SECTION: MIV
GROUPE: 1

RAPPORT MINI PROJET 2
Algorithmes de complexité temporelle exponentielle

BENATALLAH Rayan Ibrahim
LAGAAB Aya Céline

TOURS HANOI

1- l'Algorithme récursif

```
Procédure tourHanoi(entier: n, caractere D,A,I)
Début

   si ( n = 1 ) alors
      écrire("Disque 1 de "D" a "A);
   sinon
      toursHonoi(n-1, D, I, A);
      écrire("Disque "n" de "D" a " A);
      toursHonoi(n-1, I, A, D);

Fin
```

2- Sa complexité

2-1-temporelle: Sachant que c'est un algorithme récursif, sa complexité est de **O(2^n)**

2-2- spatiale: on ne stock pas les disques, on les affiche directement donc la complexité spatiale est de **O(2^n)** comme mémoire réservé pour chaque n

3-1- Programme C:

```c
#include<stdio.h>
#include<stdlib.h>


/*
n : nombre de disques utilisés
D : emplacement de départ
A : emplacement d'arrivée
I : emplacement intermédiaire
*/

void toursHanoi(int n, char D, char A, char I) {
  if (n == 1)
    printf("Disque 1 de %c a %c \n" , D , A);
  else {
    // D à A
    toursHanoi(n - 1, D, I, A);
    printf("Disque %d de %c a %c \n", n , D ,A);
    //I à A
    toursHanoi(n - 1, I, A, D);
  }
}
```

BENATALLAH Rayan Ibrahim
LAGAAB Aya Céline

3-2- Programme java:

```java
import java.io.*;
import java.math.*;
import java.util.*;
class GFG {
  static void toursHonoi(int n, char D,char A, char I)
  {
    if (n == 0) {
      return;
    }
    toursHonoi(n - 1, D, I, A);
    System.out.println("Disque " + n + " de "+ D + " a "+ A);
    toursHonoi(n - 1, I, A, D);
  }
  // Driver code
  public static void main(String args[])
  {
    int N = 5;
    // A, B and C are names of rods
    toursHonoi(N, 'A', 'C', 'B');
  }
}
```

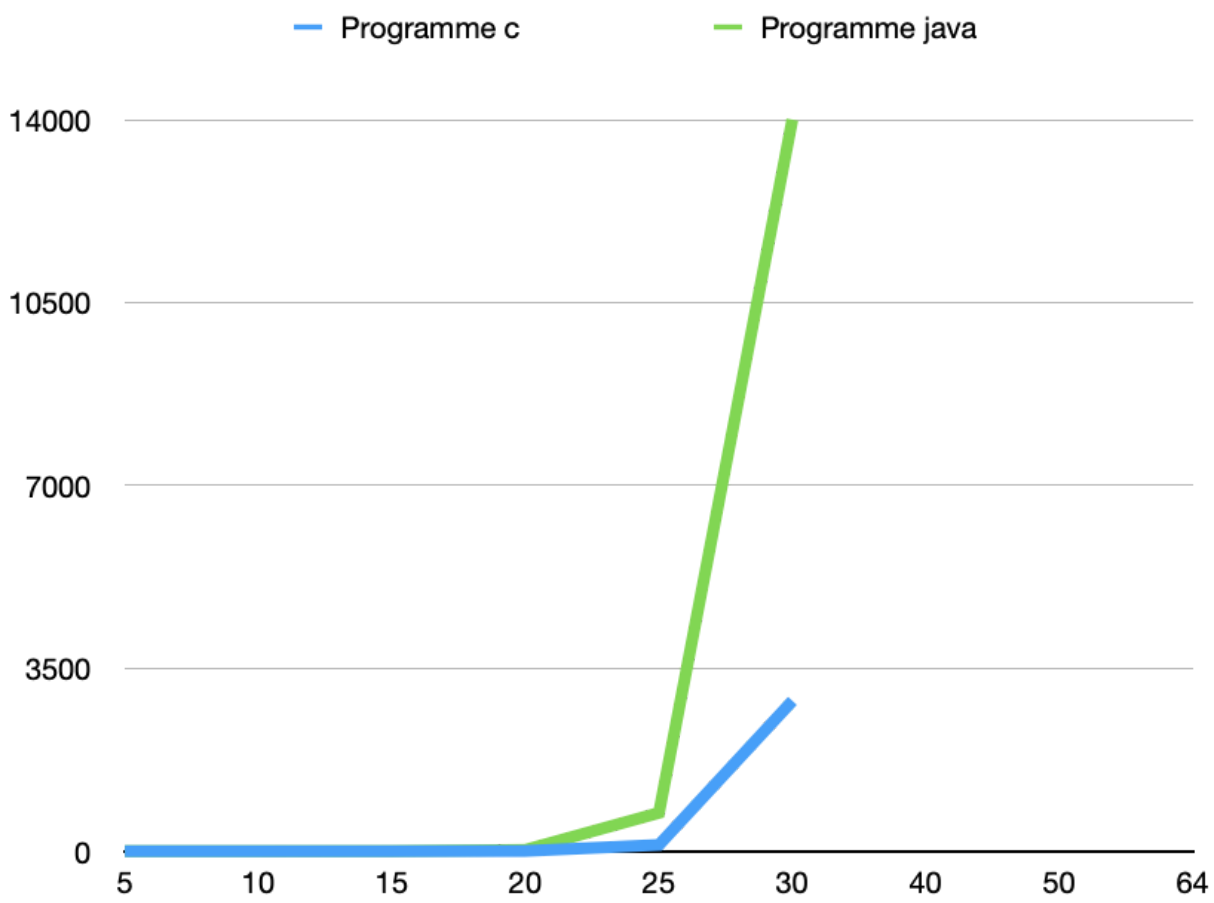
4- représentation des temps d'exécution en tableau

Grâce à l'exécution du programme C et java vu précédemment nous avons obtenu les temps d'exécution suivants:

N	5	10	15	20	25	30	40	50	64
Programme c	0,000079	0,004278	0,118660	3,717193	101,029591	3232,046564	..	..	..
Programme java	0,003	0,148	1,6	25	732,795	..	..	..	..

5-

6- Représentation graphique des résultats obtenu en dessus:



7- interprétation des résultats:

7-a-

Les mesures de temps obtenu correspondent au cas exact,il nous paraît clair de part le graphe que les temps d'exécutions suivent la même évolution, **en O(2^n)**.

7-b-

En prenant les résultats obtenus grâce à l'exécution du programme C, on remarque que les temps d'exécution sont approximativement multipliés par **2^5** à peu près lorsque **N** est incrémenté de **5**.

On en déduit que le temps d'exécution est proportionnel à N, Nous ne pouvons pas généraliser car les tests que nous avons fait n'englobe pas toutes les valeurs possibles

7-c-

si on remplace a chaque fois le N dans 2^n et on schématise les résultats graphiquement, la complexité théorique et expérimentale seront du même ordre de grandeur que la complexité théorique, donc le modèle théorique est conforme aux mesures expérimentales.

BENATALLAH Rayan Ibrahim
LAGAAB Aya Céline