Projet X : Formule de Legendre

La formule de Legendre donne une expression, pour tout nombre premier p et tout entier naturel n, de l'exposant de p dans la décomposition en facteurs premiers de $n!$ (ce qu'on appelle la valuation p-adique de $n!$. Le but de ce projet (Python) est de prouver cette formule.

1. La formule de Legendre

P.Chebychev a également établi (en 1850) une minoration $c_2 \frac{n}{\log(n)} \leqslant \pi(n)$. Une première étape dans cette direction est l'étude de la factorisation de $n!$, qui fait l'objet des questions suivantes.

Etant donné un nombre entier $n \geq 1$, et un nombre premier p, on note $\nu_p(n)$ la valuation p-adique de n définie comme étant le plus grand entier ν tel que p^ν divise n.

1. Quelle est la valuation 5-adique de $9625 = 5^3 \times 7 \times 11$?
2. Ecrire un algorithme en python `valuation` prenant en entrée un entier n et un nombre premier p, et qui renvoie $\nu_p(n)$. Vérifiez que vous retrouvez le résultat de la question précédente.
3. Montrer qu'étant donnés des entiers $a, b \geq 1$ et un nombre premier p, on a $\nu_p(ab) = \nu_p(a) + \nu_p(b)$.
4. Le but de cette question est de montrer que pour tout $n \geq 1$ et tout nombre premier p, on a (formule de Legendre)
$$\nu_p(n!) = \sum_{k \geq 1} \left\lfloor \frac{n}{p^k} \right\rfloor . \qquad (1)$$
Ici, on désigne par $\lfloor x \rfloor$ la *partie entière* d'un réel x, définie comme étant le plus grand entier m vérifiant $m \leq x < m + 1$.
   (a) Pour $k \geq 0$, on note Ω_k l'ensemble des entiers m compris entre 1 et n et tels que $\nu_p(m) = k$. Justifier que Ω_k est l'ensemble vide pour $k > \log(n)/\log(p)$.
   (b) Ecrire un programme `VerifFormule1` qui prend en argument un entier $n \geq 1$ et un nombre premier p et qui renvoie “True” si (1) est vérifiée pour n et p.
   (c) Ecrire un programme Python `Omegak` qui prend en argument n, p et k et renvoie Ω_k (sous forme de liste).
   (d) Montrer que les Ω_k forment une partition de $\{1, \ldots, n\}$: autrement dit, pour tout entier m compris entre 1 et n il existe k tel que $m \in \Omega_k$, et les Ω_k sont deux à deux disjoints.
   (e) Ecrire un programme `PartitionOmegak` qui prend en argument n et p et qui renvoie la liste des Ω_k pour k entier compris entre 0 et $\log(n)/\log(p)$.
   (f) Déduire des questions précédentes que
$$\nu_p(n!) = \sum_{k \geq 1} (\#\Omega_k) \times k.$$
   (g) Ecrire un programme Python qui permet de vérifier cette égalité pour n et p donnés.
   (h) Justifiez que le nombres de multiples de p^k inférieurs à n est $\lfloor n/p^k \rfloor$.
   (i) Ecrire un programme `Multiplepk` qui prend en argument p, k, et n et qui renvoie la liste des multiples de p^k inférieurs à n.
   (j) En déduire une expression pour $\#\Omega_k$, puis prouvez (1).

2. Pour aller plus loin

1. En utilisant la formule de Legendre, montrez que $\nu_p(n!) = N + \nu_p(N!)$, avec $N = \lfloor n/p \rfloor$.
2. En utilisant la question précédente, écrire un programme récursif `ValuationNfact` qui prend en argument n et qui renvoie $\nu_p(n!)$.
3. Ecrire un programme `Base` qui prend en argument n et p et qui renvoie l'écriture de n en base p, c'est-à-dire la liste des a_i (compris entre 0 et $p-1$) tels que
$$n = \sum_{i \geq 0} a_i p^i.$$
4. On note
$$s_p(n) = \sum_{i \geq 0} a_i,$$
où les a_i sont définis comme dans la question précédente. La quantité $s_p(n)$ est donc la somme des chiffres de n écrit en base p. Ecrire un programme `SommeChiffres` qui prend en argument n et p et qui renvoie $s_p(n)$.
5. Comparez $\nu_n(p)$ et $(n - s_p(n))/(p-1)$ pour les premières valeurs de n. Que constate-t-on ?
6. Soient n et k deux entiers > 0. Justifiez que
$$\left\lfloor \frac{n}{p^k} \right\rfloor = \sum_{i \geq k} a_i p^{i-k}.$$
7. (Difficile) En utilisant la formule de Legendre, en déduire que
$$\nu_p(n) = \sum_{i \geq 1} a_i \sum_{k=0}^{i-1} p^k,$$
puis conclure que
$$\nu_p(n) = \frac{n - s_p(n)}{p-1}.$$