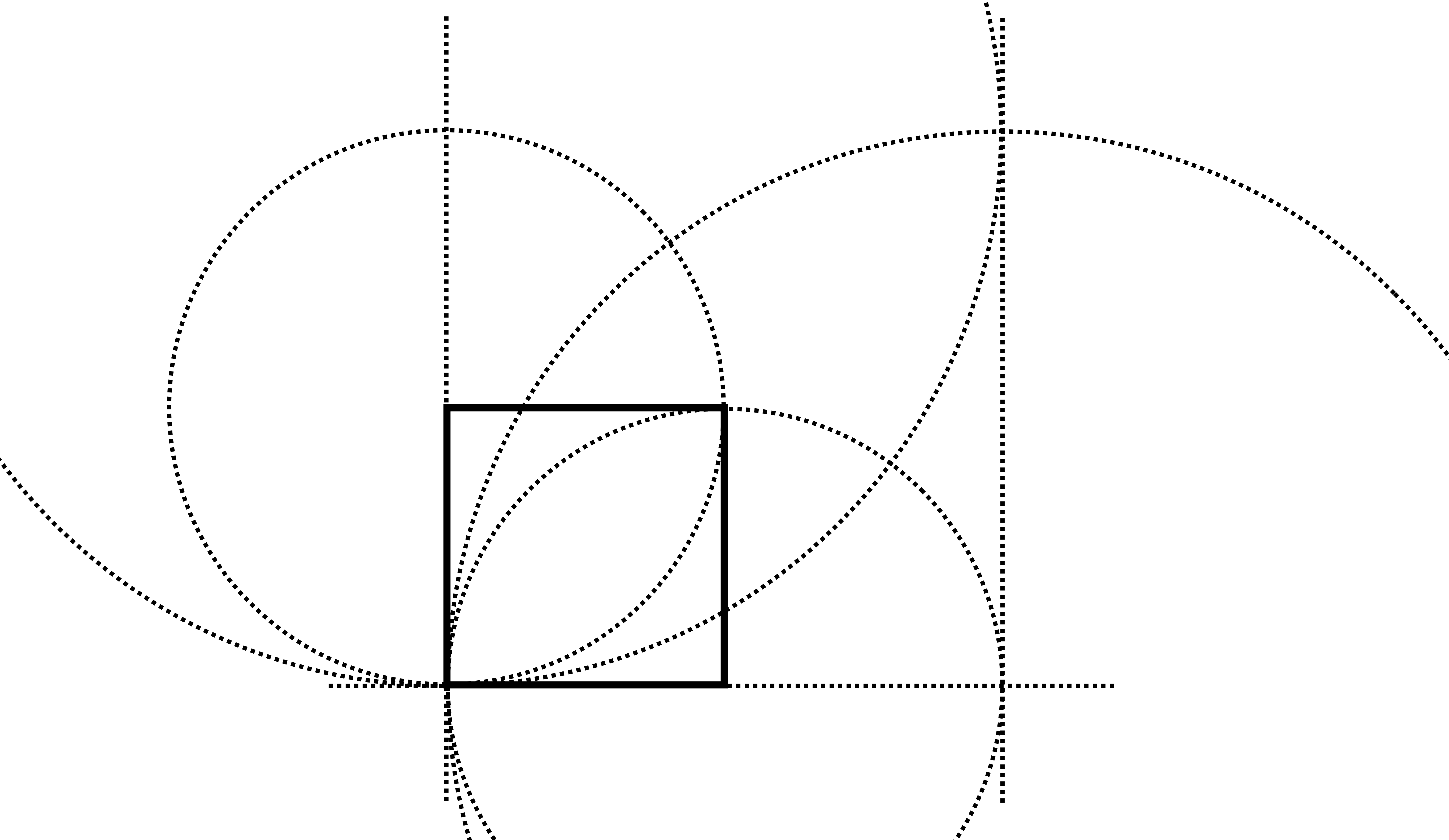
Quadruplement du carré



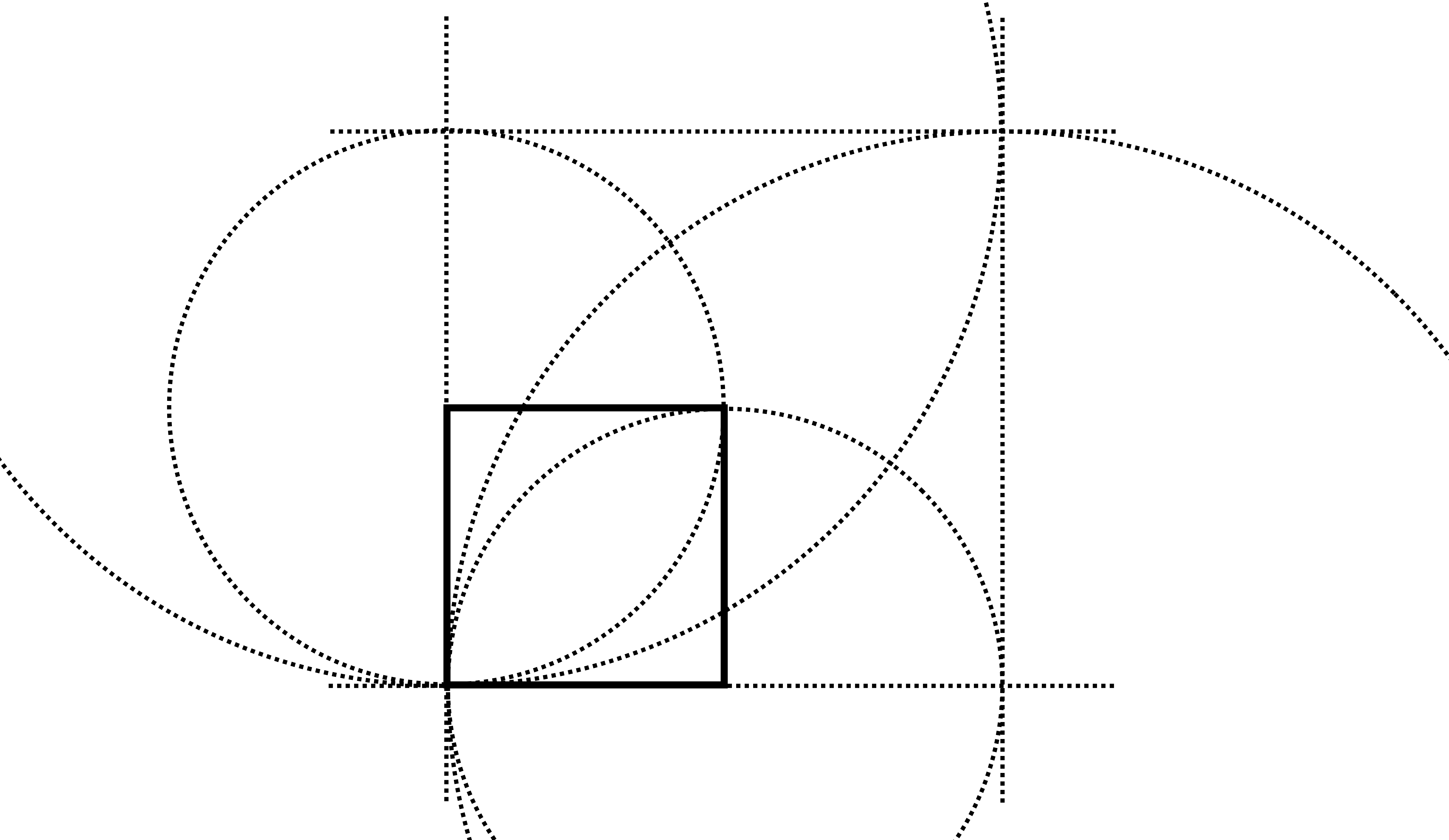
Quadruplement du carré



