

SAE 1.02 : Comparaison d'approches algorithmiques

22/01/2023

BASOL Nathan LEDERREY Lussandre
INFO1A2

Introduction

Dans cette SAE, nous reprenons le jeu « Grundy » que nous avons codé pendant la SAE d'octobre (S1.01).

On part d'une classe de départ qui contient une méthode jouerGagnant qui appelle une méthode récursive estPerdant. Comme son nom l'indique, cette méthode va choisir le meilleur coup à jouer pour l'IA.

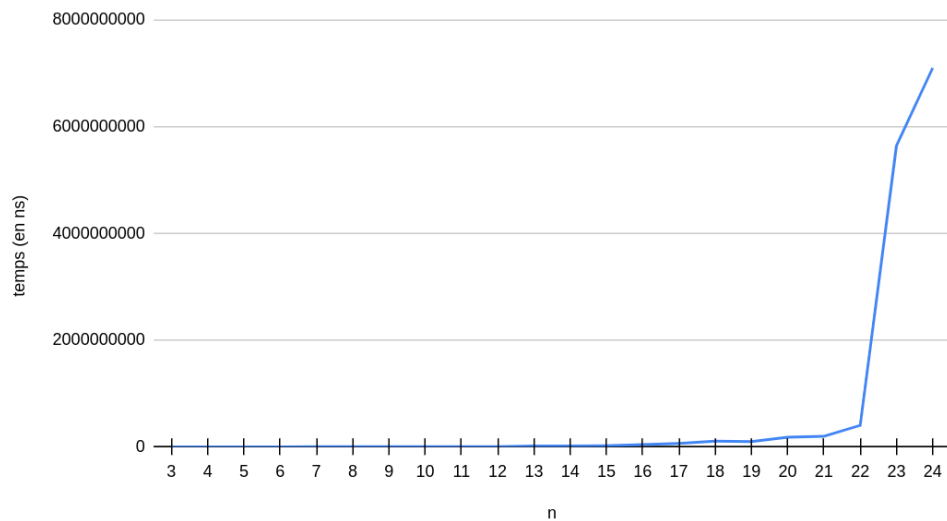
On étudiera à l'aide de tableaux et de graphiques l'efficacité de la méthode estGagnante en fonction du nombre d'allumettes de départ (évolution du temps d'exécution et de la variable cpt en fonction de n).

Nous allons réaliser plusieurs classes différentes qui vont à chaque fois améliorer la précédente pour optimiser le code afin que l'IA soit plus rapide pour jouer le meilleur coup.

À noter que nous avons créé une méthode « testEstGagnanteEfficacite » pour calculer le temps (en nanoseconde) et le cpt (de la méthode estPerdante). Dans cette méthode, nous avons appelé la méthode estGagnante pour vérifier que l'IA joue bien le bon coup (quand ceci est possible) pour un nombre d'allumettes de départ de plus en plus grand.

