Le théorème de Zeckendorf

Guilhem Repetto

I La décomposition de Fibonacci

On définit la suite de Fibonacci par $F_0 = 0, F_1 = 1, \forall n \in \mathbb{N}, F_{n+2} = F_{n+1} + F_n$.

Théorème : pour tout entier naturel N, il existe une unique suite strictement croissante d'entiers $c_0, \dots, c_k$, avec $c_0 \geq 2$ et $\forall i \in [\![0, k-1]\!], c_{i+1} > c_i + 1$ telle que

$$N = \sum_{i=0}^{k} F_{c_i}$$

Preuve :

Existence : Par récurrence forte :

- 0 est représenté par la somme vide ;
- Soit N, et F_k le plus grand nombre de Fibonacci inférieur ou égal à N. Alors, par hypothèse de récurrence, $N - F_k$ possède une représentation, notée $\sum_{l=0}^{r} F_l$. En particulier, pour tout l, on a
$$F_k + F_l \leq F_k + (N - F_k) = N < F_{k+1} = F_k + F_{k-1},$$
ce qui montre que $F_l < F_{k-1}$, et donc $l + 1 < k$. Ainsi, $N = F_k + \sum_{l=0}^{r} F_l$ ce qui est de plus une représentation correcte.

Unicité : Prouvons par récurrence que la somme de nombres de Fibonacci distincts et non deux-à-deux consécutifs, dont le plus grand est F_k, est strictement inférieure à F_{k+1}.

- Une somme de zéro termes vaut 0, et $0 < F_1$.
- Supposons la propriété vérifiée pour une somme de $n-1$ termes. Soit $F_{c_0} + \cdots + F_{c_{n-1}} + F_{c_n}$ une somme. Par hypothèse de récurrence $F_{c_0} + \cdots + F_{c_{n-1}} < F_{c_{n-1}+1}$. Comme les termes sont non-consécutifs, on a $c_{n-1} + 1 < c_n$, c'est-à-dire $c_{n-1} + 1 \leq c_n - 1$. Par croissance de la suite (F_n), on obtient alors $F_{c_{n-1}+1} \leq F_{c_n-1}$ et donc
$$F_{c_0} + \cdots + F_{c_{n-1}} + F_{c_n} < F_{c_{n-1}+1} + F_{c_n} \leq F_{c_n-1} + F_{c_n} = F_{c_n+1}$$

On raisonne ensuite par récurrence sur l'entier naturel N. D'après la propriété précédente, si $N \neq 0$, dans une décomposition de N, le plus grand terme de la suite de Fibonacci qui apparaît est l'unique F_k tel que $F_k \leq N < F_{k+1}$. Par hypothèse de récurrence, la décomposition de $N - F_k$, qui constitue le reste de la décomposition de N, est également unique.

□

II Application : un tour de magie

Matériel : onze cartes sur lesquelles sont inscrits les nombres suivants :

1; 4; 6; 9; 12; 14; 17; 19; 22; 25; 27; 30; 33; 35; 38; 40; 43; 46; 48; 51; 53; 56; 59; 61; 64; 67; 69; 72; 74; 77; 80; 82; 85; 88; 90; 93; 95; 98; 101; 103; 106; 108; 111; 114; 116; 119; 122; 124; 127; 129; 132; 135; 137; 140; 142; 145; 148; 150; 153; 156; 158; 161; 163; 166; 169; 171; 174; 177; 179; 182; 184; 187; 190; 192; 195; 197; 200

2; 7; 10; 15; 20; 23; 28; 31; 36; 41; 44; 49; 54; 57; 62; 65; 70; 75; 78; 83; 86; 91; 96; 99; 104; 109; 112; 117; 120; 125; 130; 133; 138; 143; 146; 151; 154; 159; 164; 167; 172; 175; 180; 185; 188; 193; 198

3; 4; 11; 12; 16; 17; 24; 25; 32; 33; 37; 38; 45; 46; 50; 51; 58; 59; 66; 67; 71; 72; 79; 80; 87; 88; 92; 93; 100; 101; 105; 106; 113; 114; 121; 122; 126; 127; 134; 135; 139; 140; 147; 148; 155; 156; 160; 161; 168; 169; 176; 177; 181; 182; 189; 190; 194; 195

5; 6; 7; 18; 19; 20; 26; 27; 28; 39; 40; 41; 52; 53; 54; 60; 61; 62; 73; 74; 75; 81; 82; 83; 94; 95; 96; 107; 108; 109; 115; 116; 117; 128; 129; 130; 141; 142; 143; 149; 150; 151; 162; 163; 164; 170; 171; 172; 183; 184; 185; 196; 197; 198

8; 9; 10; 11; 12; 29; 30; 31; 32; 33; 42; 43; 44; 45; 46; 63; 64; 65; 66; 67; 84; 85; 86; 87; 88; 97; 98; 99; 100; 101; 118; 119; 120; 121; 122; 131; 132; 133; 134; 135; 152; 153; 154; 155; 156; 173; 174; 175; 176; 177; 186; 187; 188; 189; 190

13; 14; 15; 16; 17; 18; 19; 20; 47; 48; 49; 50; 51; 52; 53; 54; 68; 69; 70; 71; 72; 73; 74; 75; 102; 103; 104; 105; 106; 107; 108; 109; 136; 137; 138; 139; 140; 141; 142; 143; 157; 158; 159; 160; 161; 162; 163; 164; 191; 192; 193; 194; 195; 196; 197; 198

