25/11/2021

Valentin Kaelin et Jonathan Friedli

Rapport Laboratoire 7 : Hanoi

# Introduction

Dans le cadre de ce laboratoire n°7, nous avons dû implémenter un programme permettant de résoudre le problème des tours d’Hanoi de deux façons différentes : soit directement dans le terminal ou alors via une interface graphique. Dans les deux cas l’utilisateur peut choisir la taille de la tour initiale. L’implémentation utilise une pile que nous devons créer à la main.

# Diagramme des classes

TODO

# Description des classes

TODO

# Algorithme utilisé

Nous avons utilisé l’algorithme récursif, présenté notamment sur Wikipédia : [https://fr.wikipedia.org/wiki/Tours\_de\_Hanoï - Solution\_récursive](https://fr.wikipedia.org/wiki/Tours_de_Hano%C3%AF#Solution_r%C3%A9cursive)

Cette solution a l’avantage d’être simple à implémenter et très compréhensible. L’idée de cet algorithme est la suivante :

Pour déplacer une tour de *n* disques de A vers C, on effectue ces trois étapes :

1. Déplacer la tour des *n-1* premiers disques de A vers B (appel récursif).
2. Déplacer le plus grand disque de A vers C.
3. Déplacer la tour des *n-1* premiers disques de B vers C (appel récursif).

# Question 1

Le nombre de déplacements pour résoudre la tour de Hanoï vaut 2n-1, avec n le nombre de disque de la tour. Dans notre cas, n vaut 64 donc il faut 264-1 déplacements. Ce qui vaut 18'446'744'073'709'552'000 déplacements (merci Wikipédia).

Les moines faisant un déplacement par seconde il faudrait 18'446'744'073'709'552'000 secondes. L'univers ayant environ 13.8 milliards d'année et vu que les moines n’existaient pas (à ma connaissance) au début de l'univers. Nous allons supposer que le premier déplacement a eu lieu il y a exactement 13.8 milliards d'années. Nous soustrayons donc 13'800'000'000 \* 365,25 \* 24 \* 3600 à 18'446'744'073'709'552'000 ce qui donne :

18'446'744'073'709'552'000 - 435'494'880'000'000'000 = 18'011'249'193'709'552'000 secondes restantes ce qui représente environ 570'742'046'090 années. Donc environ **570,7 milliards d'années**.

# Choix de modélisation et d’implémentation

TODO

# Tests effectués

Nous avons tout d’abord réalisé le test demandé dans la donnée permettant de lancer

Par la suite, nous avons testé que les différents cas limites étaient bien pris en compte par notre programme. Voici la liste des tests réalisés :

|  |  |
| --- | --- |
| **Test** | **Résultat attendu** |
| Création d’une matrice avec un modulo nul ou négatif | Exception lancée dans le constructeur |
| Création d’une matrice avec un nombre de lignes ou de colonnes négatif | Exception lancée dans le constructeur |
| Opération entre deux matrices ne possédant pas le même modulo | Exception lancée lors de l’opération |
| Création d’une matrice avec des lignes de différentes tailles | Exception lancée dans le constructeur |
| Création d’une matrice avec des valeurs en dehors de l’intervalle [0, modulo - 1] | Exception lancée dans le constructeur |
| Création et opération sur une matrice vide | Les opérations se déroulent sans soucis. |
| Opération sur des matrices de tailles différentes | L’opération se déroule sans soucis, des 0 sont ajoutés au besoin. |

## Implementation de la stack

## Conclusion