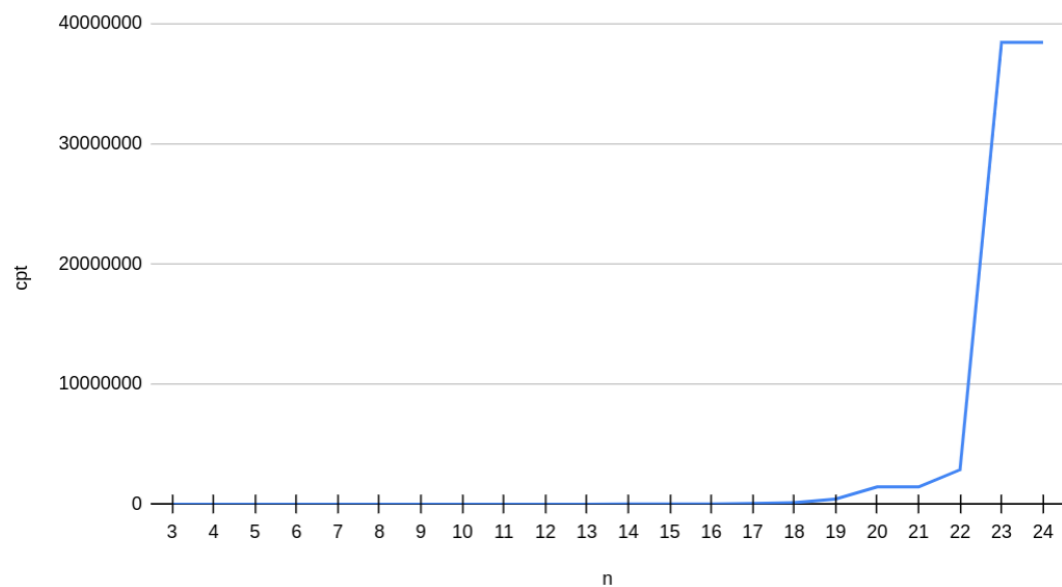
Grundy version 0

temps (en ns) par rapport à n



- De 3 à 16 allumettes, la courbe est à peu près constante. Puis, elle va légèrement croître entre 16 allumettes et 22. À partir de 22 allumettes, la courbe va croître très rapidement pour atteindre environ 6 secondes pour $n = 23$ et $n = 24$.

cpt par rapport à n



- De 3 à 18 allumettes, la courbe est à peu près constante. Puis, elle va légèrement croître entre 18 allumettes et 22. À partir de 22 allumettes, la courbe va croître très rapidement pour atteindre un cpt de environ 40 000 000 pour $n = 23$ et $n = 24$. Le cpt passe d'une valeur raisonnable pour $n = 22$, mais sa valeur augmente considérablement pour une allumette de plus.

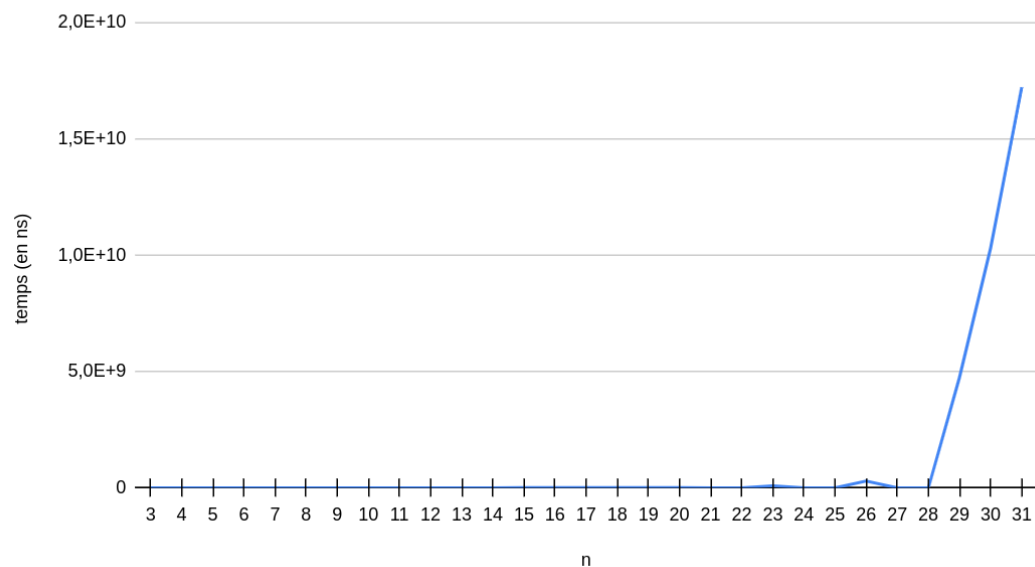
n	temps (en ns)	cpt	Joue le bon coup ?
3	71329	1	
4	64244	2	non
5	92513	3	
6	170114	7	
7	488773	18	non
8	557241	19	
9	835862	39	
10	1871706	112	non
11	1535598	113	
12	3722672	227	
13	13556228	1267	
14	15670199	2559	
15	17858945	6527	
16	41790182	20141	
17	63055426	62801	
18	104401897	128101	
19	97556360	422885	
20	179596566	1433262	non
21	195181901	1433263	
22	402421312	2866527	
23	5641087740	38440186	
24	7096208841	38440187	non

- Le temps d'exécution est inférieur à 1 seconde jusqu'à 22 allumettes. Le temps d'exécution a été multiplié par environ 15 entre 22 et 23 ce qui est énorme. Le cpt aussi évolue très rapidement d'un n à un autre. On passe d'un cpt de 1 432 262 pour 20 allumettes à un cpt de 38 440 187 pour 24 allumettes.

