Récurrences par élévation au carré modulo *m* . Graphe orienté des trajectoires

Nous étudions ici la récurrence $x_{n+1} = x_n^2$ [*m*], à partir des éléments x_0 de *U*(*m*),[1] la succession x_0, x_1, x_2, .. constituant ce que l'on appelle la trajectoire de x_0. Nous constaterons que le tracé des composantes connexes du graphe des trajectoires révèle de nombreuses propriétés relatives à la structure du groupe *U*(*m*), notamment à propos des racines carrées de ses éléments, et de leur ordre. L'étude est d'abord faite pour *m* = *p* premier, puis elle est étendue au cas où *m* est une puissance de nombre premier. Enfin nous passons au cas général où *m* est un nombre composite, ce qui nous amènera à définir une composition d'arborescences. Nous développons plus précisément le problème avec son arrière-plan algébriste, tout en dégageant la mécanique algorithmique qui permet de construire progressivement la floraison des arborescences.

Plaçons-nous dans *U*(*m*), l'ensemble des éléments inversibles de *Z/mZ*, où l'on pratique l'itération répétée $x' = x^2$, à partir de x_0 donné, ce qui donne une trajectoire en forme de ρ. En faisant de même pour chaque élément x_0 de *U*(*m*), on aboutit à un graphe de toutes les trajectoires, d'où ressortent les composantes connexes. On distingue plusieurs cas suivant les valeurs de *m*.

1. Premier cas : *m* est un nombre premier impair *p*

1.1. Arborescence *U* de 1

- **1 est le seul point fixe**, puisque l'équation $x^2 = x$ n'admet que 1 pour solution dans *U*(*p*).[2]

- **Les trajectoires convergeant vers le point fixe 1 forment un arbre binaire avec un tronc, ayant *a* étages et 2^a éléments, *a* étant tel que $p - 1 = 2^a \times h$ avec *h* impair. Chaque étage *e* à partir de l'étage 1 compte 2^{e-1} éléments.**

Intéressons-nous aux points *x* convergeant vers ce point. L'élément 1 possède deux antécédents *x* vérifiant $x^2 = 1$, à savoir 1 et –1. A son tour –1 admet deux antécédents ou non selon qu'il est un carré ou pas.

Pour *p* de la forme 4*k* + 3, –1 n'est pas un carré[3], et l'arborescence de racine 1 se réduit à un seul étage comportant –1 (*figure 1*).

---

[1] *U*(*m*) est l'ensemble des nombres ramenés modulo *m* qui sont inversibles. Il s'agit aussi des éléments de *Z*/*mZ* = {0, 1, 2, ..., *m* – 1} qui sont premiers avec *m*.

[2] *U*(*p*) est l'ensemble des éléments inversibles de *Z*/*pZ*, soit *U*(*p*) = {1, 2, ..., , *p* – 1}, 0 étant le seul élément non inversible de *Z*/*pZ*.

[3] On applique ce théorème : **dans *U*(*p*) l'équation $x^n = a$ admet $pgcd(n, p-1)$ solutions ou aucune selon que $a^{\frac{p-1}{pgcd(n,p-1)}} = 1$ ou non**. Pour $x^2 = -1$ avec $p = 4k + 3$, $pgcd(2, p-1) = 2$ et $(-1)^{2k+1} = -1 \neq 1$.

Ce théorème est une généralisation du critère d'Euler : Un élément *a* de *U*(*p*) admet deux racines carrées *x* (avec $x^2 = a$) si et seulement si $a^{(p-1)/2} = 1$.

Figure 1 : Arborescence menant à 1, réduite à un tronc lorsque p est de la forme $4k + 3$, par exemple 3, 7, 11, 19, etc.

Si p est de la forme $4k + 1$, l'élément –1 situé à l'étage 1 admet deux antécédents.. Ceux-ci vérifient $x^4 = 1$ mais pas $x^2 = 1$. Comme l'équation $x^4 = 1$ admet quatre solutions, dont deux vérifient $x^2 = 1$, on obtient à l'étage 2 deux noeuds. En écrivant $p - 1 = 2^a h$ avec h impair, l'équation $x^{2^a} = 1$ admet 2^a solutions et si l'on prend $a' > a$, l'équation $x^{2^{a'}} = 1$ admet aussi 2^a solutions[4]. L'arborescence convergeant vers 1 possède exactement 2^a éléments.

Si l'on numérote les étages de l'arborescence à partir de l'étage 0 de la racine, l'étage 1 compte un élément, l'étage 2 éventuel en compte 2, l'étage 3 éventuel en compte 4, etc. Jusqu'à l'étage e on a 2^e éléments, tous ceux vérifiant $x^{2^e} = 1$, et jusqu'à l'étage précédent $e - 1$, on en avait 2^{e-1}, tous ceux vérifiant $x^{2^{e-1}} = 1$. L'étage e en comporte exactement 2^{e-1}, tous ceux vérifiant $x^{2^e} = 1$, mais pas $x^{2^{e-1}} = 1$, ce qui impose que chaque noeud de l'étage inférieur $e - 1$ admet deux antécédents[5]. Par récurrence évidente, il en est ainsi jusqu'au dernier étage a. L'arborescence de 1 est un arbre binaire totalement équilibré, en ce sens que toutes les feuilles de l'arbre sont à la même hauteur, et il a un tronc à sa base. Chaque noeud de cet arbre, à l'exception de sa racine et de ses feuilles, possède deux antécédents (*figure 2*).

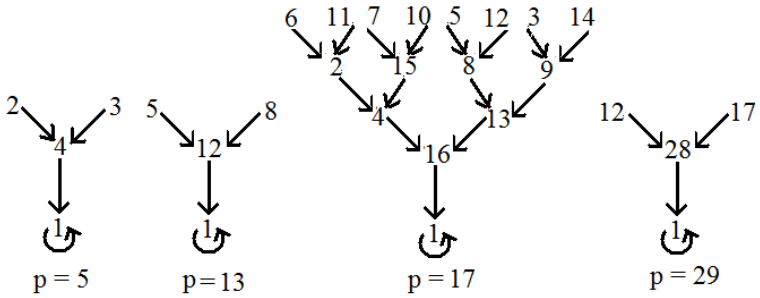


Figure 2 : Quelques arborescences de racine 1, lorsque le modulo p est un nombre premier de la forme $4k + 1$. Pour $p = 5$, 13 ou 29, l'arborescence possède 2 étages, tandis que pour $p = 17$, elle en a 4.

---

[4] Dans $U(p)$, l'équation $x^n = 1$ admet $pgcd(n, p - 1)$ solutions, et notamment si d divise $p - 1$, l'équation $x^d = 1$ admet exactement d solutions, notamment $x^{2^a} = 1$ admet 2^a solutions lorsque $p - 1 = 2^a h$.

Cette propriété n'est pas seulement valable pour $U(p)$, elle s'applique aussi au groupe multiplicatif d'un corps commutatif fini, et plus largement à tout groupe fini cyclique. Nous pourrons l'appliquer plus loin dans $U(p^k)$ qui est cyclique pour p premier impair. Cette propriété peut s'exprimer plus généralement ainsi :

Soit G un groupe fini cyclique d'ordre N. L'équation $x^n = a$ admet $pgcd(n, N)$ solutions ou aucune selon que $a^{N / pgcd(n, N)} = 1$ ou non.

Démonstration: En considérant un générateur g de G, a peut s'écrire $a = g^i$. Chercher x vérifiant $x^n = a$ revient à chercher u tel que $g^{nu} = g^i$, c'est-à-dire $n u = i\ [N]$. Posons $k = pgcd(n, N)$. Si k divise i, il y a k solutions. Si k ne divise pas i, il n'y a pas de solutions. Le fait que k divise i donne $a^{N/k} = g^{i(N/k)} = g^{N(i/k)} = 1$. Par contre si k ne divise pas i, $a^{N/k} = g^{iN/k} \neq 1$.

[5] On sait que $x^2 = a$ admet deux solutions ou aucune. Ici elle en a deux.

Appelons U l'ensemble des éléments x tels que $x^{2^a} = 1$. Cet ensemble forme l'arborescence de 1, avec ses 2^a éléments et c'est un sous-groupe de $U(p)$. Plus précisément, cette structure établit une hiérarchie des ordres[6] des éléments. Ainsi 1 a pour ordre 1, –1 a pour ordre 2 à l'étage 1. A leur tour, les éléments de l'étage 2 ont pour ordre 4, ceux de l'étage 3 ont pour ordre 8, et ainsi de suite jusqu'à l'étage final a où les 2^{a-1} feuilles ont ont pour ordre 2^a.[7] Autrement dit, les puissances successives d'un élément feuille décrivent les 2^a éléments du groupe U.[8] Chaque feuille de l'arbre est un générateur du groupe U. On dit que ce groupe est cyclique. Comme sous-groupe de $U(p)$, U est l'ensemble des éléments de $U(p)$ dont l'ordre est une puissance de 2.

1.2. Points cycliques, formant l'ensemble *C*

Par points cycliques, nous entendons les points dont la trajectoire retombe sur eux. Nous avons déja vu que 1 est un point cyclique, avec un cycle de longueur 1. On va montrer la propriété suivante :

* **Avec $p - 1 = 2^a \times h$, il y a h points cycliques. Tous les éléments d'un même cycle ont le même ordre. Les longueurs des cycles sont celles des cycles de la permutation créée par $x \rightarrow 2x$ dans Z / hZ. Chacun des points cycliques est la racine d'un arbre binaire totalement équilibré avec un tronc, ayant a étages, à l'image de l'arbre de 1.**

Outre le point 1, il peut exister d'autres points cycliques avec des périodes de longueur > 1. Pour cela considérons l'équation $c^h = 1$ avec h impair tel que $p - 1 = 2^a \times h$. Comme h divise $p - 1$, elle admet exactement h solutions, et celles-ci forment un sous-groupe C de $U(p)$, cyclique [9] comme $U(p)$, ce qui signifie qu'il possède des générateurs, au nombre de $\varphi(h)$ [10].

Par exemple pour $p = 31$, $p - 1 = 30 = 2 \times 15$, les éléments c tels que $c^{15} = 1$ [31] sont au nombre de 15 et le groupe C possède $\varphi(15) = \varphi(3 \times 5) = 2 \times 4 = 8$ générateurs. Par exemple un générateur est $g = 7$, et ses puissances successives g^k décrivent les 15 éléments du groupe C.

Si g est un des générateurs, formons la suite g, g^2, g^4, g^8, g^{16},.. où les exposants doublent à chaque étape (ces puissances de g ne décrivent pas forcément le sous-groupe, mais on peut recommencer à partir d'une autre puissance de g, etc.). Comme la fonction $x \rightarrow 2x$ [h] est une bijection, 2 étant premier avec h impair, elle provoque une permutation dans Z / hZ, et celle-ci se décompose en cycles. Si l'on considère que pour chacun de ces cycles il s'agit d'exposants de g, ceux-ci doublent à chaque étape et finissent par boucler, et cela correspond bien à la récurrence $x \rightarrow x^2$. Cela prouve que les h

---

[6] L'ordre d'un élément a est le plus petit entier positif h tel que $a^h = 1$. On dit aussi que a appartient à l'exposant h.

[7] On dispose de la propriété suivante sur les ordres :
Si a pour ordre h dans $U(p)$, alors a^k a pour ordre h / $pgcd(h, k)$.