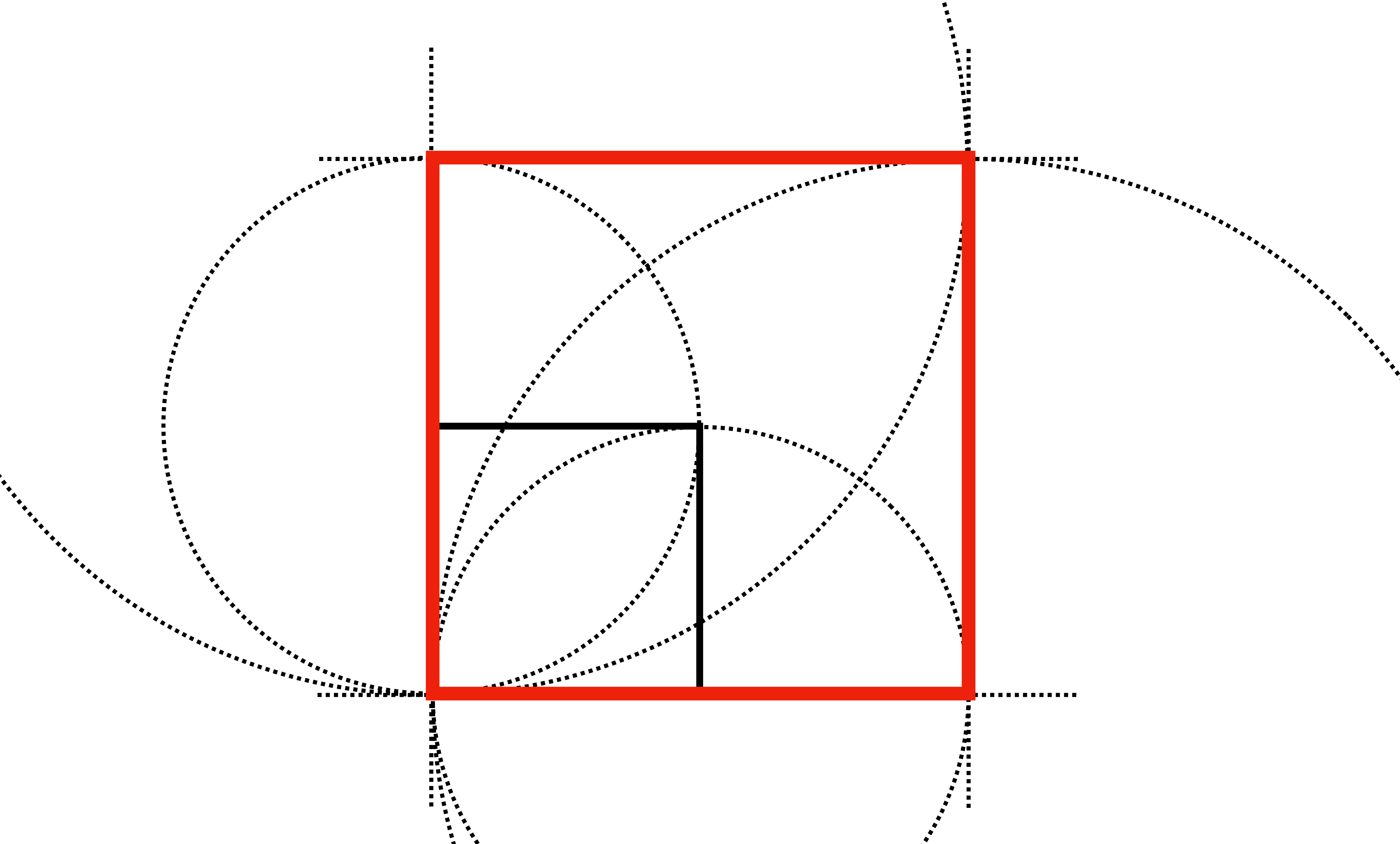
Quadruplement du carré



Quadruplement du carré



Quadruplement du carré



Efficacité des constructions à la règle et au compas

- Huit « 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 $n \to t(n)$ 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