En conclusion, cette première version de Grundy n'est efficace que pour un nombre d'allumettes inférieur à 22. Pour un nombre d'allumettes supérieur à 22, le cpt sera beaucoup trop élevé pour que le temps d'exécution soit raisonnable. On note que le cpt et le temps d'exécution augmentent beaucoup trop rapidement.

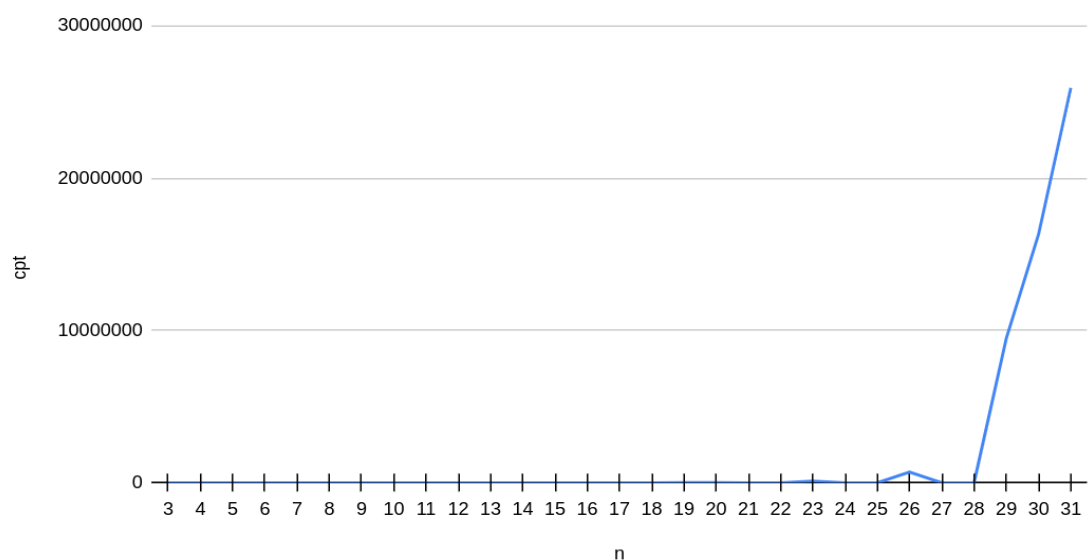
Grundy version 1

temps (en ns) par rapport à n



- De 3 à 28 allumettes, la courbe est à peu près constante. À partir de 28 allumettes, la courbe va croître très rapidement pour atteindre environ un temps d'exécution de 20 secondes pour 31 allumettes. Encore une fois, le temps d'exécution augmente très rapidement en passant de 28 à 29 allumettes. On le voit très bien avec la courbe qui croît d'un coup.

cpt par rapport à n



- La courbe du cpt en fonction de n a la même allure que la courbe du temps. De 3 à 28 allumettes, la courbe est constante avec une valeur assez faible pour le cpt. À partir de 28 allumettes, la courbe va croître pour atteindre un cpt d'environ 25 millions pour 31 allumettes.

n	temps (en ns)	cpt	Joue le bon coup ?
3	483700	1	
4	39800	2	non
5	42600	1	
6	30400	3	
7	147100	12	non
8	12200	1	
9	20500	3	
10	176400	26	non
11	12800	1	
12	30200	3	
13	1058200	103	
14	335200	57	
15	1556000	235	
16	1193500	313	
17	2548400	619	
18	2273400	1055	
19	4276000	3069	
20	9761700	5572	non
21	16300	1	
22	22400	3	
23	77748400	101036	non
24	21200	1	
25	21400	3	
26	282944600	700008	non
27	11200	1	
28	26000	3	
29	4730075600	9487607	
30	10291128300	16309853	
31	17221717200	25927287	

