LINFO1104 Séance 1 *Extra*

Programmation récursive avec des entiers

Exercices supplémentaires

1. **Test de primalité.** Écrivez une fonction {Premier N} qui renvoie **true** si l'entier N est un nombre premier, **false** sinon. Considérez que N est premier s'il n'est divisible par aucun nombre k tel que $2 \leq k < \text{N}$.
   - Écrivez une fonction auxiliaire qui teste si un nombre N n'est divisible par aucun k tel que $\text{M} \leq k < \text{N}$. L'entier M est un paramètre de cette fonction. Utilisez cette fonction pour définir Premier.
   - On peut améliorer l'efficacité de cette fonction en ne testant que les k inférieurs ou égaux à la racine carrée de N. En effet, si $\text{N} = k * l$ avec $k \leq l$, alors $k^2 \leq \text{N}$. Modifiez votre fonction auxiliaire en conséquence.

2. **Fibonacci (première partie).** Écrivez une fonction récursive qui calcule le n-ième nombre de Fibonacci, donné par la définition suivante.

   fib(0) = 0
   fib(1) = 1
   fib(n) = fib(n-1) + fib(n-2) si $n > 1$

   - Combien d'appels récursifs sont effectués si n = 1 ?
   - Combien d'appels récursifs sont effectués si n = 4 ?
   - Combien d'appels récursifs sont effectués si n = 5 ?
   - Combien d'appels récursifs sont effectués si n = 8 ?
   - Combien d'appels récursifs sont effectués avec n ?

3. **Fibonacci (deuxième partie).** Écrivez une nouvelle version de Fibonacci, où le nombre d'appels récursifs pour n est n. *Aide:* utilisez une fonction avec un (ou plusieurs) accumulateur(s). Quel est l'invariant de cette fonction?

4. **Diviseurs et multiples.** En arithmétique, on utilise fréquemment le *plus grand commun diviseur* entre deux nombres entiers. On peut le définir comme une fonction *pgcd* à deux variables entières qui satisfait les propriétés

   pgcd(m, n) = pgcd(n, m)
   pgcd(m, n) = pgcd(m+n, n)
   pgcd(m, m) = m

   De la seconde propriété, on peut dériver

   pgcd(m, n) = pgcd(r, n), où r est le reste de la division de m par n.

   - Implémentez une fonction PGCD en Oz, en utilisant ces propriétés. Cette fonction prend deux paramètres entiers. *Aide:* Sous quelle(s) condition(s) la propriété dérivée est-elle utile, c'est-à-dire $r \leq m$?
   - Implémentez une fonction PPCM prenant deux paramètres entiers et renvoyant leur *plus petit commun multiple*. Réutilisez votre fonction PGCD.

5. **Numérotation des points du plan.** Il existe une technique relativement simple pour affecter un entier naturel à chaque couple d'entiers naturels *(x, y)*. La figure suivante illustre la technique, qui consiste à numéroter les diagonales successives du quart de plan. Les numéros des points sont en bleu.

   Écrivez une fonction Numero, prenant en paramètre deux naturels x et y, et renvoyant leur numéro suivant la technique illustrée ci-dessus. Nous proposons deux techniques.

