Algorithmique
Matthieu Boyer
Planche A

Khôlle MPI
25/11/2024

Cribles

On dit qu'un entier naturel est *sans carré* s'il n'est pas divisible par le carré d'un entier supérieur à 2.

0. Donner le pseudo-code d'un algorithme permettant de calculer le nombre d'entiers naturels sans carrés inférieurs ou égaux à n.

On note $\pi(n)$ le nombre de nombres premiers inférieurs ou égaux à n. On admet le théorème suivant, dit des *nombres premiers* :

Théorème 0.1 — des Nombres Premiers

$$\pi(n) \sim_{n\to\infty} \frac{n}{\ln n}$$

En particulier, la valeur du k-ème nombre premier p_k est équivalente à $k \ln k$.

1. Donner le pseudo-code d'un algorithme permettant de calculer $\pi(n)$.
2. Donner une structure de données permettant de représenter un sous-ensemble E de $[\![1, n]\!]$ et supportant les opérations suivantes :
   - Initialisation à $[\![1, n]\!]$ en temps $\mathcal{O}(n)$
   - Suppression d'un élément en temps $\mathcal{O}(1)$
   - Énumération dans l'ordre croissant de tous les éléments de l'ensemble $E \subseteq [\![1, n]\!]$ en temps $\mathcal{O}(|E|)$.

   La structure de données occuper un espace $\mathcal{O}(n)$.
3. En déduire le pseudo-code d'un algorithme qui permet de résoudre la question 1 en temps $\mathcal{O}(n)$.
4. Donner le pseudo-code d'un algorithme permettant de calculer tous les $p \wedge q$ pour $p, q \in [\![1, n]\!]$.

On admet qu'il est possible d'implémenter une structure de données de dictionnaire dont les clefs sont des entiers (non-nécessairement contigus) de telle façon que tous les accès soient en temps $\mathcal{O}(1)$. La structure utilise un espace mémoire $\mathcal{O}(n)$ où n est le nombre de clefs différentes.

1. On note $\Phi(n)$ le nombre de paires d'enters (p, q) avec $p \wedge q = 1$. Montrer que :

$$\Phi(n) = n^2 - \sum_{d=2}^{n} \Phi\left(\left\lfloor \frac{n}{d} \right\rfloor\right) \tag{1}$$

   En déduire le pseudo-code d'un algorithme permettant de calculer $\Phi(n)$ en temps sous-linéaire par rapport à n.

Solution: Cribles

Voir `https://diplome.di.ens.fr/informatique-ens/annales/2019_InfoU-exercices.pdf` page 36.

Algorithmique
Matthieu Boyer
Planche B

Khôlle MPI
25/11/2024

Kolmogorov

Un mot binaire est un mot fini dans l'alphabet $\Sigma = \{0, 1\}$. On note Σ^* l'ensemble des mots binaires. La longueur d'un mot $w \in \Sigma^*$ est notée $|w|$. Fixons un langage de programmation L raisonnable (parmi OCaml, OCaml et OCaml). Étant donné un mot binaire w, une description de w dans le langage de programmation L est un mot binaire de la forme $0^{|p|}1pu$ où p est la représentation binaire d'un programme P écrit dans le langage de programmation L qui sur l'entrée u affiche le mot w. On définit la complexité de Kolmogorox de w dans L par :

$$K_L(w) = \min\{|d| \mid d \text{ est une description de } w \text{ dans } L\}$$

0. Montrer que pour tout mot w, $K_L(w) \leq |w| + \mathcal{O}(1)$.
1. Montrer que pour tout $n \in \mathbb{N}$ il existe un mot w de longueur $n+1$ tel que $K_L(w) > n$.
2. Montrer que pour tous mots w et u, $K_L(wu) \leq 2K_L(w) + K_L(u) + \mathcal{O}(1)$. Peut-on améliorer cette borne ?
3. Soit K_U la complexité de Kolmogorov définie pour un autre langage de programmation U. Montrer qu'il existe $c \in \mathbb{N}$ telle que pour tout mot fini w, $K_L(w) \leq K_U(w) + c$.

Une fonction partielle $f : \Sigma^* \to \Sigma^*$ est calculable s'il existe P qui, lorsqu'il prend en entrée un mot binaire w affiche $f(w)$ s'il est défini et n'affiche rien sinon.

4. Montrer que pour toute fonction partielle calculable $f : \Sigma^* \to \Sigma^*$ et tout mot fini w, on a : $K_L(f(w)) \leq K_L(w) + \mathcal{O}(1)$.

On note $\bar{n} \in \Sigma^*$ la représentation en base 2 d'un entier $n \in \mathbb{N}$. La complexité de Kolmogorov d'un entier $n \in \mathbb{N}$ est celle de $\bar{n}$. De même, on dit que $f : \mathbb{N} \to \Sigma^*$ est calculable s'il existe un programme P qui, lorsqu'il prend en entrée $w \in \Sigma^*$, affiche $f(n)$ s'il existe n tel que $\bar{n} = w$, et n'affiche rien sinon.