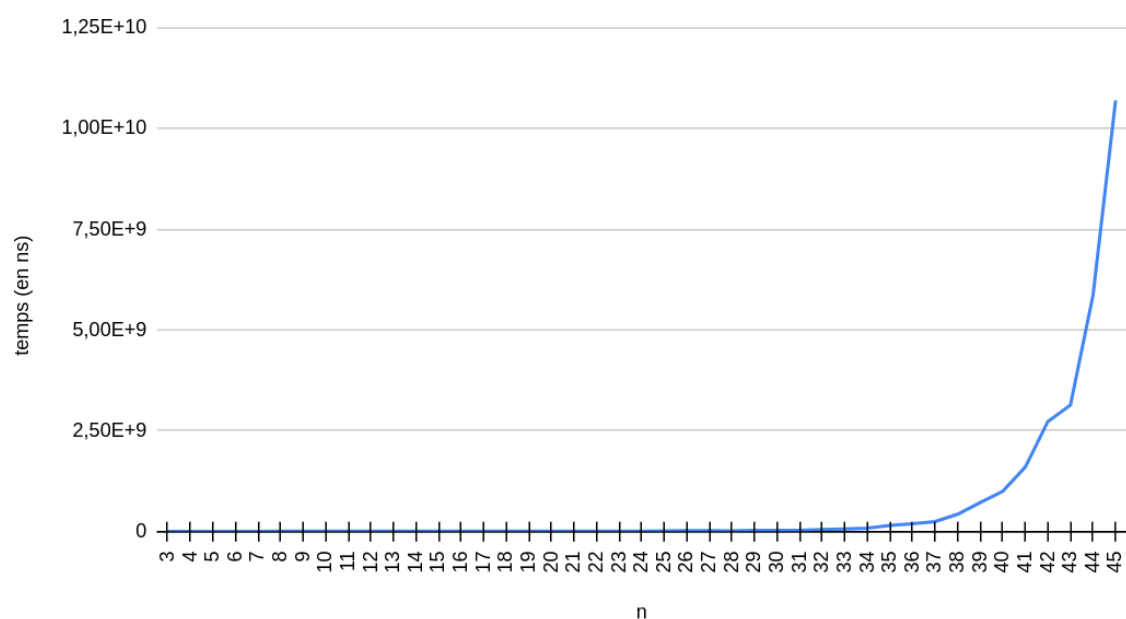
- On voit bien avec ce tableau que notre programme utilise le tableau posPerdantes. Pour 26 allumettes, le cpt est à 700 008. Cependant, pour 27 allumettes, le cpt est seulement à 1. Le programme a donc du tomber sur des décompositions déjà connues perdantes et n'a donc pas procédé à une décomposition.

En conclusion, cette version améliore le temps d'exécution pour un nombre d'allumettes de départ plus élevé qu'avant. À partir de 29 allumettes, le temps d'exécution devient beaucoup trop long.

La version 1 est donc plus efficace que la version 0. Le temps d'exécution est presque immédiat pour 28 allumettes contre 23 pour la version 1. À noter que le cpt est toujours très élevé dans cette version pour des n supérieur à 28.