21; 22; 23; 24; 25; 26; 27; 28; 29; 30; 31; 32; 33; 76; 77; 78; 79; 80; 81; 82; 83; 84; 85; 86; 87; 88; 110; 111; 112; 113; 114; 115; 116; 117; 118; 119; 120; 121; 122; 165; 166; 167; 168; 169; 170; 171; 172; 173; 174; 175; 176; 177

34; 35; 36; 37; 38; 39; 40; 41; 42; 43; 44; 45; 46; 47; 48; 49; 50; 51; 52; 53; 54; 123; 124; 125; 126; 127; 128; 129; 130; 131; 132; 133; 134; 135; 136; 137; 138; 139; 140; 141; 142; 143; 178; 179; 180; 181; 182; 183; 184; 185; 186; 187; 188; 189; 190; 191; 192; 193; 194; 195; 196; 197; 198

55; 56; 57; 58; 59; 60; 61; 62; 63; 64; 65; 66; 67; 68; 69; 70; 71; 72; 73; 74; 75; 76; 77; 78; 79; 80; 81; 82; 83; 84; 85; 86; 87; 88; 199; 200

89; 90; 91; 92; 93; 94; 95; 96; 97; 98; 99; 100; 101; 102; 103; 104; 105; 106; 107; 108; 109; 110; 111; 112; 113; 114; 115; 116; 117; 118; 119; 120; 121; 122; 123; 124; 125; 126; 127; 128; 129; 130; 131; 132; 133; 134; 135; 136; 137; 138; 139; 140; 141; 142; 143

144; 145; 146; 147; 148; 149; 150; 151; 152; 153; 154; 155; 156; 157; 158; 159; 160; 161; 162; 163; 164; 165; 166; 167; 168; 169; 170; 171; 172; 173; 174; 175; 176; 177; 178; 179; 180; 181; 182; 183; 184; 185; 186; 187; 188; 189; 190; 191; 192; 193; 194; 195; 196; 197; 198; 199; 200

Le spectateur choisit mentalement un nombre entier entre 0 et 200, et indique à la magicienne les cartes sur lesquelles le nombre est présent. Celle-ci peut alors très vite trouver le nombre choisi.

Explication : le plus petit nombre de chaque carte est un terme de la suite de Fibonacci, et une carte dont le plus petit nombre est n présente exactement les nombres entre 0 et 200 dont la décomposition de Zeckendorf contient n. Il suffit donc pour la magicienne d'additionner mentalement tous les plus petits nombres de chaque carte pour obtenir le nombre pensé. Si le spectateur ne rend aucune carte, c'est que le nombre pensé est 0 !

La magicienne aurait pu utiliser des cartes exploitant une autre décomposition, par exemple celle selon les puissances de 2 :

1 ;3 ;5 ;7 ;9 ;11 ;13...	4 ;5 ;6 ;7 ;12...
2 ;3 ;6 ;7 ;10 ;11...	8 ;9 ;10 ;11 ;12 ;13...

III Code source

On peut générer des cartes avec plus de nombres avec le code `OCaml` suivant :

```ocaml
let nfibo=50;;

(* Suite de Fibonacci *)
let f =
  let k = Array.make nfibo 0
  in
  k.(1)<-1;
  for i=2 to (nfibo-1) do
    k.(i)<-k.(i-1) + k.(i-2)
  done;
  k;;

(* Terme maximal de f inferieur a n *)
let maxmin n t =
  let r=ref 0 in
  for i=0 to (Array.length t -1) do
    if t.(i)<=n then r:= t.(i)
  done;
  !r;;

(* Decomposition de Zeckendorf de n *)
let rec zeck n =
  if n=0 then []
  else if n<=f.(1) then [n]
  else let k = maxmin n f in
  k::(zeck (n-k));;

(* Renvoie l'indice de la case de cartes qui contient i *)
let indice i cartes =
  let rec aux a b =
  if b<a then failwith "erreur"
  else if a=b then
    if fst (cartes.(a))<>i then failwith "pas de carte"
    else a
  else let m=(b+a)/2 in
    let ind = fst (cartes.(m)) in
    if ind > i then aux a m
    else if ind=i then m
    else aux m b
  in
    aux 0 (Array.length cartes - 1);;

(* Ajoute n aux cartes listees dans l *)
let rec ajoute_nombre_aux_cartes n l cartes =
  match l with
    []->()
```

```
    |t::q->
        let i=indice t cartes in
        cartes.(i)<- (fst (cartes.(i)), n :: snd (cartes.(i)));
        ajoute_nombre_aux_cartes n q cartes;;

(* Cree les cartes : tableau de la forme (premier_nombre, [liste des nombres]) *)
let cartes = let t = Array.make (nfibo-2) (0,[]) in
  for i=2 to (nfibo-1) do
    t.(i-2)<- (f.(i),[])
  done;
  t;;

(* Genere les cartes pour 200 nombres *)
for i=0 to 200 do
  ajoute_nombre_aux_cartes i (zeck i) cartes
done;;
```

On obtient alors la i^{ieme} carte avec `List.rev (snd (cartes.(i))`.

Sources : *Les clefs pour l'info*, I. Belghiti, R. Mansuy, J. Vie, pages 81-85