Document de travail pour le TD2

Question 1

En vous inspirant des questions 3 et 4 du TD1 (puissance rapide et suite de Fibonacci), vous devez :

- écrire un calcul de puissance rapide sur des matrices 2 lignes-2colonnes ;
- utiliser ce calcul pour écrire une version matricielle du calcul de la suite de Fibonacci.

Question 2

En travaillant sur des tableaux d'entiers, vous devez écrire en récursif les tris vus en cours :

- tri naïf (par insertion) ,
- tri par fusion,

ainsi qu'une recherche dichotomique dans un tableau trié.

Question 3

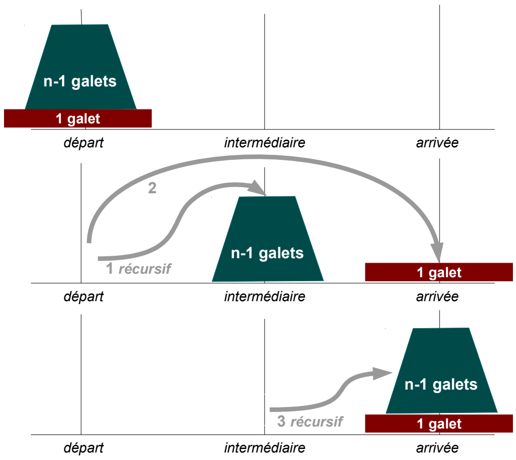
En vous inspirant du traitement récursif *propMax(i),* vu au tD1, vous devez écrire un tri par tas (heapsort) qui consiste à construire un tas maximum pour trouver le plus grand élément afin de le placer en fin de tableau.

Ouverture sur les TP

Exercice sur les tours de Hanoi

Pour cet exercice, vous pouvez travailler sur la base suivante :

- le contrôle de la récursivité se fait par le nombre *n* de galets à déplacer,
- l'**objectif** est de revenir à un déplacement simple (par exemple *n*=3 ou *n*=2),
- la **réduction** est similaire au déplacement pour *n*=2, à savoir en combinant le déplacement du plus grand jeton de la tour de départ vers la tour d'arrivée et deux appels récursifs pour *n-1* galets (en excluant le plus grand des galets) :
    - de la tour de départ vers la tour intermédiaire,
    - de la tour intermédiaire vers la tour finale,
- la valeur de *n*, diminuée de 1 à chaque appel, permet de garantir l'**arrêt des appels récursifs**.



En plus du nombre de galets à déplacer, il faut indiquer pour chaque appel récursif quel rôle joue chacune des tours. L'algorithme proposé ci-dessous prend comme déplacement simple le cas où *n*=2. Les trois tours sont indiquées dans l'ordre départ, arrivée, intermédiaire.

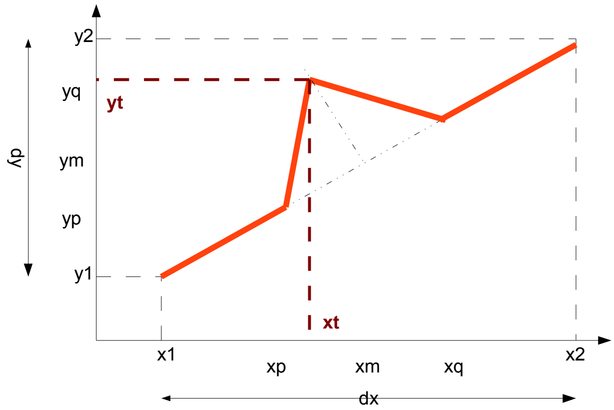
```
hanoi(n,D,A,I) :
       si n=2
       alors 3 déplacements de galets(D vers I, D vers A, I vers A)
       sinon hanoi (n-1, D, I, A)
             déplacer de D vers A
             hanoi (n-1, I, A,D)
       fsi
```

Attention, dans l'énoncé fourni en TD le nombre de galets doit être supérieur à trois : ceci ne concerne que l'appel initial et non les appels récursifs !

La courbe de Von Koch

Pour cet exercice, vous devez réfléchir à la partie récursive puis compléter le programme Java qui vous est fourni (ce programme effectue des appels à des primitives graphiques). Les calculs des segments à tracer sont construits selon le schéma ci-contre :

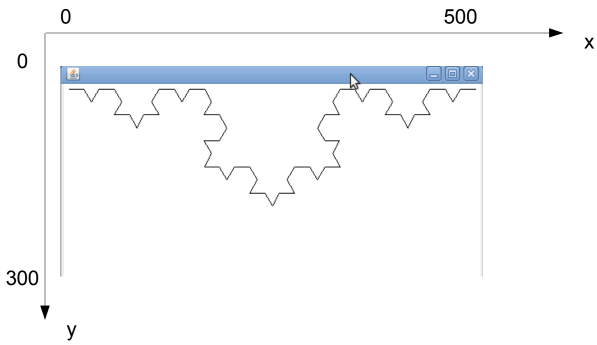
- le premier et le dernier tiers du segment entre les points (x1,y1) et (x2,y2) sont tracés simplement,
- le tiers médian de ce segment est remplacé par les deux côtés du triangle équilatéral construit sur la partie non tracée,
- la distance horizontale entre les deux points est notée dx=x2-x1,
- la distance verticale entre ces points est notée dy=y2-y1,
- les coordonnées des différents points à utiliser sont indiquées ci-contre.



les points au tiers du segment
xp=x1+dx/3 yp=y1+dy/3
xq=x1+2dx/3 yq=y1+2dy/3
le point de milieu de segment
xm=xp+dx/2 ym=y1+dy/2

le sommet du triangle (xt,yt)
$xt=xm-dy/\sqrt{12}$
$yt=ym+dx/\sqrt{12}$

Lorsque vous allez exécuter ce programme, la fenêtre graphique est orientée verticalement vers le bas. La courbe est donc construite avec la pointe vers le bas. Dans les corrections proposées sur la plateforme *ufrsciencestech.u-bourgogne.fr*, vous trouverez un programme *VonKoch.java* qui trace quatre courbes successives pour construire un flocon de neige.

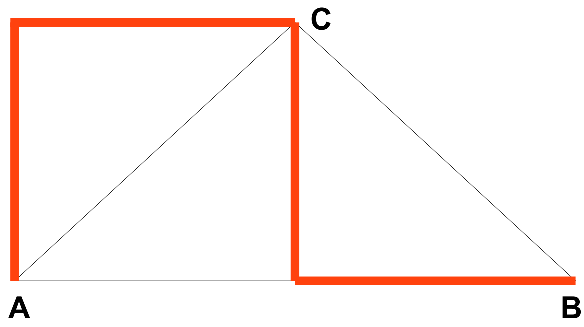


Le programme à compléter est donné dans un fichier pourVonKoch.schema, n'oubliez pas de renommer ce ficher.

La courbe du dragon

A partir de l'exemple de la courbe de Von Koch, contruisez une courbe du dragon, dont la définition est la suivante : tout segment est simplement tracé ou (s'il est trop long) remplacé par quatre segments constitués à partir du triangle rectangle isocèle basé sur ce segment.

Vous utiliserez, comme pour la courbe de Von Koch, un calcul récursif ayant pour paramètres les coordonnées des points délimitant le segment ainsi que le niveau de profondeur de la courbe. Ce niveau de profondeur détermine le choix entre un tracé simple ou un tracé récursif (pour lequel vous devez calculer les coordonnées du point C et du milieu du segment AB).



segment initial AB
triangle rectangle en C, isocèle