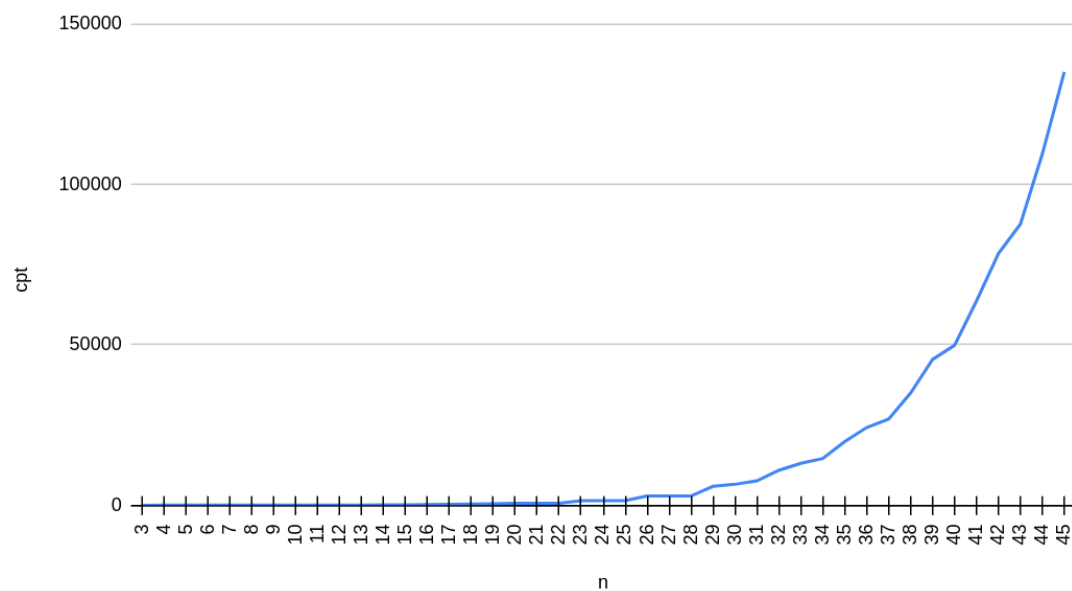
Grundy version 2

temps (en ns) par rapport à n



- De 3 à 34 allumettes, la courbe est à peu près constante. Contrairement aux versions précédentes, la courbe ne croit pas d'un coup à partir d'un certain nombre d'allumettes. Ici, elle commence à croître plutôt légèrement de 34 à 43 allumettes (entre une seconde et 3 secondes ce qui est assez raisonnable). Après 43 allumettes, la courbe croit très fortement pour atteindre environ 10 secondes de temps d'exécution pour 45 allumettes.

cpt en fonction de n



- De 3 à 28 allumettes, la courbe est à peu près constante. Ensuite, la courbe va croître de façon progressive de 28 allumettes jusqu'à 40 (cpt inférieur à 5000). Après 40 allumettes, la courbe croît fortement pour atteindre un cpt de presque 150 000 pour 45 allumettes.

n	temps (en ns)	cpt	Joue le bon coup ?
3	20300	1	
4	35400	2	non
5	26800	3	
6	39500	5	
7	89300	11	non
8	97800	12	
9	282400	14	
10	218300	29	non
11	258900	30	
12	255200	32	
13	772900	79	
14	524200	101	
15	1090300	146	
16	773900	185	
17	1045000	272	
18	1179400	330	
19	1549700	472	

20	2125200	609	non
21	1830200	610	
22	1751700	612	
23	4617800	1435	non
24	4152400	1436	
25	7570200	1438	
26	17287200	2863	non
27	15534700	2864	
28	9987300	2866	
29	27027800	5894	
30	24505900	6522	
31	23019100	7614	
32	47098700	10883	
33	59525800	13090	
34	76185600	14561	
35	144661200	19843	
36	186182400	24232	
37	241756200	26842	
38	423958500	34986	
39	713109900	45431	
40	994166900	49849	
41	1596542900	63662	
42	2726581100	78461	
43	3136020900	87623	
44	5861356100	109503	
45	10696396700	135019	

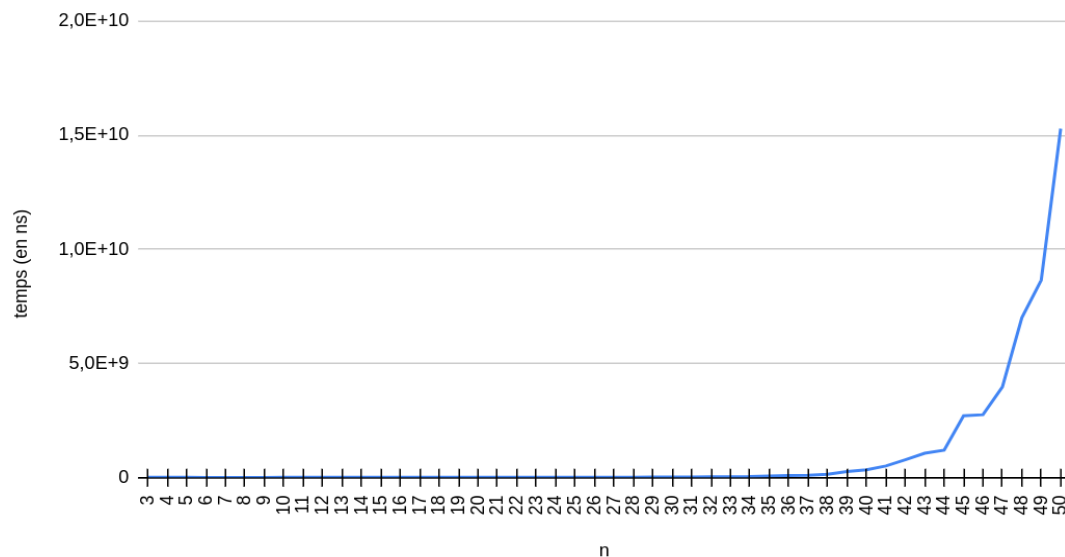
• Ce tableau de la version 2 est déjà très différent de la version 1. On voit bien que le cpt n'augmente pas d'un coup mais de façon progressive. De plus, les valeurs du cpt sont plutôt raisonnable comparé à la version 1. On commence à dépasser les une secondes pour 40 allumettes.

