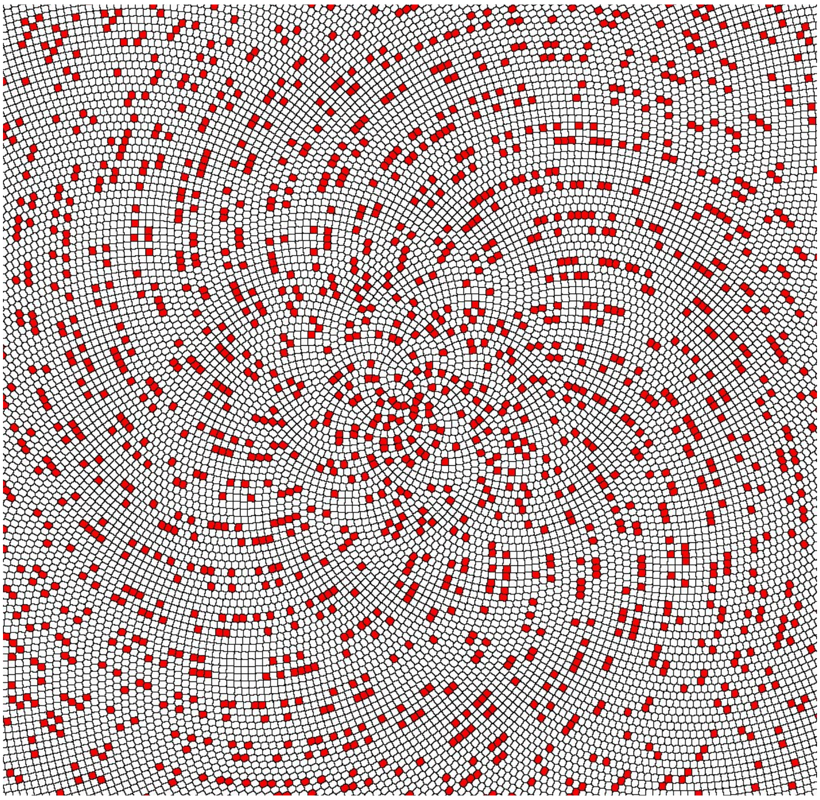
EFREI L1 TOPC
PJ 1 - Application des bases de l'algorithmique

Etude de la conjecture de Teller

v3 du 29 octobre 2016 - Franck Lepoivre & Patrick Teller

Le sujet porte sur l'étude théorique et pratique d'une conjecture de théorie des nombres.



Préambule

Finalité du PJ

Ce projet a pour buts pédagogiques :

1. de vous faire appliquer vos compétences de base en algorithmique et programmation (jusqu'aux boucles, tableaux facultatifs).
2. de vous faire ressentir et comprendre le lien filial et fraternel entre la mathématique et l'informatique.

Vous allez pour cela vous attaquer à l'étude d'un problème ouvert de théorie des nombres. Qui sait, peut-être que l'une ou l'un d'entre-vous en viendra à bout ?

Pré-requis

- Avoir suivi et participé activement aux CM > TD > TP d'arithmétique, d'algorithmique et de programmation, en complétant cette participation d'un travail personnel soutenu.

Sujet

Etant donné un entier $x > 1$, étant donné la fonction t qui à x associe la somme de ses diviseurs sauf 1. En appliquant itérativement t, on produit la suite d'entiers $x,\ t(x),\ t \circ t(x),\ t^3(x),\ \cdots, t^k(x)$.

Patrick Teller, enseignant à l'EFREI, forme la conjecture selon laquelle cette suite converge nécessairement vers un nombre premier ($\forall x \in N, lim_{k \to \infty}\ t^k(x) \in P$).

Le but de votre projet est de vérifier cette conjecture aussi loin que possible (ou la réfuter avec un contre-exemple).

Pour en savoir plus sur la proposition de Patrick Teller :
[Does iterating a certain function related to the sums of divisors eventually always result in a prime value?]

Cahier des charges impératif

Il s'agit de produire un programme informatique et un rapport d'étude que vous présenterez le jour de la soutenance et livrerez aux enseignants en charge de votre évaluation. Le sujet précise nos attentes suivant les deux axes de l'étude théorique (à présenter dans votre rapport) et informatique (fonctionnalités à réaliser et algorithmes à présenter dans votre rapport). Les questions qui devront impérativement être traitées sous l'angle théorique sont préfixées de ET (Etude théorique).

Les bases théoriques

On note t l'application qui à tout entier qui à $x > 1$ associe la somme de ses diviseurs strictement supérieurs à 1; ainsi *t*(5) = 5, *t*(6) = 11, etc.

ET > Démontrez que pour tout entier $x > 1$, x est premier ssi $x = t(x)$.