Notamment si a a pour ordre h, a^2 a pour ordre h / $pgcd(2, h)$. Si h est impair, a^2 a pour ordre h, le même que celui de a, et si h est pair, l'ordre de a^2 est h /2. Lorsqu'un élément possède deux racines carrées, celles-ci ont soit le même ordre soit un ordre double. Dans l'arborescence de 1, par récurrence évidente, les éléments d'un étage ayant tous un même ordre pair, ceux de l'étage supérieur ne peuvent avoir qu'un ordre pair, et celui-ci est alors le double de celui de l'étage inférieur.

[8] Par exemple pour $p = 13$, prenons la feuille où se trouve 5. On a $5^2 = 12$, $5^3 = 8$ et $5^4 = 1$. Le nombre 5 a pour ordre 4 et les puissances de 5 décrivent U.

[9] Ce sous-groupe est cyclique car il a des générateurs d'ordre h. En effet, si l'ordre maximal h' des éléments de ce groupe était inférieur à h, il engendrerait seulement h' éléments au lieu de h.

[10] φ est la fonction d'Euler vérifiant $\varphi(p) = p - 1$, $\varphi(p^n) = p^{n-1}(p - 1)$ pour p premier impair ou pair, et $\varphi(m\,m') = \varphi(m)\,\varphi(m')$ lorsque m et m' sont premiers entre eux.

points c du sous-groupe vérifiant $c^h = 1$ [p] sont des points cycliques qui finissent tous par retomber sur eux-mêmes : $x \rightarrow x^2 \rightarrow x^4 \rightarrow ... \rightarrow x$. Parmi eux on retrouve notamment le cycle formé par 1. La progression géométrique de raison 2 modulo h permet d'obtenir les longueurs de tous les cycles.

Reprenons l'exemple de p = 31. La permutation s'écrit :

$$\begin{pmatrix} 0\,1\,2\,3\,4\,5\,6\,7\,8\,9\,10\,11\,12\,13\,14 \\ 0\,2\,4\,6\,8\,10\,12\,14\,1\,3\,5\,7\,9\,11\,13 \end{pmatrix}$$ et elle se décompose en cycles :

(0) (1 2 4 8) (3 6 12 9) (5 10) (7 14 13 11). En considérant ces nombres comme les exposants d'un générateur, par exemple g = 7, on trouve les cycles associés à la récurrence $x \rightarrow x^2$ [31] de la *figure 3*. Remarquons que les $\varphi(15) = 8$ générateurs forment deux cycles de longueur 4. En effet la progression géométrique de raison 2 modulo 15 donne (1 2 4 8) de longueur 4.

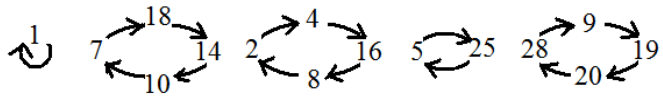


Figure 3 : Les points cycliques formant le groupe C pour p = 31.

D'autre part tous les éléments d'un même cycle ont le même ordre,[11] mais il peut y avoir plusieurs cycles pour des éléments de même ordre, comme dans l'exemple précédent.

Nous avons ainsi trouvé h points cycliques (en comptant 1). Il reste à prouver qu'il n'y en a pas d'autres. Soit c un point cyclique autre que 1 et qui vérifie c^h =1. L'équation $x^{2^a} = c$ admet 2^a solutions puisque $c^{\frac{p-1}{2^a}} = c^h = 1$ [p].

Plus précisément, l'équation $x^2 = c$ admet deux solutions, et ainsi de suite jusqu'aux 2^a solutions de $x^{2^a} = c$. Sur les deux solutions de l'équation $x^2 = c$, une et une seule est extérieure au cycle, notons-la c_1.

Lorsque p est de la forme $4k + 3$, on en reste là, et chaque élément c d'un cycle possède une excroissance à un seul étage.

Prenons maintenant p de la forme $4k + 1$. A son tour l'équation $x^4 = c$ admet 4 solutions, dont deux et deux seulement ont leur carré qui vaut c (*figure 4*). Par le même raisonnement que celui fait pour l'arborescence de 1, l'arborescence issue de c est un arbre binaire à partir de c_1, avec un élément au premier étage, deux éléments au deuxième, ..., et 2^{a-1} éléments à son dernier étage a. Cela fait un total de 2^a éléments formant l'arborescence issue de c. Il en est ainsi pour chacun des éléments c vérifiant $c^h = 1$ Cela fait un total de $2^a \times h$ éléments, ce qui épuise tous les éléments de $U(p)$. Cela prouve qu'il n'y a pas d'autres points cycliques que ceux vérifiant $c^h = 1$. Par la même occasion, on a prouvé que toutes les arborescences ont la même hauteur a et qu'elles sont des copies de l'arborescence de 1.

[11] Soit b un élément d'ordre h' (impair) dans un cycle. Les autres éléments du cycle sont de la forme b^{2^j} et ils ont pour ordre $h' / pgcd\,(h', 2^j) = h'$ aussi.

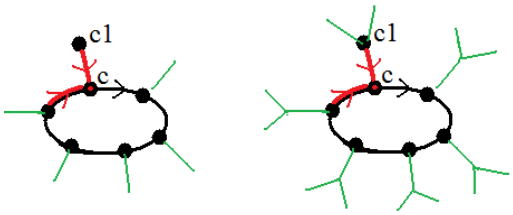


Figure 4 : *A gauche* un cycle pour $p = 4k + 3$, à droite pour $p = 4k + 1$, avec des indications sur la forme des arborescences

Passons maintenant au classement des éléments suivant leur ordre. Les éléments cycliques de C tels que $c^h = 1$ [p] ont tous un ordre impair qui divise h, et ce sont les seuls puisque leurs ordres sont tous les diviseurs impairs de $p - 1$. Tous les autres éléments de $U(p)$ ont un ordre pair, sur les arborescences au-delà de leurs racines cycliques. Prenons un cycle où les éléments on un ordre h'. A l'étage 1 se trouvent les éléments d'ordre 2 h' (on a vu qu'un élément cyclique a deux racines carrées, dont l'une est sur le cycle, avec l'ordre h', et l'autre à l'extérieur d'ordre 2h'. Puis à l'étage 2 on a deux racines carrées pour chaque élément de l'étage 1, toutes d'ordre 4h'. Et ainsi de suite, comme pour l'arborescence de 1. Ainsi, d'un étage au suivant les ordres doublent, et les éléments d'ordre maximal comme multiples de h' constituent les feuilles des arborescences. Notamment les générateurs sont les feuilles des arborescences issues des éléments cycliques d'ordre h (cf. *figure 5* pour l'exemple $p = 61$). D'où ce résultat :

Sur les arborescences, les ordres des éléments de $U(p)$ doublent lors du passage d'un étage à l'étage supérieur. Les générateurs de $U(p)$ sont les feuilles des arborescences issues des cycles où les éléments ont pour ordre h.

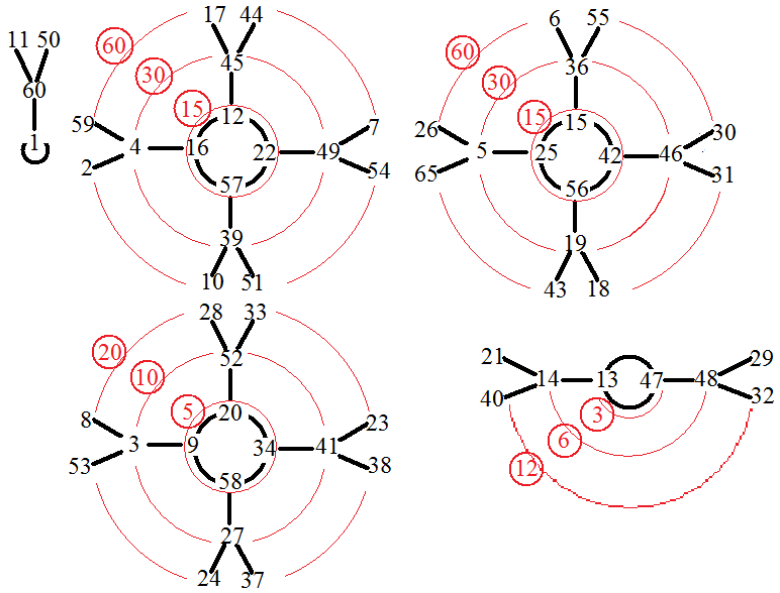


Figure 5 : Composantes connexes du graphe des trajectoires pour $p = 61$, avec en rouge les ordres des éléments.

Algorithme de recherche des générateurs de U(p) lorsque p = 4k + 3

Dans ce cas, les arborescences sont de hauteur 1.Sur le ou les cycles où les éléments ont pour ordre h, on a vu qu'un élément a deux racines carrées dont l'une est son antécédent sur le cycle, et l'autre est un générateur situé au premier étage. Tous les générateurs s'obtiennent en faisant la soustraction : p – élément cyclique d'ordre h.

Par exemple pour $p = 11$, $p - 1 = 10 = 2\times5$, il y a 5 éléments cycliques en comptant 1, et $\varphi(5) = 4$ éléments cycliques d'ordre 5. Ceux-ci forment un seul cycle, car l'ordre de 2 modulo 5 est justement 4. Il s'agit de 4, 5, 3, 9. Les générateurs sont donc 7, 8, 6, 2 (*figure 6*).

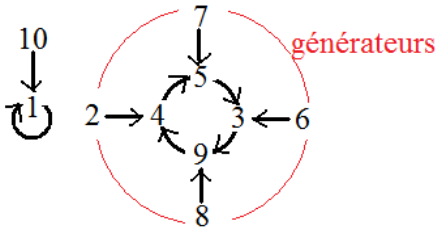


Figure 6 : Composantes du graphe pour $p = 11$, où sont indiqués les générateurs de $U(11)$.

Cela donne un algorithme pour déterminer les générateurs. Par essais au hasard, on recherche un élément cyclique d'ordre h. Par la même occasion on a tous les éléments x de ce cycle. On en déduit les générateurs en faisant $p - x$. Puis l'on recommence s'il existe d'autres cycles d'ordre h.

1.3 Décomposition de $U(p) = U \otimes C$

- **Avec $p - 1 = 2^a \times h$, l'arborescence de 1 forme un groupe cyclique U d'ordre 2^a, et l'ensemble des éléments cycliques est un groupe cyclique C d'ordre h. Tout élément z de $U(p)$ se décompose de façon unique en produit d'un élément x de U et d'un élément y de C : $z = x\,y$. Plus précisément z et x sont sur un même étage dans leurs arborescences respectives, et z et y sont dans la même composante connexe, celle du cycle où se trouve y.**

Quand x décrit U et que y décrit C, cela fait $2^a \times h = p - 1$ produits xy. Tous ces produits sont distincts : si l'on avait deux produits égaux, $x\,y = x'\,y'$, on aurait $x\,x'^{-1} = y\,y'^{-1}$, un élément de U serait égal à un élément de C, ce qui n'est possible que pour 1, car les éléments de U ont un ordre qui divise 2^a, c'est-à-dire pair sauf pour 1, et ceux de C ont un ordre qui divise h impair, d'où $C \cap U = \{1\}$. Ainsi $x\,x'^{-1} = 1$, $x = x'$ et $y = y'$. Les produits $x\,y$ décrivent $U(p)$ et sont en correspondance biunivoque avec les éléments de $U(p)$.[12]

Pour $z = x\,y$ avec x dans U et y dans C, on a :

$$z^{2^a} = x^{2^a} y^{2^a} = y^{2^a}$$

ce qui prouve que z et y sont dans la même composante connexe. Plus précisément, si z est dans l'arborescence de l'élément cyclique c, à l'étage e, y s'obtient en partant de c et en faisant marche arrière de e crans dans le cycle. Notons e' l'étage où se trouve x. Alors $z^{2^e} = x^{2^e} y^{2^e} = y^{2^e}$ puisque z^{2^e} doit être un point cyclique. Comme $x^{2^e} = 1$, il s'ensuit que $e' \leq e$. Si l'on avait $e' < e$, on aurait $z^{2^{e-1}} = x^{2^{e-1}} y^{2^{e-1}} = y^{2^{e-1}}$ et $z^{2^{e-1}}$ serait cyclique, ce qui n'est pas le cas. D'où $e' = e$.