$$\begin{array}{r} 234 \; + \\ \underline{281} \; = \\ 515 \end{array}$$

234
235
236
237
...
513
514
515

approximativement 3 opérations

au moins 281 opérations

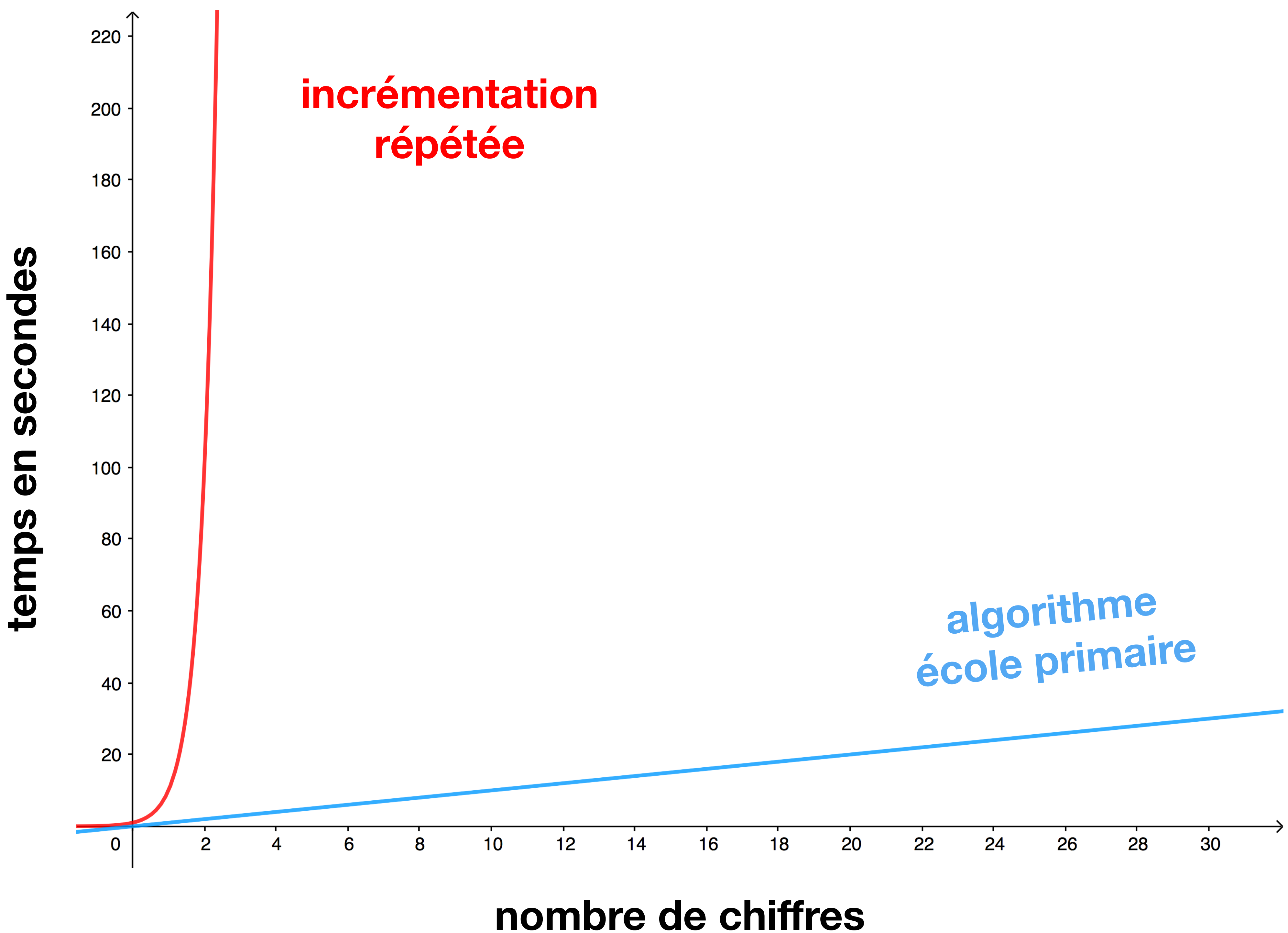
Efficacité des algorithmes pour l'addition