On notera t^n la composée *n* fois de *t* par *t*; ainsi $t^4(12) = t^3(27) = t^2(39) = t(55) = 67$.

La conjecture PT est: $\forall x \in N \setminus \{0,\ 1\}, \exists p \in N,\ t^p(x) \in P$, ou, ce qui revient au même, $\forall x \in N \setminus \{0,\ 1\}, \exists p \in N,\ t^p(x) = t^{p+1}(x)$; pour chaque x pour lequel cette conjecture est vraie on appellera $H(x)$ la valeur $t^p(x) = t^{p+1}(x)$ et on désignera par $r(x)$ (durée de vol) la valeur de p, c'est à dire le nombre d'itérations nécessaires pour atteindre un premier.

Remarquons que si x est premier $x = t(x)$ donc $r(x) = 0$.

Le minimum à réaliser pour espérer avoir la moyenne

1. Concevez et réalisez un premier algorithme appelé `sigma_v1` qui calcule et retourne la somme $\sigma(x)$ des diviseurs d'un entier positif x (y compris le diviseur 1), en testant chacun des entiers compris entre 1 et x.
2. Utilisez la fonction `sigma_v1` pour définir la fonction $t(x) = \sigma(x) - 1$ que vous nommerez `teller`.
    a. **ET >** t est-elle croissante (i.e. $x < y \Rightarrow t(x) < t(y)$) ? Expérimentez puis énoncez une réponse et démontrez là.
3. Utilisez la fonction `teller` pour définir la fonction `iterated_teller` qui retourne la limite de son application itérée à un entier. On appelle cette limite l'héritier de x et on le note $H(x)$.
4. Développez une fonction `is_prime_v1` qui indique si un entier donné est premier (test de primalité). Votre première version est simplement définie à l'aide de la fonction `teller` (à vous de voir comment).
5. A l'aide de vos fonctions, effectuez un test de la conjecture jusqu'à 1000. Pour chaque entier jusqu'à 1000, affichez sur une ligne, l'entier, son héritier.

Et la suite pour les étudiants impliqués

6. En vous appuyant sur vos fonctions précédentes et après un peu de recherche d'information, listez un nombre aussi important que possible d'entiers premiers, déficients, parfaits, abondants et super-abondants[1].

Pour chaque entier x > 1, en dérivant de la fonction `teller_iterated`, concevez et réalisez des fonctions qui retournent :

7. Le nombre $r(x)$ d'itérations de t avant convergence vers $H(x)$.
    a. **ET >** Exploitez les données obtenues pour voir comment évoluent les rapports entre $r(x)$, $H(x)$ et x.
    b. **ET >** r Est-elle monotone ? Énoncer une réponse, l'expliquer.
8. Les min et max de la suite $x,\ t(x),\ t^2(x),\ t^3(x),\ \cdots, H(x)$.
    a. **ET >** H est-elle croissante ? Expérimentez puis énoncez une réponse et démontrez là.
    b. **ET >** La suite $x,\ t(x),\ t^\circ t(x),\ t^3(x),\ \cdots, H(x)$ est-elle croissante ? Expérimentez puis énoncez une réponse et démontrez là.
9. On appellera recordman de r tout entier y tel que $\forall x < y, r(x) < r(y)$; établir une liste aussi longue que possible de recordmen de r.

[1] Voir la section réservée spéciale L1 affamés et L2 positionnés sur le sujet.

Ancêtres

Comme on l'a compris, quel que soit x, $H(x)$ est un entier premier et, si x est premier $H(x)=x$; de manière générale si y est un entier premier on appellera « ancêtre » de y tout entier x tel que $H(x)=y$.

Ainsi 11 possède trois ancêtres 4, 6, 11.

10. Rechercher les entiers premiers qui ont le plus grand nombre d'ancêtres; le recordman sera celui qui aura trouvé le nombre premier qui possède le plus grand nombre d'ancêtres.

Premiers solitaires

Les premiers solitaires sont des premiers qui ne sont héritier d'aucun autre entier qu'eux-mêmes :

11. Concevez et réalisez un « test de solitude ».
12. Produisez la plus grande liste possible d'entiers solitaires.
13. Identifiez les éventuels couples de solitaires consécutifs.

Les premiers super héritiers

14. Comptez le nombre d'ancêtres pour un premier donné (les entiers dont le premier est héritier).
15. Recherchez les super-héritiers, c'est-à-dire les premiers qui ont un maximum d'ancêtres, en valeur absolue, puis rapporté à eux-mêmes.

Diviseurs premiers

Etant donné la suite $x,\ t(x),\ t\circ t(x),\ t^3(x),\ \cdots, H(x)$ on désignera par $P(x)$ le maximum des diviseurs premiers de ces nombres; par exemple :

- si $x=8$, $t(x)=14$, $t^2(x)=23$; la liste des nombres de diviseurs premiers est [1, 2, 1];
- si $x=12$, $t(x)=27$, $t^2(x)=39$, $t^3(x)=55$, $t^4(x)=71$; la liste est [2, 1, 2, 2, 1].

