1.5em

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université des Sciences et de la Technologie Houari Boumediene
Faculté d'informatique
Département d'informatique

Rapport du projet de TP

Module : Complexite

Master 1 IV

Algorithme de complexite temporelle exponentielle

- Réalisé par :
  kASHI Thiziri
  MESSAOUDENE Lydia

1. Premiere partie

1.1. Algorithme recursif pour resoudre le probleme des "Tours de Hanoi"

Algorithm 1 Tours Hanoi(n,d,a,i)

Require: $Nb\_disques(N), Tour\_depart(D), Tour\_arrivee(A), Tour\_intermediaire(I)$
  if $Nb\_disques = 1$ **then**
    print("Disque Tour_depart a Tour_arrivee")
  else
    Tours_Hanoi($Nb\_disques - 1, D, I, A$)
    print("Disque Nb_disques de D a A ")
    Tours_Hanoi($Nb\_disques - 1, I, A, D$)
  end if

1.2. Complexite

1.2.2. complexite temporelle :

CT(n)=$O(2^n)$
La complexité temporelle est Quadratique donc elle est de $O(2^n)$, où n est le nombre total de disques. Si n = 64 et qu'un disque est déplacé par seconde, l'utilisation du plus petit nombre de déplacements disponibles prendra $2^{64} - 1$ secondes, ou environ 585 milliards d'années pour finir le jeu.

1.2.1. complexite spaciale :

CT(n)=$O(1)$

1.3. Code en C

Fichier .c

1.4. Temps d'execution

n	10	11	12	13	14	15	16	17	18	19
T	0.000000	0.031250	0.187500	0.343750	0.625000	1.000000	2.156250	3.265625	5.984375	12.406250

n	20	21	22	23	24	25	26	27
T	23.218750	45.359375	91.296875	173.375015	463.296875	682.359375	1600.421875	3033.531250

n	28	29	30	31	32	...	64
T	+3033.531250	+3033.531250	+3033.531250	+3033.531250	+3033.531250	..	+3033.531250

1.5.Graphe de vartiation de la complexite temporelle et du temps d'execution

1.5.1.Graphe de vartiation de la complexite temporelle :

