TD 1

Exercice 1

- $Rang(x,y)=(x+y)(x+y+1)/2 + x$

(0,0)	(0,1)	(1, 0)	(0,2)	(1, 1)	(2,0)	(0, 3)	(1,2)	(2, 1)	…...
0	1	2	3	4	5	6	7	8	

- Donner la version récursive de Rang

Exercice 1

- $Rang(x,y)=(x+y)(x+y+1)/2 + x$
- Donner la version récursive de Rang

$Rang(0,0) = 0$

$Rang(x,y) = Rang(x-1,y+1) + 1$ si $x \neq 0$

Exercice 1

- $Rang(x,y)=(x+y)(x+y+1)/2 + x$
- Donner la version récursive de Rang

$Rang(0,0) = 0$

si $x \neq 0$: $Rang(x,y) = Rang(x-1,y+1) + 1$

$Rang(0,y) = Rang(y-1,0) + 1$

Exercice 1

- $Rang(x,y)=(x+y)(x+y+1)/2 + x$
- Donner la fonction inverse : c'est-à-dire trouver x et y tels que $z=Rang(x,y)$

  Trouver t qui représente la somme $(x+y)$.

Exercice 1

- $Rang(x,y)=(x+y)(x+y+1)/2 + x= z$
- Donner la fonction inverse

  Trouver t qui représente la somme (x+y).

  C'est le plus grand nombre tel que $z \geq t(t+1)/2$

- t=0 ; while (t(t+1)/2≤z) t++ ; t-- ;
- x = z-t(t+1)/2 ; y=t-x ;

Exercice 1

- Rang(x,y)=(x+y)(x+y+1)/2 + x
- Rang (4,5)=9*10/2 + 4 = 49
- Rang(x,y) =8

  t=x+y est égal à 3

  x = z-t(t+1)/2 = 8-6=2

  y = t-x = 1

Exercice 2

- $c(x,y)=(x+y)(x+y+1)/2 + y$
- $h(x,y,z) = c(c(x,y),z)$

Calcul de (x,y) tel que $c(x,y)=67$

Exercice 2

- $c(x,y)=(x+y)(x+y+1)/2 + y$
- $h(x,y,z) = c(c(x,y),z)$

Calcul de (x,y) tel que $c(x,y)=67$

$t=11$; $y=1$; $x=10$ donc $(x,y)=(10,1)$

Exercice 2

- $c(x,y)=(x+y)(x+y+1)/2 + y$
- $h(x,y,z) = c(c(x,y),z)$

  $c(x,y)=67 \Rightarrow (x,y)=(10,1)$
- $h(x,y,z)=67=c(c(x,y), z)$

  donc $z=1$ et $c(x,y)=10$ ($t=4$, $y=0$, $x=4$)

  donc $(x,y,z) = (4,0,1)$

Exercice 2

- $c(x,y)=(x+y)(x+y+1)/2 + y$

  successeur(x,y) ?

Exercice 2

- $c(x,y)=(x+y)(x+y+1)/2 + y$

  $successeur(x,y) = (x-1, y+1)$ si $x \neq 0$

  $successeur(0,y)$?

Exercice 2

- $c(x,y)=(x+y)(x+y+1)/2 + y$

  $successeur(x,y) = (x-1, y+1)$ si $x \neq 0$

  $successeur(0,y) = (y+1, 0)$

Exercice 3

- Coder les listes d’entiers
- Coder d’abord les listes qui minimisent la somme des entiers des listes ne fonctionnent pas : il y a une infinité de liste dont la somme vaut 1 : (1) (01) (10) (001) (010) (100) ….

  Donc on ne code jamais la liste (2) par exemple

Exercice 3

- Coder les listes d'entiers
- Coder les listes les plus courtes ne fonctionnent pas non plus car il y a une infinité de listes de longueur 1 :

  (0) (1) (2) (3) ....

  Donc on ne code jamais la liste (0 0)

Exercice 4

- Coder les nombres rationnels
- Caractérisation :

  Les nombres rationnels s'écrivent de façon unique sous la forme a/b avec pgcd(a,b)=1.

Exercice 4

- Coder les nombres rationnels
- Caractérisation :

  Les nombres rationnels s'écrivent de façon unique sous la forme a/b avec pgcd(a,b)=1.

  Donc on code dans l'ordre de la somme a+b et à égalité l'ordre lexico.

  0/1 code 0, 1/1 : 1, 1/2 : 2, 2/1 : 3 : 1/3 : 4 ...