Par conséquent, $P(8)=2$ et $P(12)=2$.

16. Concevez et réalisez la fonction $P(x)$.
17. Déterminez un record pour $P(x)$.

Fonctions relatives

18. Concevez et réalisez les fonctions $t(x)/x$ et $H(x)/x$.
19. **ET >** Pour chacune d'elles : répondre empiriquement et si possible théoriquement à ces questions : la fonction est-elle monotone ? bornée ? semble-t-elle avoir une limite (en l'infini) ?

Une première variante de t : w

On note w l'application qui à tout entier $x > 1$ associe la somme de ses diviseurs plus 1; ainsi $w(5) = 5 + 1 + 1 = 7$, $w(6) = 6 + 3 + 2 + 1 + 1 = 13$, etc.

On note w^n la composée n fois de w par w; ainsi $w^3(5) = w^2(13) = w(15) = 25$.
La question W est: existe-t-il des $x \in N \setminus \{0, 1\}, \exists p \in N, w^p(x) \in P$?

20. Expérimentez et énoncez une conclusion; tentez de la démontrer ?
21. Dans le cas où vous pensez qu'il y a des entiers x pour lesquels W est vraie déterminer la liste la plus riche possible d'entiers pour lesquels elle est vraie.

Compléments pour les plus motivés

22. Soient (x, y) deux entiers distincts, étudiez la suite $(w^n(x) - w^n(y))/(x - y)$.
    a. **ET >** Il est possible que vous constatiez des résultats très différents suivant les couples (x, y). Peut-être pourrez vous détailler ?

Une seconde variante de t : z

On note z l'application qui à tout entier $x > 1$ associe la somme de ses diviseurs strictement inférieurs à x ; ainsi $z(1) = 0$, $z(5) = 1$, $z(6) = 3 + 2 + 1 = 6$, etc.

On note z^n la composée n fois de z par z; ainsi $w^3(18) = w^2(21) = w(10) = 8$.
On remarquera ce que vaut $z(x)$ lorsque x est premier.
La conjecture Z est: $\forall x \in N \setminus \{0, 1\}, \exists p \in N, w^p(x) \in P$

23. Expérimentez et énoncez une conclusion
24. Dans le cas où vous pensez qu'il y a des entiers x pour lesquels Z est vraie déterminer la liste la plus longue possible d'entiers pour lesquels elle est vraie.
25. Soient (x, y) deux entiers distincts, étudiez la suite $(z^n(x) - z^n(y))/(x - y)$.
    a. **ET >** Il est possible que vous constatiez des résultats très différents suivant les couples (x, y). Peut-être pourrez vous détailler ?

Nombres de Mersenne

Renseignez vous sur les nombres de Mersenne et sur la raison pour laquelle ils jouent un rôle important en théorie des nombres, et en particulier pour la recherche des plus grand nombres premiers.

26. Si x est un nombre de Mersenne, $H(x)$ l'est-il également ?
27. Les nombres de Mersenne qui sont premiers ont-ils beaucoup d'ancêtres ?

Améliorez vos performances pour aller plus vite donc plus loin

29. Concevez et réalisez un crible pour lister les nombres premiers jusqu'à un rang `MAX` donné. Enregistrer ceux-ci dans un tableau.

30. Concevez et réalisez un algorithme qui décompose un entier x en son produit unique $\prod_{i=1}^{r} p_i^{k_i}$ de puissances de facteurs premiers. Retournez cette décomposition sous la forme d'un tableau dont les index 0, 1, 2, …, *k* correspondent respectivement aux nombres premiers 2, 3, p_2, …, p_k, où pk est le (k + 1)$^{\text{ème}}$ nombre premier, et dont les valeurs sont celles des exposants de la décomposition. Par exemple, la décomposition de $3^2 7^5 11$ est codée par [0, 2, 0, 5, 1, 0, …].
31. En utilisant la relation suivante $\sigma(x) = \prod_{i=1}^{r} \frac{p_i^{k_i+1}-1}{p_i-1}$, réalisez une seconde version `sigma_v2` de sigma.
32. Réalisez une version récursive de la fonction `iterated_teller`.
33. Poussez la vérification de la conjecture aussi loin que possible. Préciser jusqu'à quel rang on peut pousser la vérification avant d'observer des erreurs. Donnez dans votre rapport une interprétation de celles-ci.
34. Appuyez-vous sur tout ce que vous avez développé précédemment et amusez-vous à changer les fonctions t, w, z pour d'autres fonctions (par exemple $\sigma(x) - 1$ ou encore $\sigma(x) - x - 1$).
    a. L'itération de ces fonctions est-elle toujours convergente ?
    b. Le cas échéant, permettent-elle également de trouver des nombres premiers ?
    c. Quand elles ne convergent pas, que se passe-t-il ?
    d. Est-ce exploitable pour trouver des nombres spéciaux autres que des premiers ?

