Complexité Algorithmique
Matthieu Boyer
Planche A

Khôlle MP2I
13/01

Un peu d'arithmétique

On rappelle l'algorithme d'Euclide :

if $b \mid a$ **then**
  $gcd \leftarrow b$
else
  $gcd \leftarrow gcd\,(b, a \mod b)$

1. Donner la complexité de l'algorithme de division euclidienne à la main et montrer sa correction.
2. Donner sa complexité.
3. Montrer que pour tout $a, b, a \wedge b = a \wedge (a - b)$ et en déduire la correction de l'algorithme.

Pour la multiplication, si x, y s'écrivent sur $n = 2^k$ chiffres, on pose $x = 10^{n/2}a + b$, $y = 10^{n/2}c + d$. Alors :

$$xy = 10^n ac + 10^{n/2}\,(ad + bc) + bd$$

Remarquons par ailleurs que

$$(a + b)\,(c + d) = (ad + bc) + (ac + bd)$$

<++>

4. En remarquant que les additions et les multiplications par des puissances de 10 peuvent être faites en $\mathcal{O}(n)$ (on justifiera succintement cette affirmation), montrer qu'on peut calculer le produit xy en $\mathcal{O}\left(n^{\log_2(3)}\right)$.

Solution: Un peu d'arithmétique

Complexité Algorithmique
Matthieu Boyer
Planche B

Khôlle MP2I
13/01

.

Étant donné une séquence de $2n$ nombres $A = x_1, y_1, \ldots, x_n, y_n$ les coordonnées de n points dans le plan, considérez les segments $p_i - p_j$ pour $i \neq j$.

1. Donnez deux algorithmes qui renvoie le nombre de segments verticaux et de segments horizontaux dans A.
2. Donnez un algorithme qui renvoie vrai si et seulement si il existe un segment vertical qui intersecte un segment horizontal.
3. Si on ne considèrait pas que les segments verticaux et horizontaux, pouvez-vous améliorer votre algorithme ?
4. Donnez un algorithme qui renvoie l'aire d'un carré de taille minimal qui contient tous les points.

Solution: .

1. On donne pour le vertical :

```
ef count_vertical(A):
        n = len(A)//2
        c = 0
        for i in range(n):
                for j in range(i + 1, n):
                        if A[2*i] == A[2*j]:
                                c = c + 1         return c
```

2. On énumère tous les segments verticaux, et tous les segments horizontaux et on regarde. Dans le pire des cas on est en $\Theta\left(n^4\right)$.
3. C'est le max des deux.

Complexité Algorithmique
Matthieu Boyer
Planche C

Khôlle MP2I
13/01

Littéralement 1984

Big Brother traque un ensemble de m téléphones cellulaires en enregistrant chaque antenne à laquelle chaque téléphone se connecte. En particulier, pour chaque utilisateur u_i, Big Brother stocke une suite ordonnée par le temps $S_i = (t_1, a_1) \dots$. Ceci signifie que u_i s'est connecté à a_1 à t_1 puis à a_2 à $t_2 > t_1$ et ainsi de suite. Écrivez un algorithme $gofk\,(S_1, S_2, \dots, S_m, k)$ qui renvoie un temps t^* tel qu'un groupe de k utilisateur est connecté à une même antenne. On cherchera à écrire $gofk$ en temps $\mathcal{O}\,(n \log m)$ où $n = \sum |S_i|$.

Solution: Littéralement 1984

On stocke dans un tas min dont la clef est le temps les $S_i[1], i, 1$. Tant que le tas n'est pas vide, on en prend le minimum (la plus vieille donnée, ce qui fait qu'on avance dans le temps). Si la clef de profondeur est $j > 1$, on enlève 1 à l'antenne dont on vient d'enlever un utilisateur. On ajoute l'utilisateur à son antenne. Si on a suffisamment de monde on s'arrête, sinon on ajoute $S_i[j+1], i, j+1$ au tas. Si le tas est vide on revoie `rien`.