Exercice 5

- $U_0 = \{()\}$
- $U_1 = \{(0)\}$
- $U_2 = \{(0,0), (1)\}$

Exercice 5

- $U_0 = \{()\}$
- $U_1 = \{(0)\}$
- $U_2 = \{(0,0), (1)\}$
- $U_3 = \{(0,0,0), (0,1), (1,0), (2)\}$
- $U_4 = \{(0,0,0,0),(0,0,1),(0,1,0),(1,0,0), (0,2), (1,1), (2,0), (3)\}$

Exercice 5

(0)

(0,0) (1)

(0,0,0) (0,1) (1,0) (2)

(0,0,0,0) (0,0,1) (0,1,0)(0,2) (1,0,0)(1,1) (2,0)(3)

- (δ, x) fils gauche $(\delta, x, 0)$ fils droit $(\delta, x+1)$

Exercice 5

(0) 1

(0,0) 2 (1) 3

(0,0,0) 4 (0,1) 5 (1,0) 6 (2) 7

(0,0,0,0) (0,0,1) (0,1,0) (0,2) (1,0,0)(1,1) (2,0)(3)

10 11

- (δ ,x) fils gauche (δ,x,0) fils droit (δ ,x+1)

Exercice 5

(0) 1

(0,0) 2 (1) 3

(0,0,0) 4 (0,1) 5 (1,0) 6 (2) 7

(0,0,0,0) (0,0,1) (0,1,0)(1,0,0) (0,2)(1,1) (2,0)(3)

10 11

- (δ ,x) fils gauche (δ,x,0) fils droit (δ ,x+1)

c 2c 2c+1

Exercice 5

- Montrez que $|U_k| = 2^{k-1}$, $k \geq 1$.
- Pour chaque liste de $U_k = (\delta, x)$ (où δ est une liste quelconque) on obtient 2 listes de U_{k+1} : $(\delta, x, 0)$ $(\delta, x+1)$.
- Réciproquement une liste de U_{k+1} ne peut être obtenue par une et une seule liste de U_k de cette façon.

Exercice 5

Donc $|U_{k+1}|=2|U_k|$, $k\geq1$. Et comme $|U_1|=1$, la conclusion s'impose.

- La première liste est (0,0,0,….0) avec k zéros
- La dernière liste est (k-1)

Exercice 5

- Pour chaque liste de $U_k = (\delta, x)$ (où δ est une liste quelconque) on obtient 2 listes de U_{k+1} : $(\delta, x, 0)$ $(\delta, x+1)$.
- Remarque les 2 listes sont consécutives dans le codage : on a comme n arbre binaire parfait donc
- $c(\delta,0)=2c(\delta)$ et $c(\delta,x)=2c(\delta,x-1)+1$

Exercice 5

- $c((0))=1$, $c((0,0))=2$, $c((1))=3$,
- $c((0,0,0))=2*c((0,0))=2*2=4$
- Etc
- Donc version récursive

$c(\delta,0)=2c(\delta)$ et $c(\delta,x)=2c(\delta,x-1)+1$

Cas de base $c(0)=1$ et $c()=0$

Exercice 5

- Version itérative : code $(x_1, x_2, x_3, \ldots x_n)$
- On part de (0) code 1 à (x_1) (à chaque incrémentation on multiplie le code par 2 et on ajoute 1)
- Puis on va à $(x_1, 0)$ (multiplie le code par 2)
- Puis on va à (x_1, x_2), puis $(x_1, x_2, 0)$ etc.

Exercice 5

- Décodage :

  Si k est pair on ajoute un 0 en tête de liste, sinon on ajoute 1 à la tête de liste (on divise k par 2 à chaque itération).

Exercice 6

- $A_0=\{1\}, A_1=\{2\}, A_2=\{4\}, A_3=\{8\}$
- $A_4=\{16,$

Exercice 6

- $A_0=\{1\}$, $A_1=\{2\}$, $A_2=\{4\}$, $A_3=\{8\}$
- $A_4=\{16, 5, 10, 20, 40, 80,\ldots\ldots\}$ 3 ? 13 ?

Exercice 6

- $A_0=\{1\}$, $A_1=\{2\}$, $A_2=\{4\}$, $A_3=\{8\}$
- $A_4=\{16, 5, 10, 20, 40, 80, \ldots\ldots\}$ 3 ? 13 ?
- $A_5=\{32\}$
- $A_6=\{64, 21, 42, 84, \ldots\ldots\}$
- Comment afficher les éléments de A_i ?

Exercice 6

- Algorithme pour afficher A_i

  n=0 ;

  Tant que (true) { si (f(n)=i) afficher(n) ; n++}

- Ne fonctionne pas :

  f(n) est-elle définie pour tout n ?

  A_i peut-être fini

Exercice 6

- $2^i \in A_i$ donc $(2^i -1)/3$ aussi (à quelle condition ?)
- Si $x \in A_i$ donc $2x$ et $(x -1)/3$ aussi (à quelle condition ?)
- En déduire un algorithme.

Exercice 6

- Algorithme affiche(A_i)

afficher(2^i) ;

si ((2^i -1) mod 3=0) alors L={(2^i -1)/3}

sinon L=$\emptyset$

Tant que (L non vide) {

n=tete(L) ; afficher(n) ;

L=ajoutfin(2n,queue(L)) ;

si ((n-1) mod 3 = 0) L=ajoutfin((n-1)/3,L) ; }

Exercice 6

- Affichage de $A_4 \cup A_6$
- Pourquoi affiche(A_4) ; affiche(A_6) ne fonctionne pas ?
- Ecrire une fonction qui donne le $n^{\text{ième}}$ élément de A_i à l'aide de affiche(A_i) (en ajoutant un compteur) : f(i, n)
- n=1 ; tantque (true) {

  afficher f(4,n) ; afficher f(6,n) ; n++}