5. Montrer que pour toute fonction calculable f, $K_L(f(n)) \leq \log_2(n) + \mathcal{O}(1)$.
6. Donner un algorithme en pseudo-code calculant une fonction surjective de $\mathbb{N}$ dans Σ^*.
7. Montrer que K_L n'est pas calculable.
8. Montrer qu'il existe une infinité de nombres premiers (en utilisant les questions précédentes).

Solution: Kolmogorov

Voir `https://diplome.di.ens.fr/informatique-ens/annales/2019_InfoU-exercices.pdf` page 60.

Algorithmique
Matthieu Boyer
Planche C

Khôlle MPI
25/11/2024

Zeckendorf

On considère la suite de Fibonacci définie par $F_0 = 0$, $F_1 = 1$ et $\forall n \in \mathbb{N}, F_{n+2} = F_{n+1} + F_n$.

0. Donnez un algorithme en $\mathcal{O}(\ln n)$ renvoyant F_n. On se souviendra que si on note $\varphi = \frac{1+\sqrt{5}}{2}$ et $\bar{\varphi} = \frac{1-\sqrt{5}}{2}$, on a $\varphi^2 = \varphi + 1$ et $\bar{\varphi}^2 = \bar{\varphi}^2$.
1. Montrer que pour tout entier n, on peut trouver k entiers c_i tels que :

$$n = \sum_{i=0}^{k} F_{c_i}, \text{ avec } 1 < c_0 \text{ et } \forall i \geq 1, c_i > c_{i-1} + 1$$

2. Soit $Z \subseteq \mathbb{N} \setminus \{0, 1\}$ un ensemble fini qui ne contient pas d'entiers consécutifs. Soit $m = \max Z$. Montrer que :

$$F_{m+1} > \sum_{k \in Z} F_k$$

3. Prouver l'unicité de la décomposition de tout entier comme somme de termes de la suite de Fibonacci.
4. Montrer que le langage $\mathcal{L}$ sur l'alphabet $\{0, 1\}$ contenant le mot vide et le mots ne possédant pas deux 1 consécutifs mais terminant par 1 est régulier.

Pour $u \in \mathcal{L}$ de longueur $|u|$, et $k \leq |u|$ on pose $u_k = 0$ si la k-ème lettre de u est 0, $u_k = 1$ sinon. On dit que u est la représentation de Zeckendorf de n si :

$$n = \sum_{k=1}^{|u|} u_k F_{k+1}$$

Cette représentation est unique.

5. Quelle est la longueur de la représentation d'un entier ?

Solution: Zeckendorf

0. On utilise l'exponentiation rapide après avoir vérifié les deux équations suivantes :

$$\bar{\varphi} = -\frac{1}{\varphi} \text{ et } F_n = \frac{1}{\sqrt{5}}\left(\varphi^n - \bar{\varphi}^n\right)$$

1. Par récurrence forte, on considère la propriété vraie jusqu'au rang $n \geq 0$. On a $n+1 \geq F_2$, on peut donc considérer le plus grand $m \geq 2$ tel que $n+1 \geq F_m$. Par hypothèse, on a une décomposition pour $M = n+1-F_m \leq n$. Alors, pour tout F_i de cette représentation, $F_m + F_i \leq F_m + M = N < F_{m+1} = F_m + F_{m-1}$. Donc $F_i < F_{m-1}$.
2. On procède par récurrence forte sur m, le maximum de notre ensemble Z. La propriété à démiontrer est vraie pour $m = 2$ et $m = 3$. On suppose la propriété vraie pour $m \geq 2$. Soit Z un ensemble fini de $\mathbb{N} \setminus \{0,1\}$ de maximum $m+1$ sans nombres consécutifs. Le deuxième plus grand élément de Z est au plus $m-1$. De ce fait on a par hypothèse de récurrence et par croissance de F, $F_m > \sum_{k \in Z \setminus \{m+1\}} F_k$. D'où :

$$F_{m+2} = F_{m+1} + F_m > \sum_{k \in Z} F_k$$

3. Les langages rationnels sont stables par complémentaires. Le langage des mots contenant 11 est rationnel, c'est celui reconnu par `(0+1)*11(0+1)*` . Ainsi le langage des mots ne contenant pas 11 est rationnel. De plus les langages rationnels sont stables par intersection, donc par intersection avec le langage de `(0+1)*1` le langage des mots ne contenant pas 11 et se terminant par 1 est rationnel. Les langages rationnels sont stables par union, donc le langage contenant le mot vide et les mots ne contenant pas 11 mais se terminant par 1 est rationnel.
4. La longueur de la représentation de 0 est 0 car il s'agit du mot vide. Pour un entier strictement positif n, cette longueur correspond à $m-1$ où m est le plus grand entier tel que $F_m < n$. Comme $\bar{\varphi}^n$ est négligeable devant φ^n, la longueur de la représentation de n est $\lfloor \log_\varphi(n) \rfloor$.