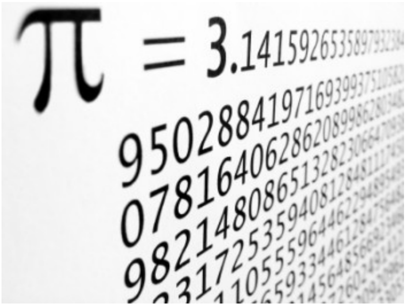
Java & Algorithme – Projet

Calculons le nombre PI



Le nombre PI est à la base de beaucoup de calculs en mathématiques (et pas seulement utilisé dans des calculs de rayon ou d'aire de cercles). Il existe beaucoup de méthodes afin de calculer le nombre PI, et récemment, une méthode très rapide a été mise en place par un ingénieur français, qui a réussi en quelques mois à calculer de façon la plus précise au monde ce nombre avec son seul ordinateur portable (http://linuxfr.org/2010/01/05/26318.html).

Une meilleure détermination des décimales (les chiffres après la virgule) du chiffre PI (il a été prouvé qu'il y en avait une infinité) permet non seulement une meilleure précision de certains calculs (vu la précision atteinte aujourd'hui, on peut s'estimer satisfait pour la précision des calculs), mais aussi une meilleure compréhension des concepts mathématiques à la base de ce nombre.

Dans ce projet, vous devrez comparer deux méthodes de calculs du chiffre PI, et déterminer laquelle est la meilleure. **Vous devrez rendre un petit rapport où, en plus d'expliquer les difficultés rencontrées lors de ce projet, vous donnerez vos conclusions sur quelle méthode est meilleure pour calculer Pi. Vous rendrez avec ce rapport votre code source et une rapide explication sur comment utiliser votre programme (pas besoin de copier coller l'intégralité de votre code dans le rapport).**

1 PREMIÈRE MÉTHODE DE CALCUL : CONSTRUIRE UN CERCLE ET COMPTER SES POINTS

Voici comment procéder pour calculer π avec cette première méthode. On considère tous les points à coordonnées entières contenus dans le carré allant du point **(0,0)** jusqu'à **(n,n)**. On va tenter de trouver quels sont les points entiers qui sont contenus dans le quart de cercle de rayon **n** et de centre **(0,0)**, c'est à dire les points à une distance plus petite ou égale à **n** du centre. Le nombre obtenu nous donnera une approximation de l'aire du cercle.

Pour obtenir π, on compte combien de points sont contenus dans le cercle afin d'obtenir une approximation de l'aire du cercle (par exemple, sur la figure A sur la page suivante, on a 26 points rouges dans le quart de cercle) : posons **m** comme étant le nombre de points dans le quart de cercle. Le nombre total de points dans le cercle entier (et pas seulement dans le quart de cercle) n'est pas égal à **4m**, car il faut éviter de compter plusieurs fois le centre et les points des bords. En réalité, le nombre total de points dans le cercle est égal à **4(m-n-1)+1**.

On sait que le cercle possède de rayon **n** (par exemple, sur la figure A, on a un cercle de rayon **5**). On sait que l'aire du cercle (à peu près égale à **4(m-n-1)+1**) est aussi égale à **πn^2**. On a donc :

$$\Pi \approx \frac{4(m-n-1)+1}{n^2}$$

En regardant les figures A, B et C (en rouge, les points considérés dans les cercles, en gris, les points hors des cercles), on se rend compte que plus **n** est grand, et meilleur sera le résultat du programme (vous aurez une meilleure approximation de π) mais aussi, plus le programme sera long à s'exécuter.