Exemple pour p = 61

Prenons l'élément $z = 8$. En s'aidant de la *figure 5* où sont représentées les trajectoires, on constate que z est à l'étage 2, et qu'il dans l'arborescence de l'élément cyclique 9. Avec $z = x\,y$, $z^4 = x^4\,y^4 = y^4$, on trouve y en faisant marche arrière de deux crans à partir de 9, soit $y = 34$. A son tour, x est à l'étage 2 sur l'arborescence de 1. On trouve $x = 11$. Finalement $8 = 11 \times 34$ [61]. On trouverait de même $3 = 60 \times 58$.

- **Pour tout z de $U(p)$, z^h appartient à U, z et z^h étant dans le même étage.**

$z^h = (xy)^h = x^h\ y^h = x^h$, d'où z^h est dans U. Ainsi le fait d'élever un élément à la puissance h le transporte dans l'arbre U. Notons e l'étage de z et e' l'étage de z^h .

[12] Finalement le groupe multiplicatif $U(p)$ est isomorphe au groupe additif $Z/2^aZ \times Z/hZ$.

$z^{2^e} = c$ (élément cyclique), d'où $z^{2^e h} = (z^h)^{2^e} = 1$, l'ordre de z^h divise 2^e, d'où $e' \leq e$. D'autre part $(z^h)^{2^{e'}} = (z^{2^{e'}})^h = 1$, $z^{2^{e'}}$ appartient à un cycle d'où $e' \geq e$. Finalement $e' = e$.

- ***z* a pour ordre le produit des ordres de *x* et de *y*.**

Comme $z = x\,y$, z a comme ordre le ppmc des ordres de x et y, c'est-à-dire ici le produit de leurs ordres. Comme les ordres doublent lors du passage d'un étage de l'arborescence de 1 au suivant, on retrouve le fait qu'il en est de même sur toutes les arborescences. Les éléments y des cycles ont un ordre qui divise h, et pour chaque diviseur h' de h, il y a $\varphi(h')$ éléments d'ordre h'. Remarquons aussi que les feuilles des arborescences sont évidemment les non-carrés, vérifiant $z^{(p-1)/2} = -1$, avec les générateurs parmi eux. Tous les autres éléments sont les carrés, vérifiant $z^{(p-1)/2} = 1$. Comme z et z^h sont au même étage, le fait que z soit un carré équivaut au fait que z^h est un carré.

1.4. Programmation

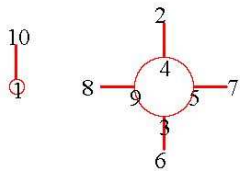
1.4.1. Tracé des trajectoires avec leurs composantes connexes lorsque p est de la forme 4k + 3

Il s'agit du cas le plus simple puisque les arborescences n'ont qu'un étage. Nous ne donnons ci-dessous que le programme en mode texte. Sa conversion en mode dessin donne des résultats comme ceux de la *figure 7*. On commence par l'arborescence de 1, dans le cycle numéro 1. Puis on prend des éléments au hasard *px*[1] qui n'ont pas déjà été traités, ceux tels que *dejavu*[*px*[1]] est à 0. A chaque fois, on détermine la trajectoire de *px*[1], en mettant ses éléments à *dejavu*[] = 1, en enregistrant la longueur de la période correspondante *lperiode*[*ncycle*] du cycle numéro *ncycle*, et l'indice d'entrée *ie* qui vaut soit 1 soit 2. Dans chacun de ces deux cas possibles, on détermine aussi les feuilles associées au cycle, que l'on met à *dejavu*[] = 1. On répète cela jusqu'à ce que tous les éléments de $U(p)$ soient placés dans les arborescences, grâce à la variable *nb* qui compte le nombre des éléments placés dans les arrborescences à chaque étape, finissant par atteindre $p - 1$.

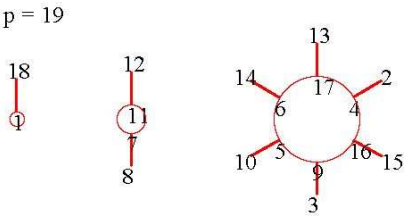
```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#define p 43     /* autres tests : 32911  32939 32887*/
void f(int jj);
int px[10000],lperiode[600],nb,cycle[600][10000],ncycle;
int dejavu[p+1],feuille[600][10000],nb;

int main()
{ int i,j,ie;
  srand(time(NULL));
  printf("%d \n\n",p); if (p%4 ==1) exit(0);
  cycle[1][1]=1; ncycle=1;  /* arborescence de 1 , dans cycle[1] */
  lperiode[1]=1; dejavu[1]=1; dejavu[p-1]=1; nb=2; feuille[1][1]=p-1;
  printf(" cycle 1 : 1  feuille %d ",feuille[1][1]);
  ncycle++;
  do                        /* autres arborescences */
    { do px[1]=rand()%(p-3)+2; while(dejavu[px[1]]==1);  /* point de départ pris au hasard */
      f(1); i=1;
      for(;;)
        { i++;
          f(2*i-2); f(2*i-1);
          if (px[i]==px[2*i])
              { lperiode[ncycle]=i;    break;}
        }
      if (px[1]==p - px[1+lperiode[ncycle]]) ie=2; else ie=1; /* deux cas selon l'indice d'entrée ie */
      if (ie==1)
        {  printf("\n\n CYCLE %d : ",ncycle);
```

```
            for(j=1;j<=lperiode[ncycle];j++)
              { cycle[ncycle][j]=px[j]; dejavu[px[j]]=1;  /* les points du cycle */
                printf(" %d ", cycle[ncycle][j]);
              }
            for(j=1;j<=lperiode[ncycle];j++)
               { feuille[ncycle][j]=p - px[j-1+lperiode[ncycle]]; dejavu[feuille[ncycle][j]]=1;}
            printf( "    feuilles : ");  /* feuilles associées au cycle */
            for(j=1;j<=lperiode[ncycle];j++) printf( " %d",feuille[ncycle][j]);
            nb+=2*lperiode[ncycle];
            ncycle++;
          }
        else if (ie==2)
          { printf("\n\n CYCLE %d : ",ncycle);
            for(j=2;j<=lperiode[ncycle]+1;j++)
               { cycle[ncycle][j-1]=px[j]; dejavu[px[j]]=1;
                 printf(" %d ", cycle[ncycle][j-1]);
               }
             for(j=2;j<=lperiode[ncycle]+1;j++)
               { feuille[ncycle][j-1]=p - px[j-1+lperiode[ncycle]]; dejavu[feuille[ncycle][j-1]]=1;}
             printf( "    feuilles : ");
            for(j=1;j<=lperiode[ncycle];j++) printf( " %d",feuille[ncycle][j]);
            nb+=2*lperiode[ncycle];
            ncycle++;
          }
      }
  while(nb<p-1);  /* on recommence jusqu'à ce que tous les éléments soient traités */

  printf("\n\n %d CYCLES :",ncycle-1);
  for(j=1;j<ncycle;j++) printf(" (%d : %d)", j, lperiode[j]); /* affichage de la longueur de chaque cycle */
  getchar();return 0;
}

void f(int jj)   /* successeur d'un élément jj */
{ px[jj+1]=px[jj]*px[jj];
  while(px[jj+1]<0) px[jj+1]+=p;   while (px[jj+1]>=p) px[jj+1]-=p;
 }
```

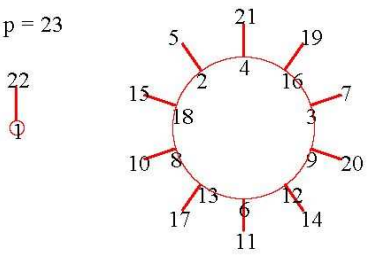
p = 11



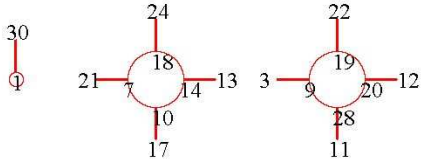
p = 19

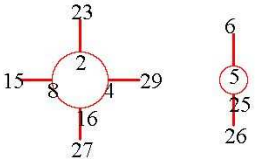


p = 23



p = 31





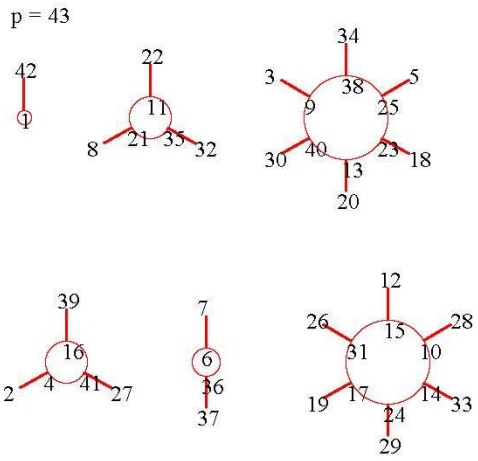


Figure 7 : Exemples de trajectoires lorsque $p = 4k + 3$

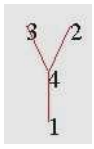
1.4.2. Tracé des composantes connexes lorsque p est de la forme 4k + 1

Les arborescences issues des cycles ont toutes le même nombre d'étages, au moins égal à 2. Le programme qui suit donne les résultats de la *figure 8*.