En conclusion, cette version améliore le temps d'exécution pour un nombre d'allumettes de départ plus élevé qu'avant. Jusqu'à 37 allumettes, le temps est presque instantané. Pour 43 allumettes, l'IA mettra 3 secondes pour jouer ce qui est encore pas mal. Au-delà, le temps d'exécution sera trop long

La version 2 est donc plus efficace que la version 1. On note que les valeurs du cpt sont beaucoup moins élevées qu'avant ce qui montre que le programme exécute moins de fois la méthode récursive. Le programme est donc plus optimisé.

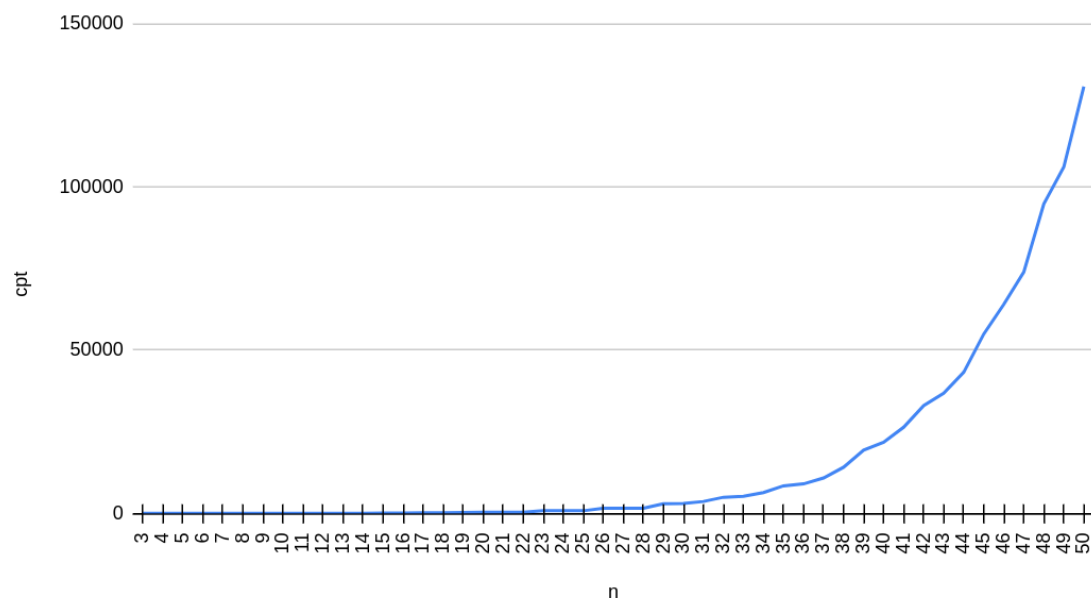
Grundy version 3

temps (en ns) par rapport à n



- De 3 à 37 allumettes, la courbe est à peu près constante. Tout comme dans la version 2, la courbe va croître légèrement cette fois-ci de 37 à 47 allumettes (environ 4 secondes pour 47). Après 47 allumettes, la courbe croît très fortement pour atteindre environ 15 secondes de temps d'exécution pour 50 allumettes.

cpt en fonction de n



- De 3 à 30 allumettes, la courbe est à peu près constante. Ensuite, la courbe va croître de façon progressive de 28 allumettes jusqu'à 40 (cpt inférieur à 5000). Après 40 allumettes, la courbe croît fortement pour atteindre un cpt de presque 150 000 pour 45 allumettes.

n	temps (en ns)	cpt	Joue le bon coup ?
3	49700	1	
4	58400	2	non
5	197700	3	
6	106100	5	
7	116600	11	non
8	110000	12	
9	143800	14	
10	390600	24	non
11	391800	25	
12	309000	27	
13	859100	69	
14	758500	75	
15	1071500	114	
16	1128200	147	
17	1332900	180	
18	1334900	232	
19	1869800	312	
20	2554900	376	non
21	2418400	377	
22	3088400	379	
23	6743600	816	non
24	5608700	817	
25	3576900	819	
26	7129200	1575	non
27	5163300	1576	
28	6047500	1578	
29	12235700	3001	
30	16533600	3034	
31	16984600	3664	
32	25726600	4963	
33	30330400	5283	
34	34993100	6399	
35	62439200	8469	
36	78113700	9052	
37	92369700	10857	
38	134416400	14153	
39	253129400	19429	
40	334343800	21792	
41	500893600	26412	
42	770484300	33068	
43	1059380200	36833	