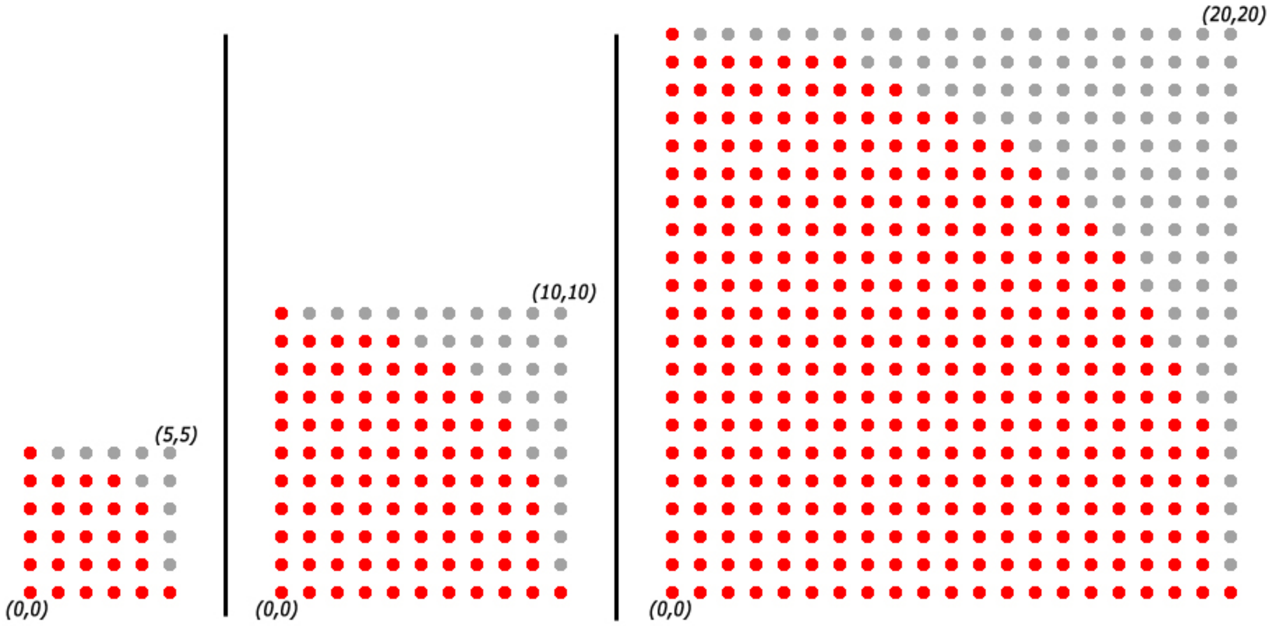


Figure A
n=5
m=26
pi≈3.24

Figure B
n=10
m=90
pi≈3.17

Figure C
n=20
m=335
pi≈3.1425

Pour implémenter cette formule, vous devrez parcourir tous les couples d'entiers **(a,b)** tels que **a** et **b** soient tous les deux compris entre **0** et **n** (inclus), et décider si le point de coordonnées **(a,b)** est inclus dans le cercle de rayon **n** et de centre **(0,0)**. En comptant tous les points inclus dans le cercle, vous pourrez en déduire une approximation de π avec la méthode décrite avant.

2 SECONDE MÉTHODE : DÉCRYPTER ET IMPLÉMENTER UNE FORMULE PLUS COMPLIQUÉE

Sur la page wikipedia de π, on trouve la formule de Madhava de Sangamagrama (http://fr.wikipedia.org/wiki/Pi#Formules_et_calculs_jusqu.27en_1900), qui donne :

$$\Pi = 4\sum_{k=0}^{\infty} \frac{(-1)^k}{2\mathrm{k}+1}$$

Évidemment, on ne peut pas faire une somme à l'infini... On choisira donc un nombre **n**, et on calculera π en faisant :

$$\Pi \approx 4\sum_{k=0}^{n} \frac{(-1)^k}{2\mathrm{k}+1}$$

Pour implémenter cette formule, vous utiliserez une boucle *tant que* afin de réaliser la somme de **k** égal à **0** jusqu'à **k** égal à **n**.

3 COMPARAISON DES MÉTHODES

Dans votre rapport, vous vous intéresserez à laquelle des deux méthodes est la plus performante. Pour ce faire, vous pouvez exécuter vos deux programmes java en modifiant la valeur de **n**, et en regardant quel résultat vous obtenez. Vous pourrez tracer, sur un graphique (utilisez par exemple Excel), le résultat obtenu par le programme en fonction du temps de calcul. Pour connaître le temps de calcul d'un programme, vous utiliserez la commande

```
time java MonProgramme
```

afin d'obtenir le temps d'exécution du programme. A temps d'exécution égal, la méthode qui donne le résultat le plus précis sera considérée comme la meilleure. Vous pourrez choisir π=3,141 592 653 589 793 238 pour vos comparaisons.