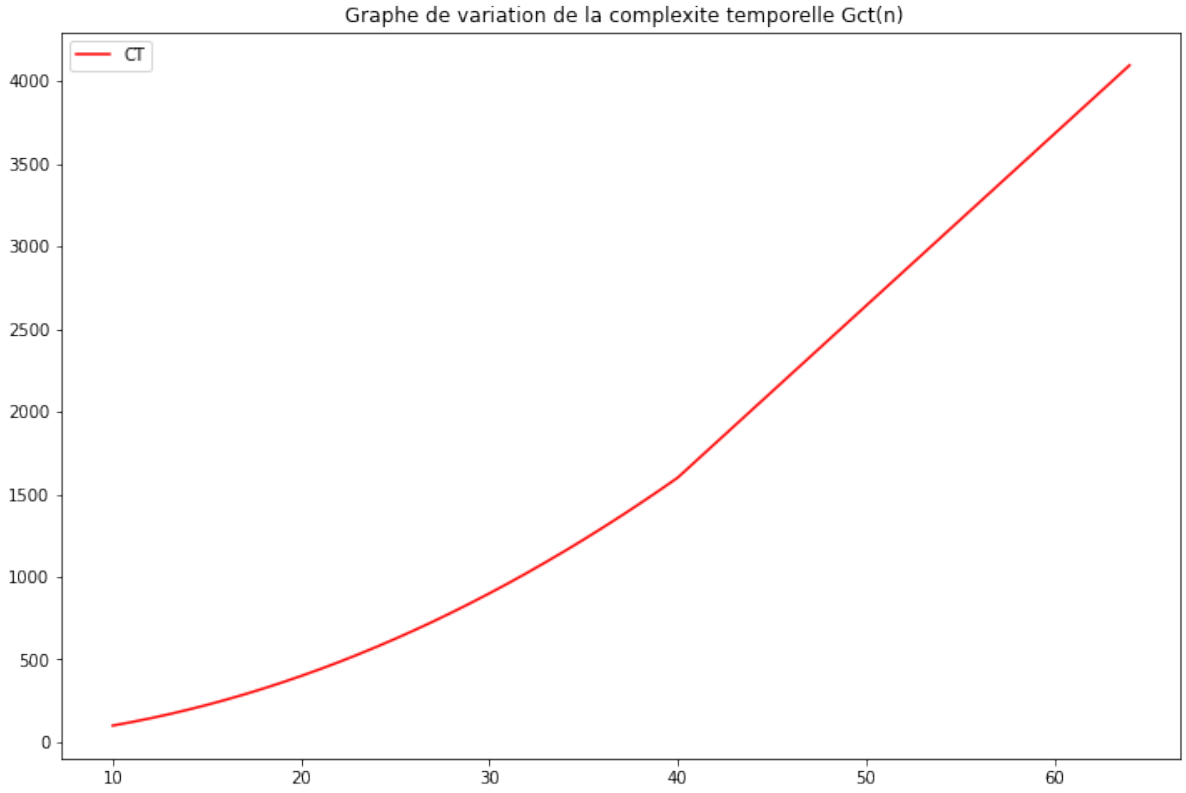

FIGURE 1 – Graphe de vartiation de la complexite temporelle $G_{ct}(n)$

1.5.2.Graphe de vartiation du temps d'execution :

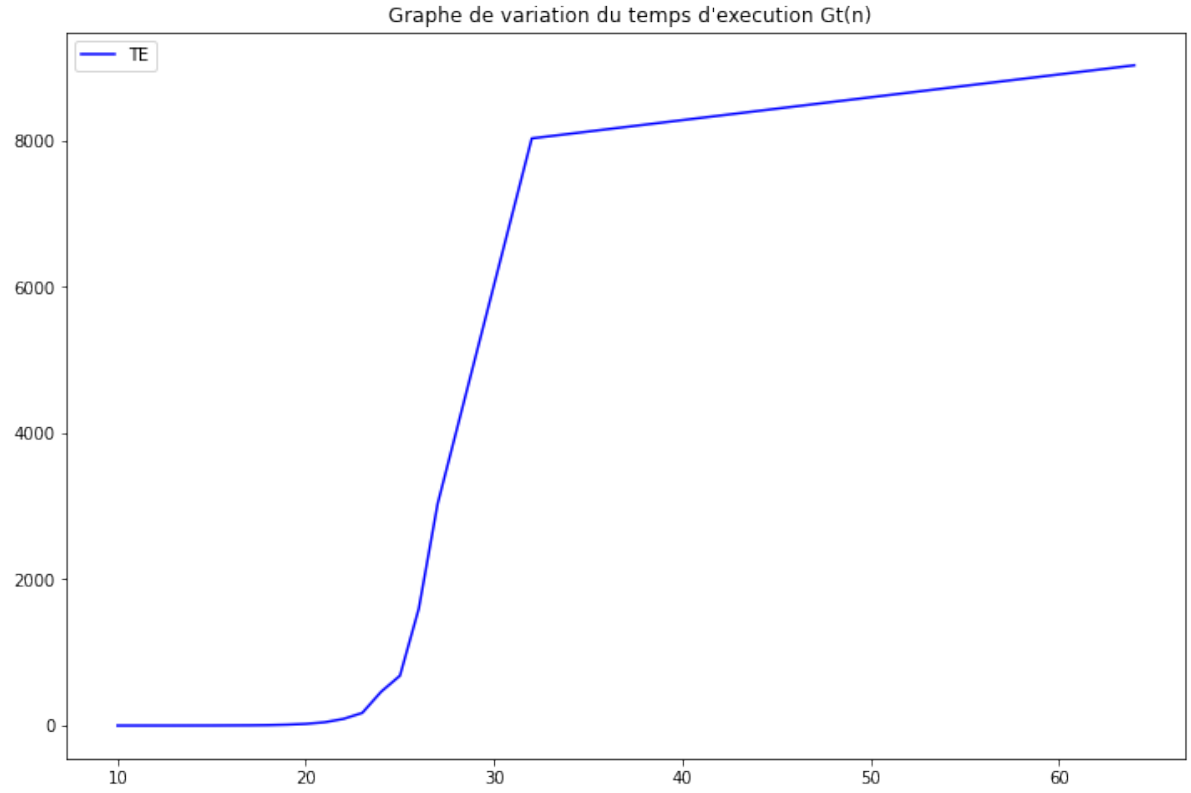

FIGURE 2 – Graphe de vartiation du temps d'execution $G_t(n)$)

1.7.Interpretation des resultats

1.7.1. a quoi corespendent les mesures de temps :

dans les tours de hanoi , il n y'a pas de pire , moyen ou meuilleur cas .