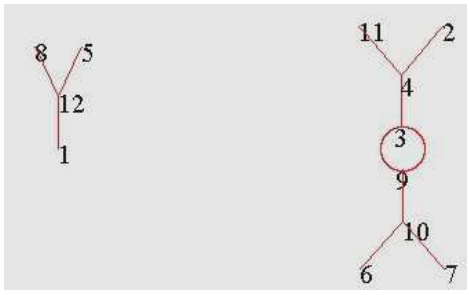
```
#include <SDL/SDL.h>
#include <SDL/SDL_ttf.h>
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <math.h>
#define p 61
#define L 37
void afficher(int i);
void arbre1(int i,int e);
void arbre (int q,int e,float angle);
void dessincycle(int nc);
void pause(void);
void putpixel(int xe, int ye, Uint32 c);
Uint32 getpixel(int xe, int ye);
void circle( int xo, int yo, int R, Uint32 couleur);
void line(int x0,int y0, int x1,int y1, Uint32 c);
SDL_Surface * screen;
Uint32 white,black,red;
SDL_Surface *texte;  SDL_Rect position;TTF_Font *police=NULL;char chiffre[2000];
SDL_Color couleurnoire={0,0,0};
int x,succ[p+1],feuille[p+1],lperiode[100],nb,px[p+1],cycle[100][100];
int feuil[p+1],nbfeuilles, nbcycles,dejavucycle[80],flag,nbetages;
int antec1[100],antec2[100],dejavufeuille[100],longueurcycle[20];
int xorig,yorig,R[20];
int xx[p+1],yy[p+1];

int main(int argc, char ** argv)
{ int i,j,k,ie;
   SDL_Init(SDL_INIT_VIDEO);
   screen=SDL_SetVideoMode(800,600,32, SDL_HWSURFACE|SDL_DOUBLEBUF);
   white=SDL_MapRGB(screen->format,255,255,255);
   black=SDL_MapRGB(screen->format,0,0,0);
   red=SDL_MapRGB(screen->format,255,0,0);
   SDL_FillRect(screen,0,white);
   TTF_Init(); police=TTF_OpenFont("times.ttf",20);

for(x=1;x<p;x++) feuille[x]=1;
```

```
for(x=1;x<p;x++) {succ[x]=x*x; while(succ[x]>=p) succ[x]-=p;
                          feuille[succ[x]]=0;  antec1[succ[x]]=x; antec2[succ[x]]=p-x;
                         }
  /* les feuilles ne sont pas les successeurs d'un élément */
i=0;
for(x=1;x<p;x++) if (feuille[x]==1) {feuil[i++] =x; }
nbfeuilles=i;
x=p-1; ie=1;
while (x!=0)
    {ie++; x=antec1[x];}
nbetages=ie-1;
/* arbre de 1 */
xorig=50+30*nbetages;yorig=100+40*nbetages;
xx[1]=xorig;yy[1]=yorig; afficher(1);
xx[p-1]=xorig;yy[p-1]=yorig-L;afficher(p-1);
line(xx[1],yy[1],xx[p-1],yy[p-1],red);
arbre1(p-1,nbetages-1);
dejavucycle[1]=1;
/* Autres cycles et leurs arborescences */
nbcycles=0;
for(k=0;k<nbfeuilles/2;k++)
   {  x=feuil[k]; if (dejavufeuille[x]==0)
         { px[0]=x;
           i=0; flag=0;
           for(;;)
              { px[i+1]=px[i]*px[i]; while(px[i+1]>=p) px[i+1] -= p;
                for(j=i; j>=0; j--) if (px[i+1]==px[j])
                    {lperiode[px[0]]=i+1-j;flag=1; break;}
                if (flag==0) i++;
                else break;
              }
            if (dejavucycle[px[ie]]==0)
              { nbcycles++;
                for(i=ie;i< ie + lperiode[x];i++) cycle[nbcycles][i-ie]=px[i];
                longueurcycle[nbcycles]=lperiode[x];
                xorig+=250; if (xorig>700) { xorig=150;yorig+=230;}
                dessincycle(nbcycles);
                for(i=0;i<longueurcycle[nbcycles];i++)
                      {dejavucycle[cycle[nbcycles][i]]=1;  }
              }
         }
   }
  SDL_Flip(screen);pause(); return 0;
}

void afficher(int i)
   { sprintf( chiffre,"%d",i);
     texte=TTF_RenderText_Solid(police,chiffre,couleurnoire);
     position.x=xx[i]; position.y=yy[i]-10;
     SDL_BlitSurface(texte,NULL,screen,&position);
   }
 void arbre1 (int i, int e)
  { if (e!=0)
       { xx[antec1[i]]=xx[i]-10-7*e*e;  yy[antec1[i]]=yy[i]-L;
         xx[antec2[i]]=xx[i]+10+7*e*e;  yy[antec2[i]]=yy[i]-L;
         afficher(antec1[i]);afficher(antec2[i]);
         line(xx[i],yy[i],xx[antec1[i]],yy[antec1[i]],red);
         line(xx[i],yy[i],xx[antec2[i]],yy[antec2[i]],red);
         if (e==1) {dejavufeuille[antec1[i]]=1;dejavufeuille[antec2[i]]=1;}
         arbre1(antec1[i],e-1); arbre1(antec2[i],e-1);
```

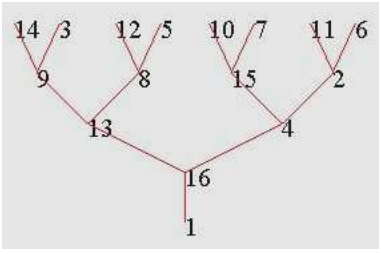
```
      }
  }
void arbre(int q, int e,float angle)
  { float angle1,angle2;
     if (e!=0)
       { angle1=angle +0.7; angle2=angle-0.7;
          xx[antec1[q]]=xx[q]+(L+10)*cos(angle1);  yy[antec1[q]]=yy[q]-(L+10)*sin(angle1);
         xx[antec2[q]]=xx[q]+(L+10)*cos(angle2);  yy[antec2[q]]=yy[q]-(L+10)*sin(angle2);
         afficher(antec1[q]);afficher(antec2[q]);
         line(xx[q],yy[q],xx[antec1[q]],yy[antec1[q]],red);
         line(xx[q],yy[q],xx[antec2[q]],yy[antec2[q]],red);
         if (e==1) {dejavufeuille[antec1[q]]=1;dejavufeuille[antec2[q]]=1;}
         arbre(antec1[q],e-1,angle1); arbre(antec2[q],e-1,angle2);
       }
  }
void dessincycle(int nc)
{  float angle; int j,lc,ant1;
   lc=longueurcycle[nc];
   R[nc]=8*lc;
   for(j=0;j<lc;j++)
   { angle= -2*M_PI/ (float)longueurcycle[nc]*(j-1)+M_PI/2.;
     circle(xorig,yorig, R[nc],red);
     xx[j]=xorig+R[nc]*cos(angle);  yy[j]=yorig-R[nc]*sin(angle);
     sprintf( chiffre,"%d",cycle[nc][j]);
     texte=TTF_RenderText_Solid(police,chiffre,couleurnoire);
     position.x=xx[j]-5; position.y=yy[j]-6;
     SDL_BlitSurface(texte,NULL,screen,&position);
     ant1=p-cycle[nc][(j-1+lc)%lc];
     xx[ant1]=xorig+(R[nc]+L)*cos(angle);    yy[ant1]=yorig-(R[nc]+L)*sin(angle);
     line(xx[j],yy[j], xx[ant1],yy[ant1],red);
     sprintf( chiffre,"%d",ant1);
     texte=TTF_RenderText_Solid(police,chiffre,couleurnoire);
     position.x=xx[ant1]; position.y=yy[ant1]-6;
     SDL_BlitSurface(texte,NULL,screen,&position);
     arbre(ant1,nbetages-1,angle);
   }
  SDL_Flip(screen); pause();
}
```



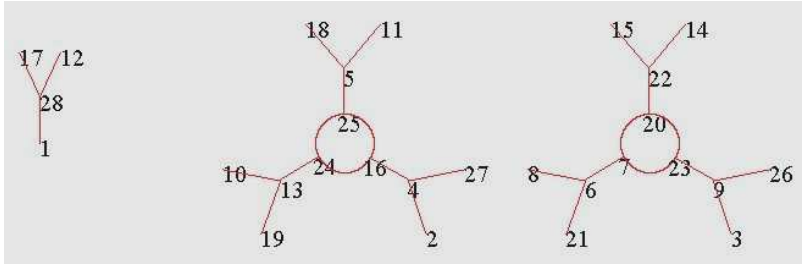
p = 5



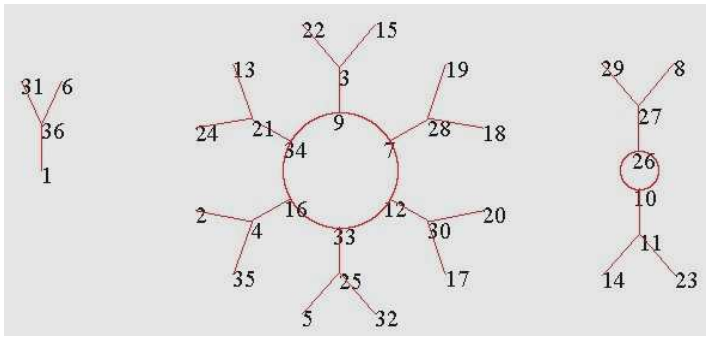
p =13



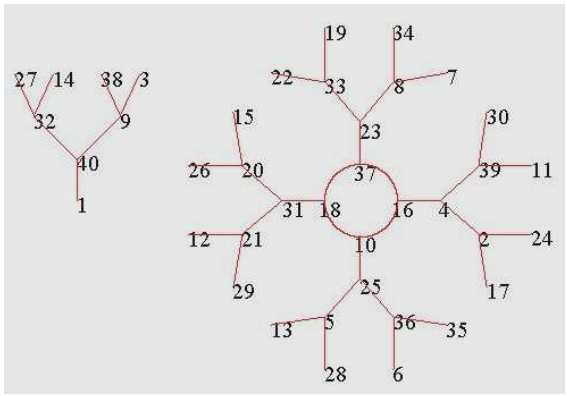
p = 17



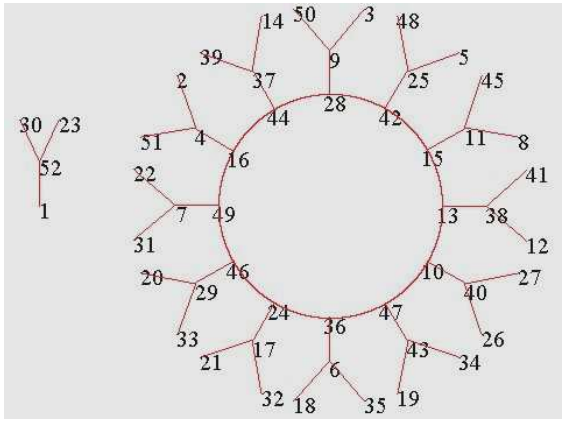
p = 29



$p = 37$



$p = 41$



$p = 53$

Figure 8 : Exemples de trajectoires lorsque $p = 4k + 1$

1.4.3. Recherche des racines carrées

Si p est de la forme $4k + 3$, les éléments des cycles sont les carrés, les feuilles des arborescences, toutes à l'étage 1, sont les non-carrés. Une racine carrée r d'un élément cyclique z est son prédécesseur sur le cycle correspondant et l'autre est la feuille associée $p - r$. Il suffit de déterminer la trajectoire de z par élévations au carré, les éléments étant mis dans le tableau *px*[] à partir de l'indice 1, et d'en déduire la longueur *lperiode* de la période et l'indice d'entrée *ie* dans le cycle. Si *ie* vaut 2, z n'a pas de racine carrée, et si *ie* vaut 1, on a facilement les deux racines carrées.

Si p est de la forme $4k + 1$, on commence par écrire $p - 1 = 2^a h$ avec h impair, avec a donnant la hauteur des arborescences. Puis on fait la trajectoire de z comme précédemment, pour avoir la longueur *lperiode* de la période la plus courte, et l'indice d'entrée *ie* dans la boucle, ce qui permet aussi de connaître à quel étage se trouve z. Si z est à l'étage a, z est un non-carré. Sinon c'est un carré, dont on va chercher les racines carrées. On utilise alors le fait que tout élément z de $U(p)$ s'écrit de façon unique $z = xy$.

