

Autres algorithmes simples (partie 1)

Sébastien Jean

IUT de Valence
Département Informatique

v1.0, 13 octobre 2025

Longueur de la suite de Syracuse

Enoncé du problème

Une suite de Syracuse est une suite d'entiers naturels $(u_0 \dots u_i)$ définie de la manière suivante : on part d'un nombre entier strictement positif ; s'il est pair, on le divise par 2 ; s'il est impair, on le multiplie par 3 et on lui ajoute 1. Il existe un indice i tel que $u_i = 1$. On cherche l'indice i .

Spécification du problème

- Donnée d'entrée : n , entier (terme de départ)
- Donnée de sortie : $indice$, entier (indice du 1er terme valant 1)
- Pré-condition : $n \geq 1$
- Post-condition : La sortie $indice$ est égale au nombre d'itérations nécessaires pour atteindre 1 en appliquant la transformation de Syracuse à partir de n .

Signature de la fonction

- **longueur_syracuse** (n : entier) : entier

Longueur de la suite de Syracuse

Corps de la fonction



- On doit calculer des termes successifs → **itération**
- Identifier la condition et en déduire la **structure de contrôle adaptée**
 - Dépassement d'un nombre de tours ? → POUR
 - Autre mais évaluée avant d'avoir fait un tour ? → TANT QUE
 - Autre mais évaluée après d'avoir fait un tour ? → REPETER

Longueur de la suite de Syracuse

- Utilisation de **TANT QUE**
 - Itération pour calculer le terme suivant, **nombre de tours non prévisible**
 - Condition d'arrêt (*valeur du terme à 1*) évaluée **avant** avoir d'effectué un tour (*n* est le premier terme)

```
FONCTION longueur_syracuse (n : entier) : entier
```

```
VARIABLE indice : entier
```

```
VARIABLE copie : entier
```

```
indice ← 0
```

```
copie ← n
```

... (suite après)

```
FIN FONCTION
```

Longueur de la suite de Syracuse

```
FONCTION longueur_syracuse (n : entier) : entier
```

```
...
```

```
TANT QUE copie ≠ 1
  SI copie mod 2 = 0 ALORS
    copie ← copie div 2
  SINON
    copie ← 3 * copie + 1
  FIN SI
  indice ← indice +1
FIN TANT QUE
```

```
RETOURNER indice
```

```
FIN FONCTION
```

Nième terme de la suite de Tribonacci

Enoncé du problème

Une suite de Tribonacci est une suite d'entiers où chaque terme est la somme des trois termes qui le précèdent :

- $T_0 = 0, T_1 = 1, T_2 = 1$
- Pour tout entier naturel $n \geq 3$ $T_n = T_{n-1} + T_{n-2} + T_{n-3}$

Spécification du problème

- Donnée d'entrée : i , entier (indice du terme recherché)
- Donnée de sortie : t , entier (valeur du terme)
- Pré-condition : $i \geq 3$ (on ne veut calculer qu'à partir du rang 3)
- Post-condition : la sortie t est égale au terme de rang i de la suite de Tribonacci, définie par $T_0 = 0, T_1 = 1, T_2 = 1$ et $T_n = T_{n-1} + T_{n-2} + T_{n-3}$ pour $n \geq 3$.

Signature de la fonction

- **tribonacci** (n : entier) : entier

Nième terme de la suite de Tribonacci

Corps de la fonction



T_0	T_1	T_2	T_3		
0	1	1	2	T_4	
	1	1	2	4	T_5
		1	2	4	7

- Structure de contrôle ?
- Variables nécessaires ?

Nième terme de la suite de Tribonacci

- Utilisation d'un **POUR** (nombre d'itérations connu)
- **4 variables** pour mémoriser le terme à calculer et les 3 précédents

```
FONCTION tribonacci (n : entier) : entier
```

```
VARIABLE i      : entier  
VARIABLE tm0    : entier  
VARIABLE tm1    : entier  
VARIABLE tm2    : entier  
VARIABLE r      : entier
```

```
tm0 ← 0  
tm1 ← 1  
tm2 ← 1  
r   ← 0
```

... (suite après)

Nième terme de la suite de Tribonacci

```
FONCTION tribonacci (n : entier) : entier
```

```
...
```

```
POUR i DE 3 A n PAR PAS DE 1
```

```
  r ← tm0 + tm1 + tm2
```

```
  tm0 ← tm1
```

```
  tm1 ← tm2
```

```
  tm2 ← r
```