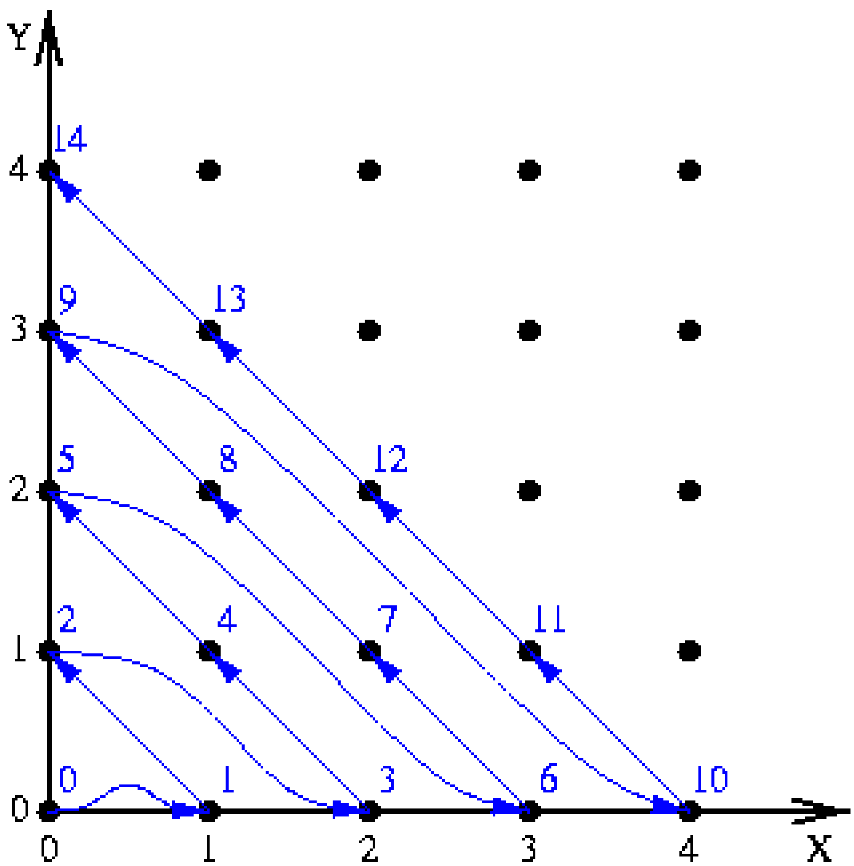

Figure 1: Numérotation des points du plan

- Exprimez le numéro d'un point en fonction du numéro de son prédécesseur. Le prédécesseur est défini en suivant les flèches bleues en arrière. Par exemple, les prédécesseurs successifs de *(3, 2)* sont *(4, 1)*, *(5, 0)*, *(0, 4)*, *(1, 3)*, ... Implémentez une version itérative de Numero.
- Améliorez l'efficacité de votre fonction sur base des deux observations suivantes. Tout d'abord, un point *(x, y)* se trouve sur la même diagonale que le point *(x+y, 0)*. Ensuite, le numéro d'un point sur l'axe des X, de la forme *(x, 0)*, correspond au nombre de points dans le triangle *(0, 0)*, *(x-1, 0)*, *(0, x-1)*. Et ce nombre est un nombre triangulaire...
- Étant donné un numéro n, pouvez-vous retrouver les coordonnées *(x, y)* du point correspondant? *Aide:* Trouvez d'abord la coordonnée x en comparant n aux numéros des points du type *(x, 0)*. Ensuite, calculez la coordonnée y. Vos fonctions sont-elles itératives? Quel est leur invariant?

6. **Sous les pavés...** Un paveur philosophe se pose un jour la question suivante: *"De combien de façons puis-je paver une surface carrée de côté entier* n *avec des pavés carrés identiques et de côté entier?"* L'exemple ci-dessous montre les quatre pavages possibles pour n=6. De gauche à droite, on a utilisé des pavés de côté 6, 3, 2 et 1.

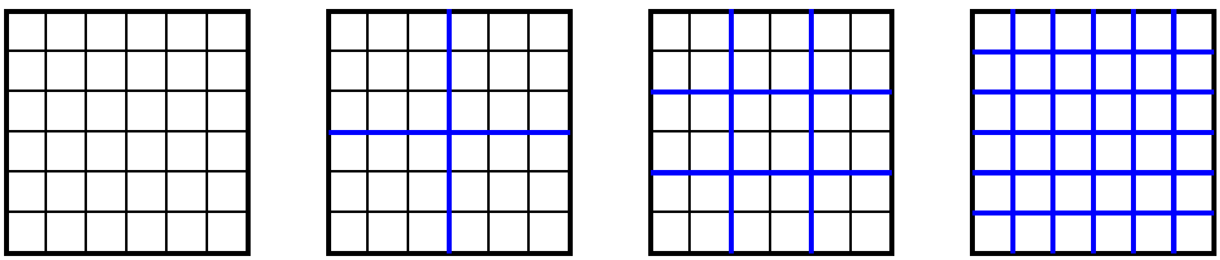

Figure 2: Pavages possibles pour n=6

- Ecrivez une fonction NombrePavages qui prend n en paramètre et renvoie le nombre de pavages en question.
- Supposez maintenant que le paveur réalise tous les pavages possibles. Combien de pavés aura-t-il utilisé? Dans l'exemple *n=6*, il y en a *1+4+9+36=50*. Ecrivez une fonction NombrePaves qui prend n en paramètre et renvoie ce nombre.

Guillaume Maudoux, Boriss Mejías et Raphaël Collet - 26 septembre 2007 (Last Update : 5 février 2020)