Commençons par chercher y. Avec z à l'étage e, y est dans le cycle associé et il suffit de faire e pas en marche arrière sur le cycle en partant du point d'entrée d'indice *ie* Une fois trouvé y, il suffit de reculer d'un cran dans le cycle pour avoir une racine carrée *racy* de y.

Passons à x qui est dans l'arborescence de 1, au même étage que z. Pour trouver une racine carrée de x, on va utiliser son inverse $x^{-1} = y\, z^{-1}$, ce qui nécessite le calcul de l'inverse de z. Cela étant fait, on doit connaître une feuille de l'arborescence de 1. Pour cela on procède par essais au hasard. Il y a en gros une chance sur deux de tomber sur une feuille f en prenant un élément au hasard de $U(p)$ (noté *px1*[1] dans le programme), entre 2 et $p - 2$. Grâce à sa trajectoire, on reconnaît une feuille si elle est à l'étage a, et si cette feuille *px1*[1] n'est pas dans l'arborescence de 1, on la remplace par *px1*[1] élevé à la puissance h (on a vu qu'un élément z et sa puissance z^h sont au même étage, avec z^h dans l'arborescence de 1). La trajectoire de cette feuille f étant connue, on prend sur cette trajectoire

l'élément f_e situé à l'étage e, étage où se trouve z. On sait que x est au même étage e, et son inverse x^{-1} aussi, puisque x et son inverse ont le même ordre. Formons le produit $u = x^{-1} f_e$. Comme ces deux éléments sont sur l'arborescence de 1, leur produit u est aussi sur l'arborescence de 1, et comme ils ont le même ordre 2^e, leur produit u admet un ordre strictement inférieur, il est à un étage e' avec $e' < e$. Si l'on tombe sur $u = 1$, c'est fini, et l'on a non seulement $x = f_e$, et aussi une racine carrée de x égale à f_{e+1}. Sinon on recommence en prenant un nouvel u égal à l'ancien u multiplié par l'élément $f_{e'}$ de la trajectoire. Ce nouvel u est à un étage $e'' < e'$. On continue ainsi jusqu'à avoir un nouvel u égal à 1. On en déduit que $x = f_e f_{e'} f_{e''}$... et en faisant un décalage d'un cran on trouve une racine carrée de x, soit $f_{e+1} f_{e'+1} f_{e''+1}$ ...

Ayant obtenu une racine carrée de y et une racine carrée de x, leur produit donne une racine carrée de z. Illustrons cette méthode dans l'exemple suivant.

Exemple : Racines carrées de z = 328 pour p = 769

- On constate d'abord que $p = 2^8 \times 3$. Les feuilles sont à l'étage 8.
- On détermine la trajectoire de z par élévations au carré :

  1 2 3 4 5 6 7 8 9

  328 → 693 → 393 → 649 → 558 → 688 → 409 → 408 → 360

La période est 2 et l'indice d'entrée est 8 : z est à l'étage 7, ce n'est pas une feuille, c'est un carré.

- Pour avoir y, on tourne de 7 crans en marche arrière à partir du point d'entrée 408 dans la période, soit $y =$ 360, et une racine carrée de y est à 8 crans, soit 408.
- On détermine l'inverse de x, soit $x^{-1} = y\, z^{-1} = 470$.
- On cherche par essais une feuille f d'une arborescence, située à l'étage 8. Si l'on ne tombe pas sur une feuille de l'arborescence de 1, on remplace f par $f^h = f^3$ pour être dans l'arborescence de 1. On trouve la feuille $f = 668$ avec sa trajectoire :

indices : 1 2 3 4 5 6 7 8 9

668 → 204 → 90 → 410 → 458 → 596 → 707 → 768 → 1

étages : 8 7 6 5 4 3 2 1

Comme x, x^{-1} est à l'étage 7. On forme :

- $u = x^{-1} \times$ élément à l'étage 7 dans l'arborescence de 1 = $470 \times$ **204** = 524. On constate que 524 est à l'étage 6.
- Puis $u = u \times$élément à l'étage 7 = $524 \times$ **90** = 251, qui est à l'étage 5.
- Puis $u = u \times$élément à l'étage 5 = $251 \times$ **410** = 633. On constate que 633 est à l'étage 4.
- Puis $u = u \times$élément à l'étage 4 = $633 \times$ **458** = 1. C'est fini.
- On en déduit $x = 204 \times 90 \times 410 \times 458$ et par décalage d'un cran en marche arrière, une racine carrée de x est $668 \times 204 \times 90 \times 410 = 399$.

Par multiplication d'une racine carrée de y et d'une racine carrée de x, on trouve une racine carrée de z, soit 533 dans le cas présent, l'autre racine carrée de z étant 236.

Le programme découle de ce qui précède.

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#define p  769
#define z 340
void f(int jj);void f1(int jj);
int inversemodulo(int a, int m);
int px[10000],lperiode,ie,px1[10000];
```

```
int main()
{ int i,j,k,ie,rac1,rac2,verif,nbcrans,racy,y,invz,invx,etagez,u,a,h;
  int pp,cumul,racx,indice[100],uu;
 srand(time(NULL));
 /********* trajectoire de z **********/
 px[1]=z;  f(1); i=1;
 for(;;)
   { i++;  f(2*i-2); f(2*i-1);  if (px[i]==px[2*i])  { lperiode=i;    break;}
   }  /* lperiode est la plus courte période si p = 4k + 3,
         mais pas forcément la plus courte pour p = 4k+1 */
 if (p%4==3)   /* p de la forme 4k+3 */
   { if (px[1]==p - px[1+lperiode]) ie=2; else ie=1;
     if (ie==2)  printf("\n\n %d est un non-carre",z);
     else if (ie==1)
        { rac1=px[lperiode]; rac2=p-rac1;
          printf("\n\n %d a deux racines carrees : %d  et %d ",z,rac1,rac2);
        }
   }
 else if(p%4==1)   /* p de la forme 4k+1 */
   {  pp= p-1; a = 0;  while(pp%2==0) { pp=pp/2;a++;}     h=pp;
      for(j=i+1;j<=2*i;j++) if ( px[i]==px[j] ) {lperiode= j-i;  break;} /* lperiode est la plus courte */
      while(px[i]==px[i+lperiode]) i--;
      ie = i+1;
      if (ie-1==a) {printf("\n\n %d non carre",z); getchar();exit(0);}
      nbcrans=lperiode-ie;
      while(nbcrans<0) nbcrans+=lperiode;
      racy=px[ie+nbcrans];y=px[ie+(1+nbcrans)%lperiode];
      invz=inversemodulo(z,p);
      invx=y*invz; while (invx>=p) invx-=p;
      etagez=ie-1;
      /******** feuille de 1 **********/
      do
         { px1[1]=rand()%(p-3)+2; f1(1);  i=1; /* recherche d'une feuille au hasard */
           for(;;)
             { i++;  f1(2*i-2); f1(2*i-1); if (px1[i]==px1[2*i])  { lperiode=i;    break;}
             }
           for(j=i+1;j<=2*i;j++) if ( px1[i]==px1[j] ) {lperiode= j-i;  break;}
            while(px1[i]==px1[i+lperiode]) i--;
            ie = i+1;
         }
      while(ie!=(a+1));
      if (lperiode!=1)  /* si la feuille n'est pas dans l'arbre de 1, on l'élève à la puissance h */
        { cumul=px1[1];
          for(j=2;j<=h;j++)  { cumul=cumul*px1[1]; while (cumul>=p) cumul-=p;}
          px1[1]=cumul; f1(1);  i=1;
          for(;;)  /* trajectoire de la feuille f de l'arborescence de 1 */
             { i++;   f1(2*i-2); f1(2*i-1);  if (px1[i]==px1[2*i])  { lperiode=i;    break;}
             }
          for(j=i+1;j<=2*i;j++) if ( px1[i]==px1[j] ) {lperiode= j-i;  break;}
           while(px1[i]==px1[i+lperiode]) i--;
           ie = i+1;
        }
      indice[0]=ie-etagez; /* indice correspondant à l'étage de z */
      u=px1[indice[0]]*invx; while (u>=p) u-=p;  /* calcul du premier u */
      if (u==1) racx=px1[indice[0]-1];
      else  /* calcul des autres u et de la racine carrée racx de x */
         { j=0; racx=px1[indice[0]-1];
           while(u!=1)
               { j++;
                 uu=u; k=1;
```

```
                do { uu = uu*uu; while(uu>=p) uu-=p; k++;}
                while(uu!=1);
                indice[j]=ie-k+1;
                u=u*px1[indice[j]]; while(u>=p) u-=p;
                racx*=px1[indice[j]-1];while(racx>=p) racx-=p;
               }
           }
        rac1=racx*racy; while(rac1>=p) rac1-=p; /* racines carrées de z */
        printf("\n\n %d a deux racines carrees [%d] : %d et %d",z,p,rac1,p-rac1);
    }
  getchar();return 0;
}