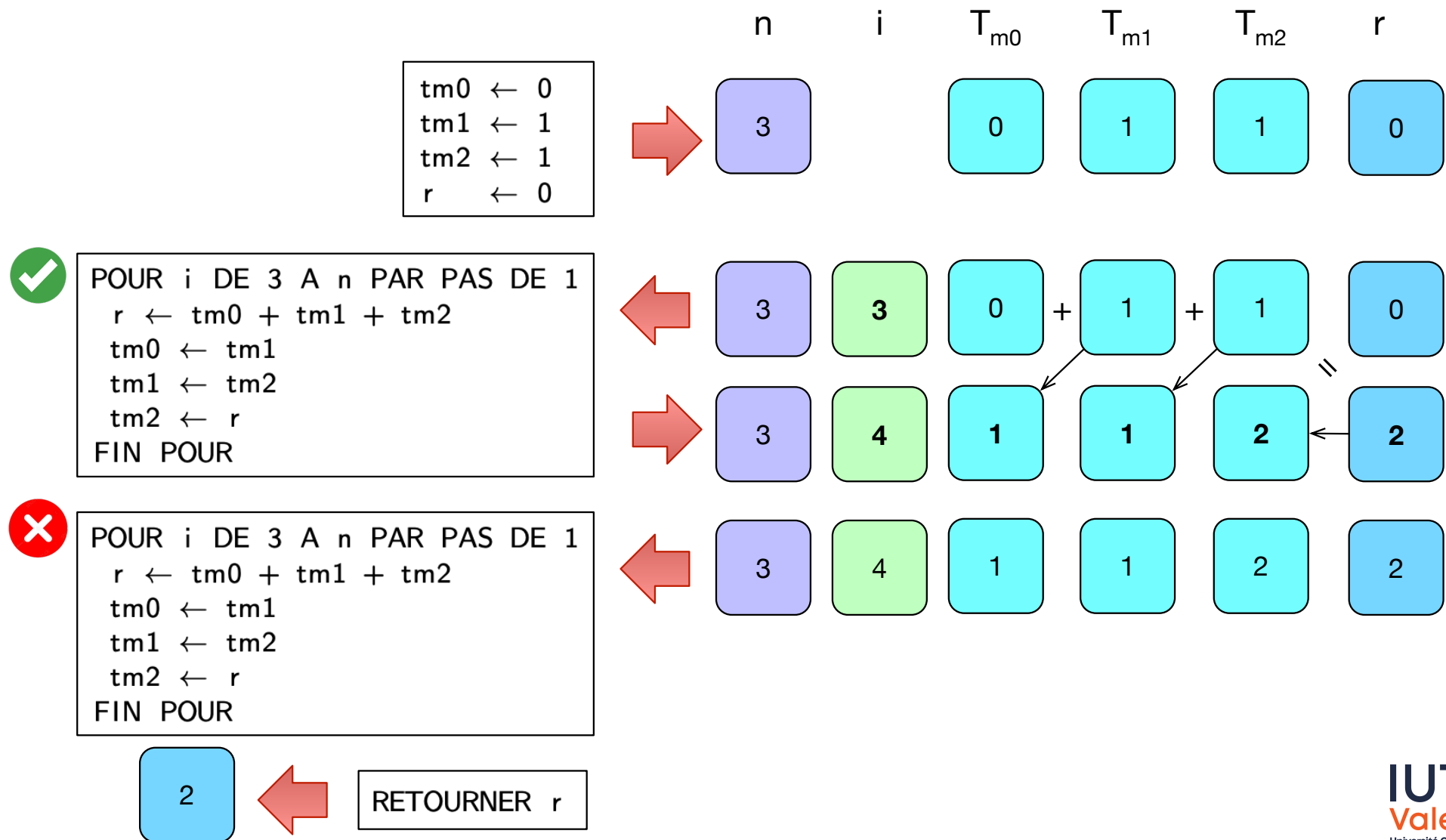
```
FIN POUR
```

```
RETOURNER r
```

```
FIN FONCTION
```

Nième terme de la suite de Tribonacci

- Déroulé de l'algorithme pour $n = 3$



Fin !