44	1192045700	43261
45	2699970700	54972
46	2746326800	64053
47	3967893000	73983
48	6998521600	94829
49	8648197400	106158
50	15286569900	130788

- Ce tableau de la version 3 ressemble à celui de la version 2. Le cpt augmente toujours de manière progressive. Cependant, le programme semble plus optimisé qu'avant car le cap des une seconde a ici été atteint pour $n = 43$ contre 40 allumettes pour la version 2.

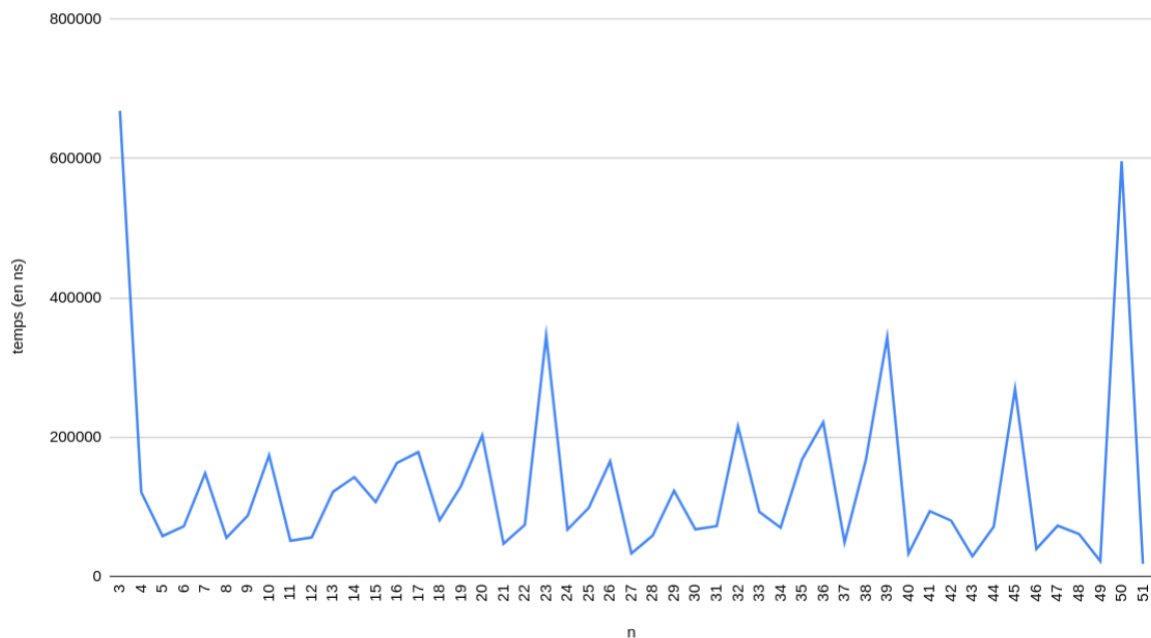
En conclusion, cette version améliore le temps d'exécution pour un nombre d'allumettes de départ un peu plus élevé qu'avant (+ 5 allumettes).

La version 3 est donc légèrement plus efficace que la version 2 . Les valeurs du cpt sont moins élevés pour des n équivalents. Pour $n = 45$, le cpt était de 135 000 pour la version 2 contre 55 000 pour cette nouvelle version. Le fait de supprimer les tas perdants a donc eu un impact sur l'efficacité de la méthode estGagnante.