void f(int jj) /* successeur de jj sur sa trajectoire */
{ px[jj+1]=px[jj]*px[jj]; while (px[jj+1]>=p) px[jj+1]-=p;
}
 void f1(int jj)  /* idem mais sur l'arborescence de 1 */
{ px1[jj+1]=px1[jj]*px1[jj];   while (px1[jj+1]>=p) px1[jj+1]-=p;
 }
int inversemodulo(int a, int m)  /* calcul d'inverse */
{ int x0=1,x1=0,r0=a,r1=m,q,r2,x2;
   while (r1!=0)
     { q=r0/r1;   r2=r0-q*r1;   x2=x0-x1*q;    x0=x1; x1=x2; r0=r1; r1=r2;  }
   while (x0>=m) x0-=m; while (x0<0) x0+=m;
   return x0 ;
}
```

2. Deuxième cas : le modulo *m* est une puissance p^k de nombre premier impair

On est maintenant dans $U(p^k)$ qui compte $p^{k-1}(p-1) = 2^a p^{k-1} h$ éléments. Puisque $U(p^k)$ est un groupe cyclique, il se décompose en produit direct de deux sous-groupes cycliques *U'* et *C'*, avec *U'* d'ordre 2^a et *C'* d'ordre $p^{k-1} h$. Tout élément *z* de $U(p^k)$ se décompose de façon unique en produit d'un élément de *U'* et d'un élément de *C'*, ceux-ci étant obtenus de la même façon qu'on l'a fait modulo *p*. Plus précisément :

* ***C'* est l'ensemble des éléments *x* tels que $x^{p^{k-1}h} = 1$ [p^k]. C'est aussi l'ensemble des p^{k-1} *h* copies des *h* éléments cycliques de *C* dans *U*(*p*). Il se décompose suivant $C = C_1 \otimes C_2$, où C_1 est le groupe cyclique formé de p^{k-1} copies de 1, et où C_2 est le groupe cyclique constitué par les *h* copies des *h* éléments de *C* qui conservent leur ordre lors du passage de [*p*] à [p^k].**

Considérons le groupe *C'* formé des éléments *x* tels que $x^{p^{k-1}h} = 1$ [p^k]. Il compte exactement $pgcd(p^{k-1}(p-1), p^{k-1} h) = p^{k-1} h$ éléments. D'autre part, tous ces *x* vérifient aussi la même équation modulo *p*, soit $x^{p^{k-1}h} = 1$ ou $x^h = 1$ [*p*] . Les *x* obtenus ainsi sont des copies des éléments de *C*, ce groupe *C* ayant *h* éléments et les copies d'un élément *a* étant *a* + *qp*. Or ces copies sont au nombre de $p^{k-1} h$, autant qu'il y a d'éléments dans *C'*. Le groupe *C'* est l'ensemble des copies des *h* éléments cycliques de *C* vérifiant $x^h = 1$ [*p*]. Le sous-groupe *C'* est cyclique et il admet des générateurs d'ordre $p^{k-1} h$. Par élévations au carré successives, tout élément de *C'* finit par retomber sur lui-même. Pour connaître la partition en cycles de ces éléments, il suffit de décomposer la permutation $x \to 2x$ [$p^{k-1} h$] en cycles.

Dans *C*', considérons le sous-groupe C_1 possédant p^{k-1} éléments. Comme l'équation $x^{p^{k-1}} = 1$ [p^k] équivaut à $x = 1$ [*p*] [13], C_1 est aussi l'ensemble des p^{k-1} copies de 1. D'autre part, le groupe *C'* des

[13] Il s'agit là d'une application de la formule $a^{p^s} = b^{p^s}$ [p^{k+s}] $\Leftrightarrow a = b$[p^k] lorsque *p* est premier impair et que *b* est premier avec *p*.

points cycliques contient aussi un sous-groupe C_2 de h éléments, ces éléments vérifiant $x^h = 1\ [p^k]$. Il s'agit des copies d'éléments cycliques modulo p qui conservent leur ordre lorsque l'on passe modulo p^k. On sait que chaque élément modulo p a une et une seule de ces copies qui conserve son ordre. Cela donne tous les éléments de C_2. Finalement tout élément de C' se décompose de façon unique en produit d'un élément de C_1 et d'un élément de C_2 (puisque h est premier avec p^{k-1}). On en déduit une méthode constructive pour obtenir les longueurs des cycles de C' pour les puissances successives de p.

Exemples

1) $p = 3$. Le groupe C se réduit au cycle de 1. Quanr on prend les puissances successives de 3, C_2 se réduit toujours un cycle de longueur 1. Les longueurs des cycles de C' sont donc celles de C_1. Pour $p^2 = 9$, on trouve pour les 3 éléments de C' un cycle de longueur 1 et un cycle de longueur 2. Pour $p^3 = 27$, les 9 éléments de C' ont des cycles de longueur 1, 2, 6. Pour $p^4 = 81$, les cycles sont de longueur 1, 2, 6, 18. Pour $p^6 = 243$, les cycles sont de longueur 1, 2, 6, 18, 54.

2) $p = 7$. Le groupe C est formé du cycle de 1 et d'un cycle de longueur 2 où les éléments ont pour ordre 3.

Passons à $p^2 = 49$: le groupe C_2 est la reproduction du groupe C précédent. Quant au groupe C_1, il est formé du cycle de 1 et de deux cycles de longueur 3 (où les éléments ont pour ordre 7). A son tour le groupe C' est formé par la composition des cycles de C_1 et C_2, ce qui les redonne, ainsi que deux nouveaux cycles de longueur 6 (où les éléments ont pour ordre 21) [14]. Finalement C' est constitué d'un cycle de longueur 1, d'un cycle de longueur 2, de deux cycles de longueur 3, et de deux cycles de longueur 6.

Passons à $p^3 = 343$: le groupe C_2 a la même forme que précédemment, le groupe C_1 est formé des copies de 1, avec 1 élément d'ordre 1, $p - 1$ copies d'ordre p et $p^2 - p$ copies d'ordre p^2. Cela fait un cycle d'ordre 1, deux cycles de longueur 3, et deux cycles de longueur 21. Pour obtenir les cycles de C', on compose les cycles de C_1 et C_2 , ce qui fait apparaître comme cycles supplémentaires deux cycles de longueur 6 et deux cycles de longueur 42. On constate que lors du passage d'une puissance de p à la suivante, il y a héritage de tous les cycles déjà obtenus, avec en plus l'adonction de nouveaux cycles. Et cela se généralise.

3) $p = 11$. On a vu que dans C les cycles sont de longueur 1 et 4. Passons à $p^2 = 121$, où le groupe C' a $p\,h = 55$ éléments. C_2 reproduit C avec ses cycles de longueur 1 et 4. Dans C_1, avec ses 11 copies de 1, on constate que l'on a deux cycles de longueur 1 et 10. Composons les cycles de C_2 et C_1 : on obtient 5 cycles de longueur 1, 4, 10, 20, 20, car pour les deux derniers cycles, le *ppmc* de 10 et 4 est 20 et non 40.

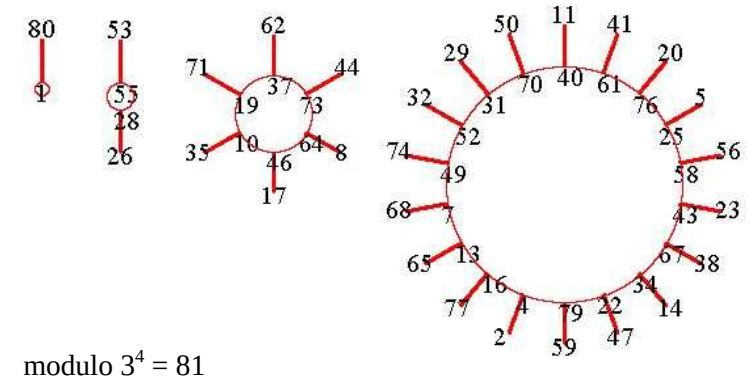
Des dessins des cycles et de leurs arborescences sont donnés sur la *figure 9*.

* **Les arborescences issues des points cycliques comportent *a* étages et sont identiques par leur forme à celles obtenues modulo *p*.**

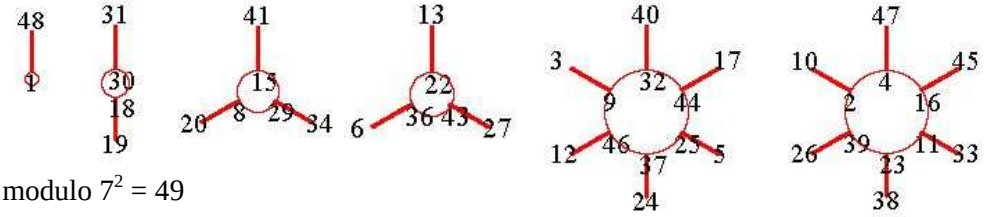
Le groupe U' est formé des éléments x tels que $x^{2^a} = 1\ [p^k]$. Ils sont au nombre de 2^a, et sont des copies des éléments de U, U et U' comportant le même nombre d'éléments. Les arborescences issues des points cycliques sont identiques modulo p^k à celles obtenues modulo p. Cela tient à ce que l'équation $x^2 = b\ [p^k]$ a exactement le même nombre de solutions que $x^2 = b\ [p]$, les deux racines carrées obtenues modulo p^k étant des copies des racines carrées modulo p. Les arborescences comportent donc a étages, et ont la forme d'arbres binaires totalement équilibrés avec un tronc.

---

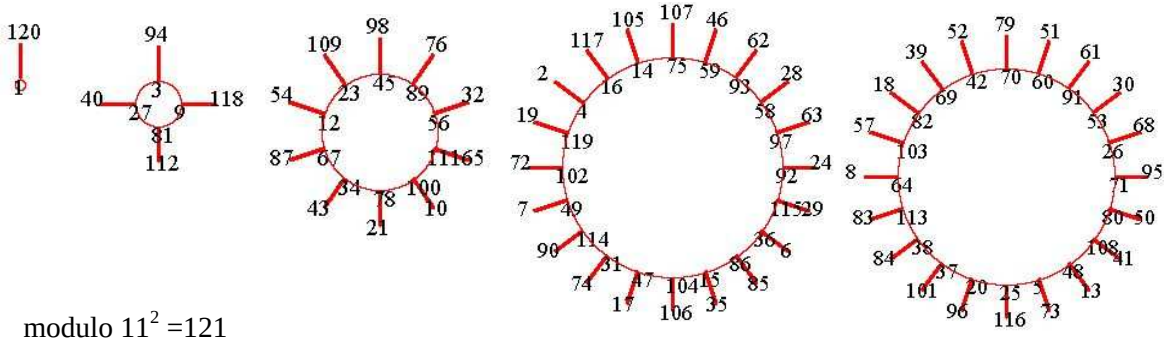
[14] Le composition de deux cycles, l'un de C_1 avec pour longueur a et l'autre de C_2 avec pour longueur b, donne naissance à des cycles dont le nombre est $pgcd(a, b)$ et la longueur $ppmc(a, b)$.



modulo $3^4 = 81$



modulo $7^2 = 49$



modulo $11^2 = 121$

Figure 9 : Arborescences pour 81, 49 et 121

3. Troisième cas : *m* est une puissance de 2

Posons $m = 2^a$. L'ensemble $U(2^a)$ est constitué des nombres impairs, au nombre de 2^{a-1} :
$U(2^a) = \{1, 3, 5,..., 2^a - 1\}$. Les cas spéciaux où $a = 1$ et $a = 2$ étant particulièrement simples, nous supposerons que $a > 2$, disposant alors des théorèmes classiques :

* **L'élément 5 a pour ordre 2^{a-2}. Tout élément de $U(2^a)$ s'écrit de façon unique sous la forme $\pm 5^j$. Il en découle notamment que $U(2^a)$ n'est pas un groupe cyclique.**

* **Dans $U(2^a)$, l'équation $x^n = b$ admet une solution unique lorsque n est impair. Lorsque n est pair, on calcule le pgcd de n et 2^{a-2}, soit 2^c, et l'équation admet 2^{c+1} solutions ou aucune selon que $b = 1\ [2^{c+2}]$ ou non.**

Ainsi, l'équation $x^{2^h} = 1\ [2^a]$ admet 2^{h+1} solutions lorsque $h \leq a - 2$, et 2^{a-1} solutions (tous les éléments de $U(2^a)$) pour $h \geq a - 2$. L'action de la récurrence $x \to x^2$ aboutit à la formation d'une arborescence unique convergeant vers le point fixe 1. Plus précisément, l'équation $x^2 = 1$ admet quatre solutions (dont 1 d'ordre un, et trois éléments d'ordre 2). Pour $a > 3$, l'équation $x^4 = 1$ admet 8 solutions (outre les précédentes, il y a quatre éléments d'ordre 4), puis $x^8 = 1$ admet 16 solutions (outre toutes les précédentes, il y a 8 éléments d'ordre 8), et ainsi de suite jusqu'à $x^{2^{a-2}} = 1$ qui donne tous les éléments de $U(2^a)$ et en particulier 2^{a-2} éléments d'ordre 2^{a-2}, qui constituent les

feuilles de l'arborescence. On vérifie aisément que ces feuilles sont constituées des éléments de la forme $\pm 5^{j}$ avec *j* impair. D'autre part, les éléments de la forme -5^{j} ne peuvent pas être des carrés. Ainsi tout élément de la forme -5^{j} (au nombre de 2^{a-2}) ou de la forme 5^{j} avec *j* impair (au nombre de 2^{a-3}) n'a pas de racine carrée. Sur les 2^{a-1} éléments de $U(2^a)$, il y a 2^{a-3} carrés, qui possèdent chacun 4 racines carrées.

Tous ces renseignements permettent de construire l'arborescence de 1. Il se dégage une mécanique d'évolution lorsque *a* augmente. Pour passer de l'arbre correspondant à 2^a à celui correspondant à 2^{a+1}, il suffit de doubler les exposants des 5 dans l'arbre de 2^a et de rajouter un étage de feuilles avec tous les exposants impairs (*figure 10*).

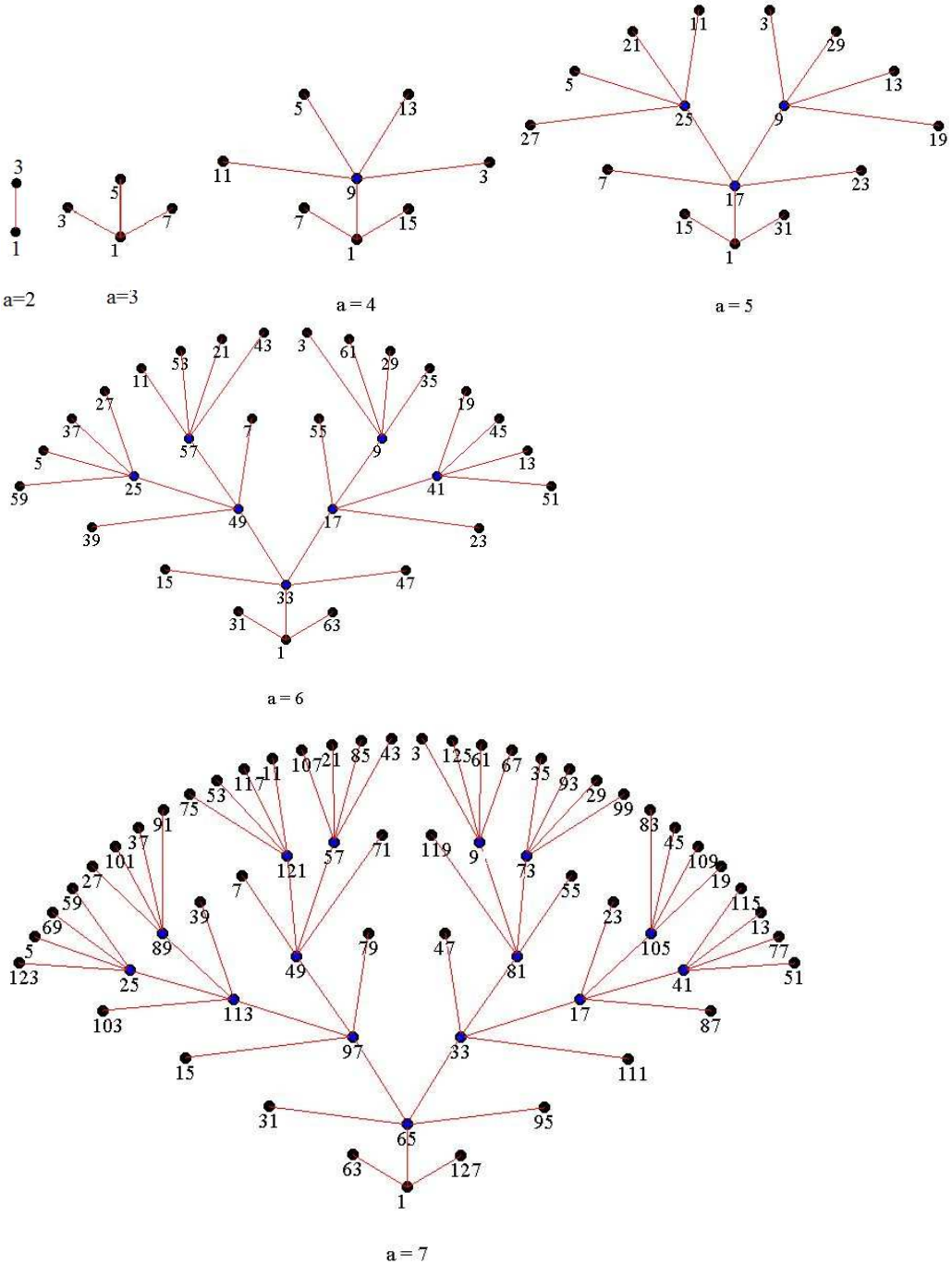


Figure 10 : Arborescence de 1, pour $m = 2^a$, avec *a* de 2 à 7.

On trouvera ci-dessous le programme complet qui a permis d'obtenir les résultats de la *figure 9*.