Pour les L2 et les L1 qui ont vraiment faim !

Se préparer pour battre des records

D'abord affûtez vos algorithmes

En vous appuyant sur une recherche dans la littérature, et en partant par exemple des pages http://fr.wikipedia.org/wiki/Somme_des_diviseurs et http://en.wikipedia.org/wiki/Primality_test, cherchez à améliorer les performances des fonctions `sigma` et `is_prime`, d'une part dans le cadre d'un appel unique, d'autre part dans le cadre d'appels successifs avec des entiers successifs passés en argument.
Présentez dans votre rapport les algorithmes réalisés, leurs performances théoriques, puis confirmez ces performances à l'aide de mesures pratiques (`#include <time.h>`) comparatives (benchmark) entre les différentes approches envisagées.

Puis dépassez les limites imposées par les types primitifs du C

Le plus grand entier qui peut être représenté en C avec le type `unsigned long` est 2^{32} - 1 (2^{64} - 1 avec l'`unsigned long long` en C99). Cela est ennuyeux car vous empêche d'explorer plus avant la validité de la conjecture.

En utilisant des tableaux d'entiers[2], concevez et réalisez des fonctions pour effectuer les opérations arithmétiques de bases sur de grands nombres. Et pourquoi pas, tentez de dépasser le précédent record de vérification de la conjecture, actuellement fixé à 2.10^6.

Evidemment, cela est un gros chantier, puisqu'il vous faut implémenter toutes les opérations arithmétiques de base :

1. addition et soustraction
2. multiplication et division (naïve, Karatsuba, Schönhage-Strassen (FFT))
3. modulo et test de division rapides
4. plus grand commun diviseur
5. etc, suivant les besoins de vos algorithmes de plus haut niveau.

Exploitez vos connaissances des arbres pour identifier des propriétés mathématiques exploitables pour des pistes de démonstration

Héritiers

Tout nombre est ancêtre au pire de lui même (solitaires) ou d'un unique héritier.

Par ailleurs, pour une raison évidente que nous tairons puisque c'est une question élémentaire posée plus haut, le nombre d'héritiers d'un premier est fini.

Concevez et réalisez un algorithme qui retourne, pour chaque nombre premier, l'arbre fini de ses ancêtres conçu de la manière suivante :

1. Si t(x) = t(y) = z, alors z est le parent de x et y.
2. Si une fratrie comporte 2 ou davantage d'éléments, organisez ces éléments en ordre croissant, en utilisant une liste chaînée (dont la référence de tête est stockée dans le noeud parent qui représente z).

Autres arbres numériques

En vous appuyant sur la question 34), tentez de découvrir et de produire d'autres arbres numériques, d'en dégager les propriétés.

Tentez ensuite l'interprétation théorique de ces résultats.

Lancez-vous dans la compétition et cherchez à battre des records

Enregistrement des records

A chaque fois que vous battrez un record de vérification de la conjecture de Teller ou d'autres défis listés ci-après, vous l'enregistrez dans le formulaire suivant : FORMULAIRE.
Les résultats seront visibles de tous.

---

[2] **NB > Les L2** positionnés sur le sujet devront impérativement utiliser des listes chaînées. Il n'est pas interdit à des L1 qui maîtrisent déjà ces techniques de les employer également.

Quels records viser ?

1. La vérification de la conjecture de Teller elle-même.
2. A COMPLETER

Ressources complémentaires

Eléments d'arithmétique

=> En complément du cours de Patrick Teller.
Cours d'arithmétique - Animath
Les nombres premiers

OEIS > The Online Encyclopedia of Integer Sequences

=> Pour vérifier vos résultats.
Une suite étrange de nombre sort de votre machine, prenez le réflexe d'en saisir les premiers termes dans le moteur de recherche OEIS, cela vous donnera des connexions avec des éléments théoriques. Et si votre suite est inconnue, c'est bon signe, c'est que vous foulez une terre vierge !

Pour les L2 ou L1 affamés : GMP

=> La librairie de référence pour calculer avec des nombres sans limite de taille.
https://en.wikipedia.org/wiki/GNU_Multiple_Precision_Arithmetic_Library
Vous pouvez vous en inspirer, mais, en particulier les L2, vous devez créer vos propres structures de données et algorithmes !

Pour vous donner encore plus soif

http://geekcafe.fr/la-theorie-sur-les-nombres-premiers-jumeaux-fait-un-grand-pas-en-avant/