Grundy version 4 (graphiques et tableau vont jusqu'à 51)

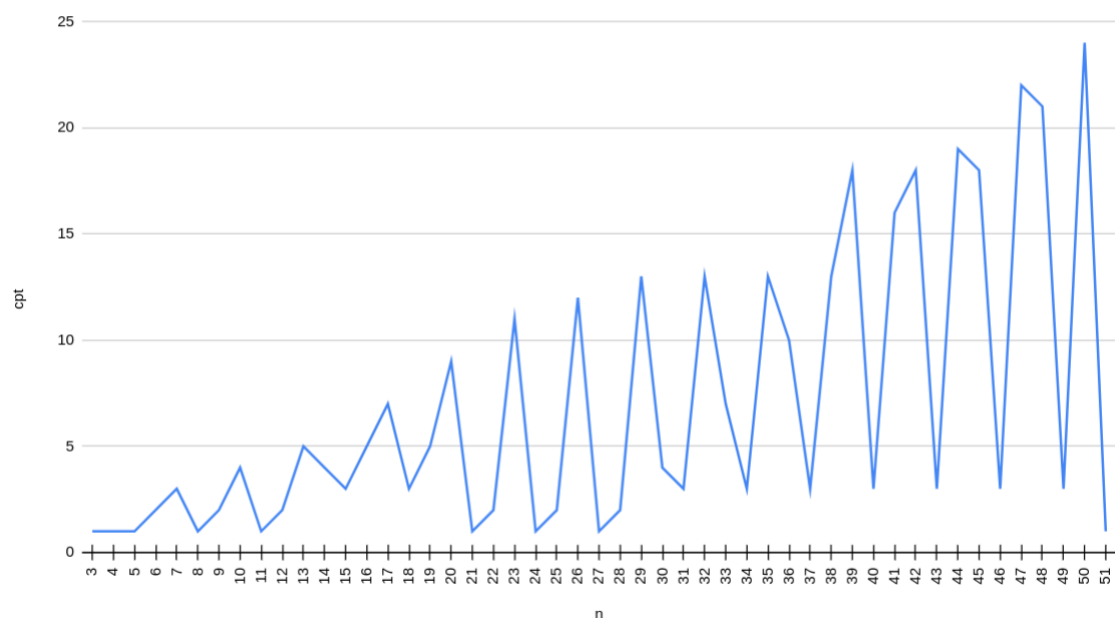
si on voudrait aller au-delà de 50 décompositions, il faudrait déterminer pour chacune des décompositions désirées leur type, ce qui requiert des calculs mathématiques complexes.

temps (en ns) par rapport à n



- Cette courbe est très différente des autres versions. Premièrement, on voit que la fourchette de temps est entre 0 et 600 000 nanosecondes pour 51 allumettes ce qui est extrêmement peu comparé à la version précédente (15 secondes pour 50 allumettes). Également, la courbe n'est pas strictement croissante mais elle oscille pour des n de plus en plus grands.

cpt par rapport à n



- La courbe du cpt est aussi très différente de la courbe de la version 3. Premièrement, on voit que la fourchette de cpt est entre 0 et 25 pour 51 allumettes. C'est encore une fois très peu (pour $n = 50$, le cpt était à 130 000 pour la version 3 contre 25 ici). Aussi, la courbe n'est pas strictement croissante mais elle oscille pour des n de plus en plus grands. Cependant, on voit bien qu'il y a une tendance légèrement croissante.

n	temps (en ns)	cpt	Joue le bon coup ?
3	667900	1	
4	121200	1	non
5	58100	1	
6	72000	2	
7	148300	3	non
8	55500	1	
9	87300	2	
10	174100	4	non
11	51400	1	
12	56100	2	
13	121700	5	
14	142600	4	
15	106800	3	
16	162900	5	
17	178500	7	
18	80700	3	
19	129600	5	
20	202300	9	non

21	47000	1	
22	74300	2	
23	345400	11	non
24	67400	1	
25	98700	2	
26	165300	12	non
27	33100	1	
28	58900	2	
29	123100	13	
30	67700	4	
31	72400	3	
32	215400	13	
33	92700	7	
34	70100	3	
35	167700	13	
36	221100	10	
37	49100	3	
38	167400	13	
39	343300	18	
40	32900	3	
41	93700	16	
42	80200	18	
43	29000	3	
44	71600	19	
45	269100	18	
46	39700	3	
47	72900	22	
48	61000	21	
49	22100	3	
50	595900	24	
51	18200	1	

En conclusion, cette version améliore considérablement la version 3. On le voit notamment avec les valeurs de cpt qui sont très faible pour un nombre d'allumettes élevé mais aussi avec le temps d'exécution qui relève de l'instantané pour des grands n.