```
#include <SDL/SDL.h>
#include <SDL/SDL_ttf.h>
#include <stdlib.h>
#include <stdio.h>
#include <math.h>
#define xorig 400
#define yorig 500
int puiss5(int aa, int exp, int s);
void afficher (int nombre,int e, int i);
void effacer (int nombre,int e, int i);
void pause(void);                          /* fonctions disponibles dans graphical functions */
void putpixel(int xe, int ye, Uint32 c);
Uint32 getpixel(int xe, int ye);
void filldisc( int xo, int yo, int R, Uint32 c);
void line(int x0,int y0, int x1,int y1, Uint32 c);
SDL_Surface * screen;Uint32 white,black,red,blue,yellow;
SDL_Surface *texte;  SDL_Rect position;TTF_Font *police=NULL;char chiffre[2000];
SDL_Color couleurnoire={0,0,0};
SDL_Color couleurblanche={255,255,255};
int expo[10][128],signe[10][128],nombre[10][128];
int xe[10][128],ye[10][128];

int main(int argc, char ** argv)
{ int a,i,j,k,kk,R[10],d,dd,etage,nbelements[10],e;
  int q[26]={1,2,5,6,9,10,13,14,17,18,21,22,25,26,29,30,33,34,37,38,41,42,45,46,49,50};
  double alpha[10];
   SDL_Init(SDL_INIT_VIDEO);
   screen=SDL_SetVideoMode(800,600,32, SDL_HWSURFACE|SDL_DOUBLEBUF);
   white=SDL_MapRGB(screen->format,255,255,255);
   black=SDL_MapRGB(screen->format,0,0,0);
   blue=SDL_MapRGB(screen->format,0,0,255);
   red=SDL_MapRGB(screen->format,255,0,0);
   SDL_FillRect(screen,0,white);
   TTF_Init(); police=TTF_OpenFont("times.ttf",20);
/* etage 0 */
filldisc(xorig,yorig,5, black);
xe[0][0]=xorig; ye[0][0]=yorig;  nbelements[0]=1;
expo[0][0]=2; signe[0][0]=1;
nombre[0][0]=puiss5(3,expo[0][0],signe[0][0]); afficher(nombre[0][0],0,0);
/* etage 1 */
R[3]=60; alpha[3]=2.*M_PI/3.*(1./2.);
etage=1;  nbelements[1]=3;
for(j=0;j<nbelements[1];j++)
   { xe[etage][j]=xorig+R[3]*cos(M_PI/6.+j*alpha[3]);
     ye[etage][j]=yorig-R[3]*sin(M_PI/6.+j*alpha[3]);
     filldisc(xe[etage][j],ye[etage][j],5,black);
     line(xorig,yorig,xorig+R[3]*cos(M_PI/6.+j*alpha[3]),
                      yorig-R[3]*sin(M_PI/6.+j*alpha[3]),red);
   }
 expo[1][0]=2; expo[1][1]=1;expo[1][2]=1;
 signe[1][0]=-1; signe[1][1]=1;signe[1][2]=-1;
 for(j=0;j<3;j++)
   { nombre[1][j]=puiss5(3,expo[1][j],signe[1][j]);   afficher(nombre[1][j],1,j); }
 SDL_Flip(screen);pause();
/* etages 2, 3, 4, 5, etc. quand a vaut 4, 5, 6, 7... */
nbelements[2]= 4;dd=1;
for(a=4;a<=7;a++)   /* à partir de l'étage 2 (et a = 4) */
  { sprintf( chiffre,"a = %d",a);
    texte=TTF_RenderText_Solid(police,chiffre,couleurnoire);
    position.x=380; position.y=550;
```

```
    SDL_BlitSurface(texte,NULL,screen,&position);

     etage=a-2;
     for(e=0;e<etage;e++) for(j=0; j<nbelements[e];j++)
       { effacer(nombre[e][j],e,j);
         expo[e][j]*=2;
         nombre[e][j]=puiss5(a,expo[e][j],signe[e][j]);   afficher(nombre[e][j],e,j);
       }
    R[a]=R[a-1]+92; alpha[a]=2.*M_PI/3.*(1./(float)(nbelements[etage]-1));
    /* feuilles  du nouvel étage */
    for(j=0;j<nbelements[etage];j++)
       {xe[etage][j]=xorig+R[a]*cos(M_PI/6.+j*alpha[a]);
         ye[etage][j]=yorig-R[a]*sin(M_PI/6.+j*alpha[a]);
         filldisc(xe[etage][j],ye[etage][j],5,black);
       }
    /* dessin des liens entre feuilles et leurs prédécesseurs */
    kk=0;
    for(k=0;k<dd;k++) for(j=0;j<4;j++)
      { line(xe[etage-1][q[k]], ye[etage-1][q[k]], xe[etage][kk],ye[etage][kk] ,red);
        kk++;
        filldisc(xe[etage-1][q[k]], ye[etage-1][q[k]],3,blue);
      }
     kk=0;
    for(k=0;k<2*dd;k+=2)
      {  expo[etage][q[k]]=expo[etage-1][q[kk]]/2+2*dd;  signe[etage][q[k]]=1;
         expo[etage][q[k+1]]=expo[etage-1][q[kk]]/2;   signe[etage][q[k+1]]=1;
         expo[etage][q[k]-1]=expo[etage][q[k]];   signe[etage][q[k]-1]=-1;
         expo[etage][q[k+1]+1]=expo[etage][q[k+1]];   signe[etage][q[k+1]+1]=-1;

         nombre[etage][q[k]]=puiss5(a,expo[etage][q[k]],signe[etage][q[k]]);
         afficher(nombre[etage][q[k]],etage,q[k]);
         nombre[etage][q[k+1]]=puiss5(a,expo[etage][q[k+1]],signe[etage][q[k+1]]);
         afficher(nombre[etage][q[k+1]],etage,q[k+1]);
         nombre[etage][q[k]-1]=puiss5(a,expo[etage][q[k]-1],signe[etage][q[k]-1]);
         afficher(nombre[etage][q[k]-1],etage,q[k]-1);
         nombre[etage][q[k+1]+1]=puiss5(a,expo[etage][q[k+1]+1],signe[etage][q[k+1]+1]);
         afficher(nombre[etage][q[k+1]+1],etage,q[k+1]+1);
         kk++;
      }
    nbelements[etage+1]=2*nbelements[etage]; dd=2*dd;
    SDL_Flip(screen);pause();
    sprintf( chiffre,"a = %d",a);  /* on efface a */
    texte=TTF_RenderText_Solid(police,chiffre,couleurblanche);
    position.x=380; position.y=550;
    SDL_BlitSurface(texte,NULL,screen,&position);
  }
SDL_Flip(screen);pause();return 0;
}

int puiss5(int aa, int exp, int s)  /* calcule s*2 exp ramené modulo 2a */
{ int modulo,i,q;
   modulo=pow(2,aa);
   q=1;
   for(i=1;i<=exp;i++)
     { q=5*q;  while(q>=modulo) q-=modulo;  }
    if (s==-1)
     { q=-q; while (q<0) q+=modulo;  }
   return q;
}
void afficher (int nombre,int e, int i)
```