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

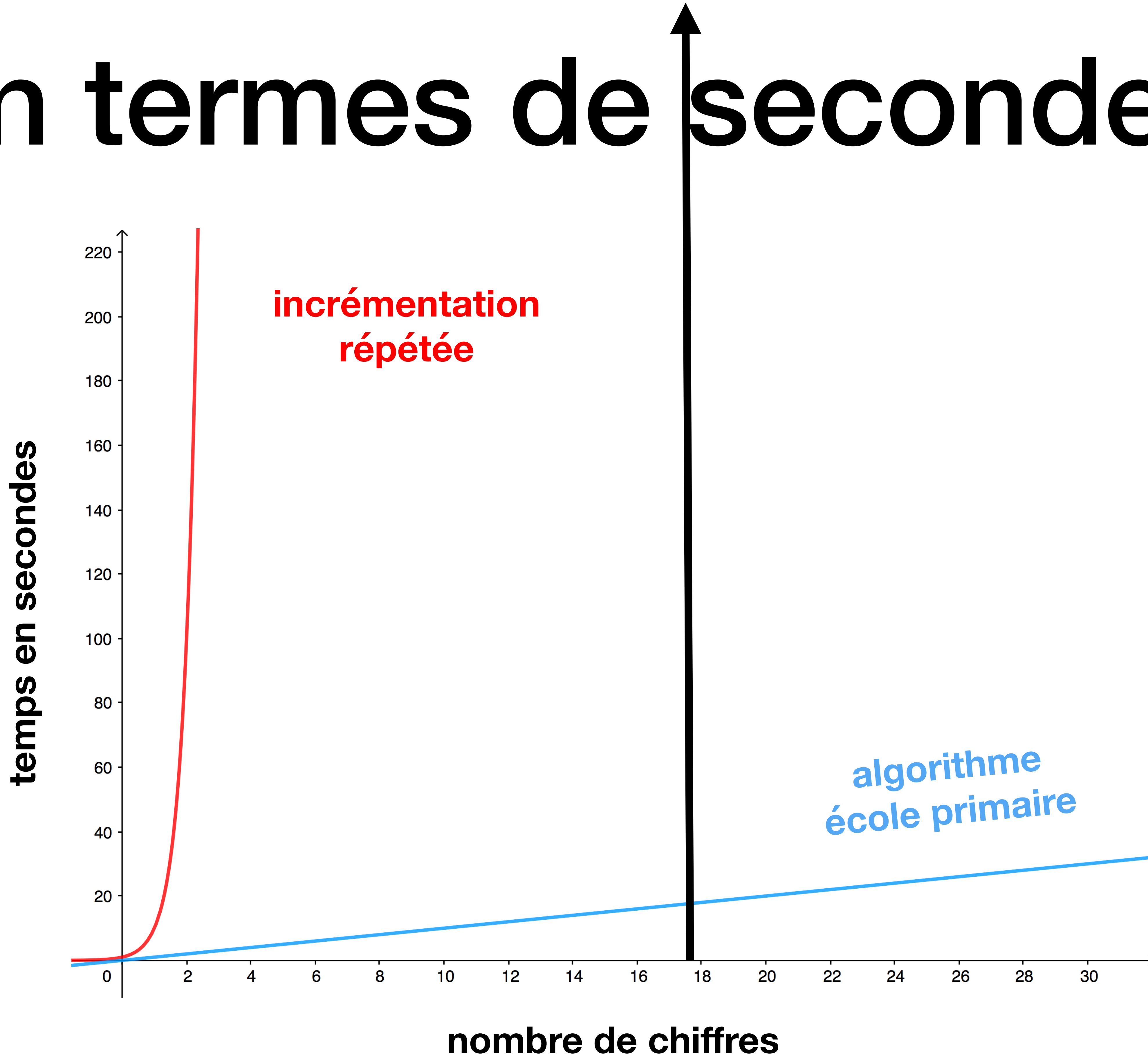
- Algorithme de l'« incrémentation répétée » : au moins 10^n opérations pour des nombres de n chiffres !

- Supposons qu'on fasse une opération par seconde

En termes de secondes



En termes de secondes



En termes de secondes