1.7.2. comparaison :

- $n_{suivant} = n_{precedant} + 1 compte à T(n) nous remarque qu'il augmented'une façon exponentielle pour n = 14 T est égal à 0.03 et pour n = 14 T' est égale à 0.62, nous pouvons également comparer l'évolution de T pour l'intervalle n = [18, 27] nous pouvons remarquer que n'a aucun moment T diminue et qu'il augmented'une façon considérable$
-

T(n)=10.5806+1.4780298860203$\times 10^6 * e^{0.801676x}$

1.8. temps d'execusion en java(seconde) :

n	10	11	12	13	14	15	16	17	18	19
T	0.000000	0.000000	0.000000	1	2	5	10	19	41	92

n	20	21	22	23	24	25	26	27
T	185	352	1121	2221.05489	3628.0892	4828.359375	+5000	+5000

1.9.Graphe de vartiation de la complexite temporelle et du temps d'execution EN JAVA

1.5.1.Graphe de vartiation de la complexite temporelle EN JAVA :

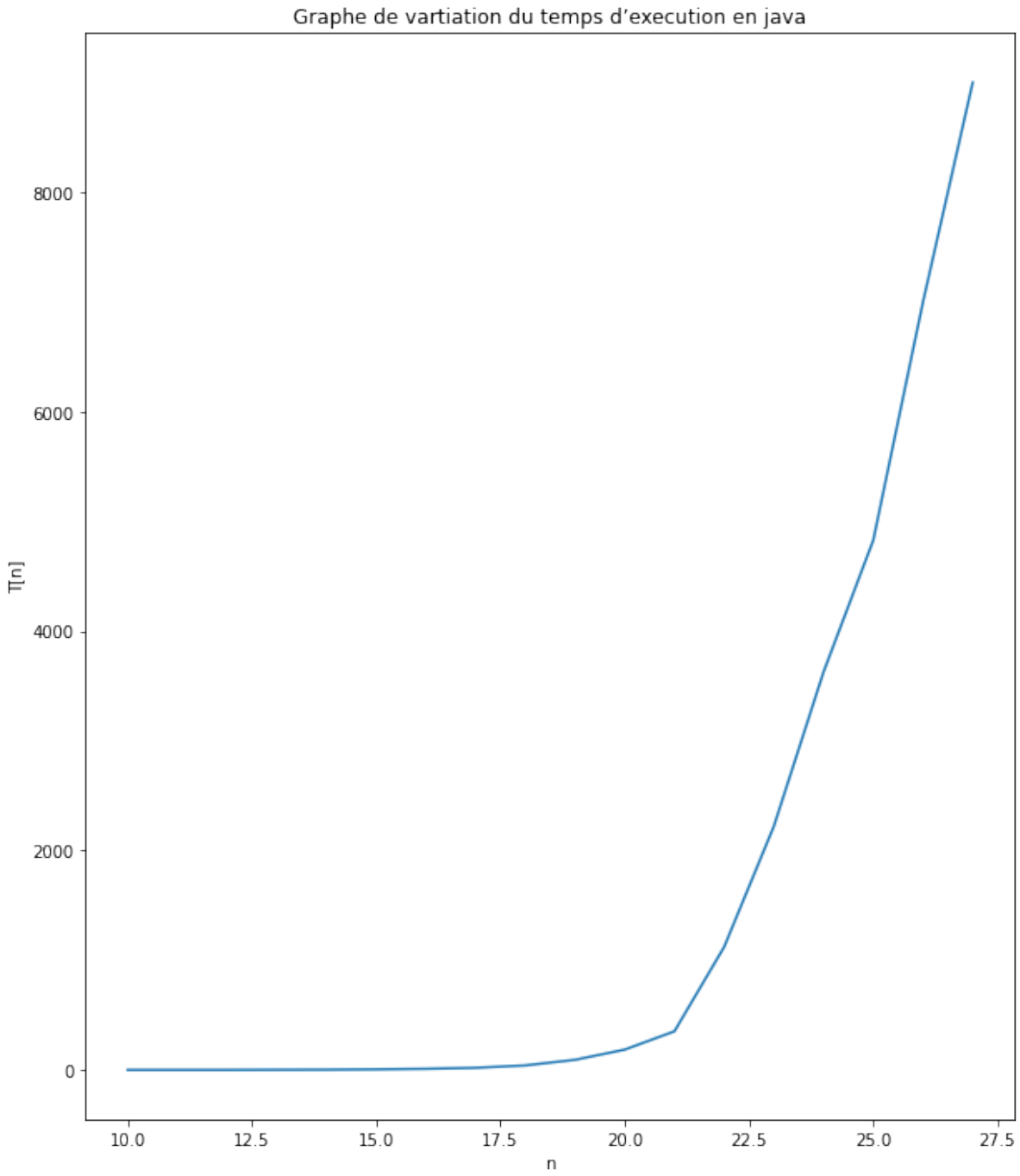

FIGURE 3 – Graphe de vartiation de la complexite temporelle $G_{ct}(n)$