```
{
 sprintf( chiffre," %d",nombre);
  texte=TTF_RenderText_Solid(police,chiffre,couleurnoire);
  position.x=xe[e][i]-15; position.y=ye[e][i];
  SDL_BlitSurface(texte,NULL,screen,&position);
}
void effacer (int nombre,int e, int i)
{
 sprintf( chiffre," %d",nombre);
  texte=TTF_RenderText_Solid(police,chiffre,couleurblanche);
  position.x=xe[e][i]-15; position.y=ye[e][i];
  SDL_BlitSurface(texte,NULL,screen,&position);
}
```

4. Quatrième cas : *m* est un nombre composite

Connaissant le graphe des trajectoires pour deux nombres premiers p_1 et p_2 avec $m = p_1 p_2$, celui de *m* s'en déduit.. Et cela se généralise à un nombre quelconque de puissances de nombres premiers. La composition des arborescences se fait en appliquant le théorème chinois, grâce à l'isomorphisme qu'il établit entre $U(p_1)$ $U(p_2)$ avec $U(m)$. Si a est un élément de $U(p_1)$ et b un élément de $U(p_2)$, leur correspondant dans $U(m)$ est le couple (a, b).[15]

De là découlent les *règles du jeu* suivantes :

* Les points cycliques de $U(m)$ s'obtiennent en prenant tous les couples de points cycliques de $U(p_1)$ et $U(p_2)$. Leur nombre est le produit de leurs nombres dans $U(p_1)$ et $U(p_2)$.

* Si $U(p_1)$ contient k_1 cycles et que $U(p_2)$ en possède k_2, on compose pour $U(m)$ chaque cycle de l'un avec chaque cycle de l'autre, mais le résultat de cette composition peut donner plusieurs cycles. Si un cycle de $U(p_1)$ possède c_1 éléments, et que le cycle de $U(p_2)$ en contient c_2, le nombre de cycles de leur composition est $c_1 c_2 / (ppmc\ (c_1 , c_2))$ ou encore $pgcd(c_1, c_2)$, et la longueur de chacun de ces cycles est $ppmc(c_1, c_2)$. Par exemple, la composition du cycle $(a_1\ a_2)$ avec le cycle $(b_1\ b_2\ b_3)$ donne le cycle $((a_1, b_1)\ (a_2, b_2)\ (a_1,b_3)\ (a_2,b_1)\ (a_1,b_2)\ (a_2,b_3))$. Mais la composition du cycle $(a_1\ a_2)$ avec le cycle $(b_1\ b_2)$ donne les deux cycles $((a_1, b_1)(a_2, b_2))$ et $((a_1, b_2)(a_2, b_1))$.

* Tout noeud de $U(m)$ (un noeud est un élément qui a des antécédents, c'est-à-dire des racines carrées) est formé d'un couple de deux noeuds, l'un dans $U(p_1)$ et l'autre dans $U(p_2)$. Par contre le couple de deux feuilles (éléments sans racines carrées) ou le couple d'un noeud et d'une feuille donne une feuille. Le nombre des racines carrées d'un noeud est le produit de leurs nombres dans $U(p_1)$ et $U(p_2)$.

* Le nombre d'étages des arborescences est le maximum des deux nombres d'étages pour p_1 et p_2, et l'ordre des éléments est le ppmc des ordres de leurs composants.

Ces règles donnent l'algorithme de composition des graphes des trajectoires. Les *portraits* de nombres obtenus grâce à ces cycles et ces arborescences dévoilent la structure interne des groupes $U(m)$, non seulement à propos des racines carrées, mais aussi sur l'ordre des éléments et la présence éventuelle de générateurs. **[16]**

---

[15] Par exemple, résoudre $x^2 = 1$ [40] revient à chercher les couples (a, b) vérifiant $a^2 = 1$ [8] et $b^2 = 1$ [5], ce qui donne a = 1, 3, 5 ou 7 et b = 1 ou 4 , d'où 8 couples-soltions (a, b), ce qui permet de trouver les x correspondants, grâce à la relation $x = -15a + 16b\ [40]$, – 15 étant le nombre correspondant au couple (1, 0) et 16 celui corespondant au couple (0,1).

[16] La construction des arborescences permet d'ailleurs de retrouver que les groupes $U(m)$ sont cycliques (ils ont des générateurs) si et seulement si m est de la forme 2, 4, p^a ou $2p^a$ (avec p premier impair).

Exemple : 91 = 7×13

Comme *U*(7) possède deux cycles, de longueurs 1 et 2 (que nous notons 1 et 2) et *U*(13) aussi, avec ses cycles notés 1’ et 2’, *U*(91) a 3×3= 9 points cycliques. La composition des cycles: 1-1’, 1-2’, 2-1’, et 2-2’ donne respectivement un cycle de longueur 1, un cycle de longueur 2, un cycle de longueur 2, tandis que la dernière composition 2-2’ donne deux cycles de longueur 2 chacun. D’autre part, chaque noeud possède 2×2 = 4 racines carrées. Les arbres sont à deux étages, et l’ordre maximal des éléments est 12, ce qui prouve que *U*(91) n’est pas cyclique, n’ayant pas de générateurs (*figure 11*).

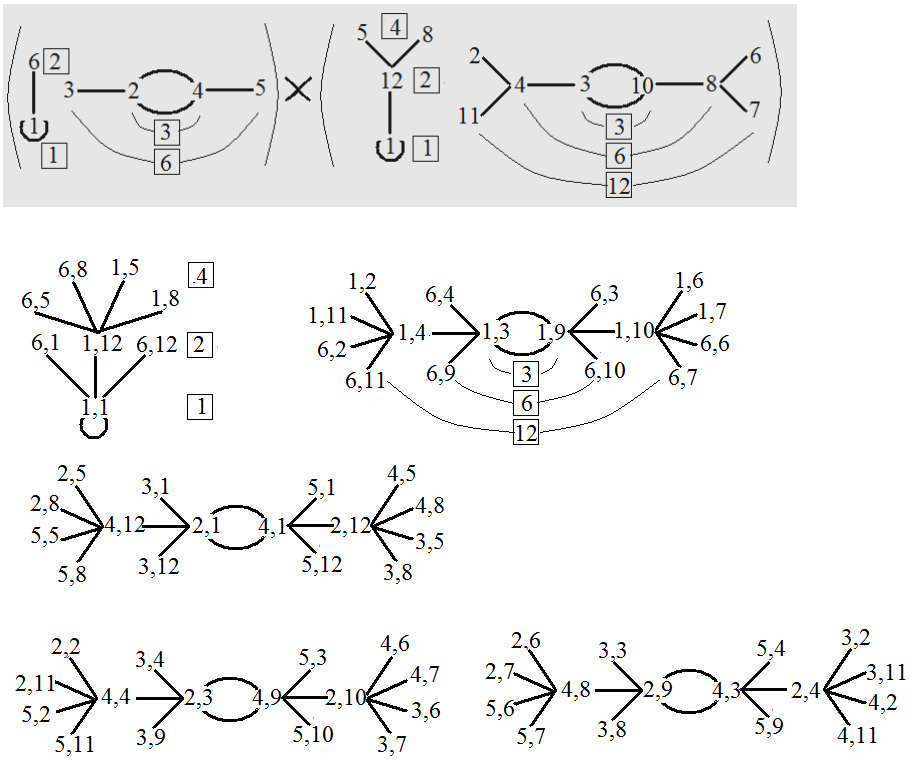
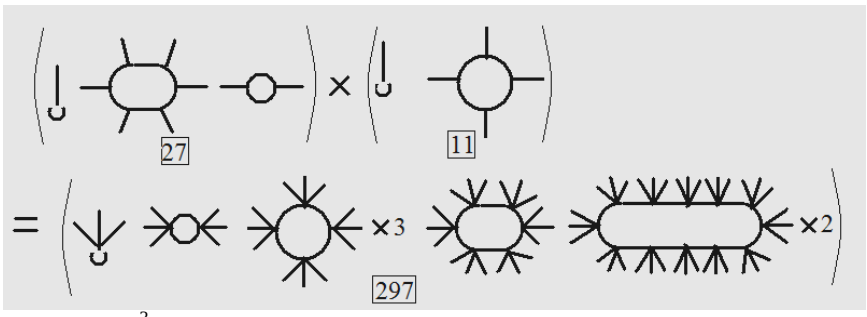
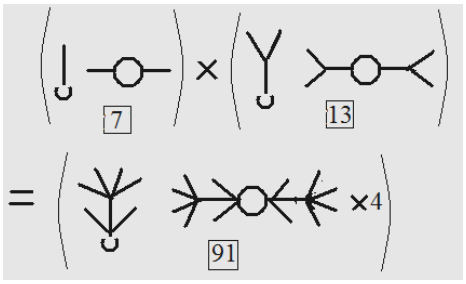


Figure 11 : *En haut* les graphes de *U*(7) et *U*(13), avec les ordres des éléments encadrés. *Au-dessous* les 5 arborescences de *U*(91), avec 91 = 7×13, obtenues par composition de celles de 7 et 13.

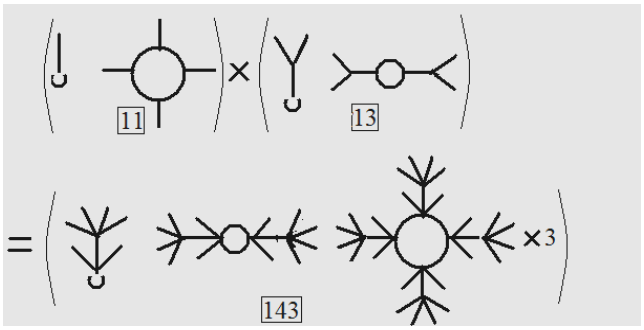
Sur la *figure 12*, nous donnons l’allure des graphes obtenus lorsque l’on compose plusieurs types de trajectoires.



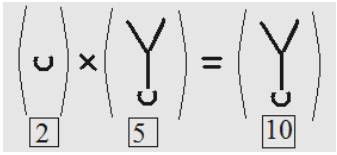
m = 297 = 3^3 ×11



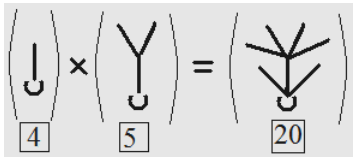
$m = 7 \times 13$



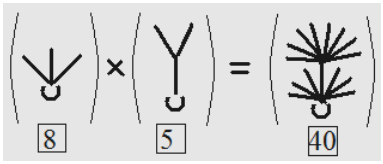
$m = 11 \times 13$



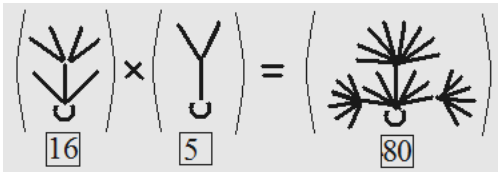
$m = 10 = 2 \times 5$



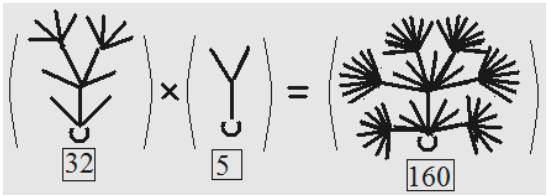
$m = 20 = 4 \times 5$



$m = 40 = 8 \times 5$



$m = 80 = 16 \times 5$



$m = 160 = 32 \times 5$

Figure 12 : Divers exemples de composition d'arborescences (la composition étant notée comme une multiplication).