

Preuve et Analyse des Algorithmes

09:00–11:00

aucun document autorisé

3 juillet 2015

Q 1. Pour un entier n , on note

$$H_n = \sum_{k < n} \frac{1}{k}, \quad S_n = \sum_{\substack{p < n \\ p \text{ premier}}} \frac{1}{p}.$$

1. Comparer S_n et H_n .
2. Estimer H_n en fonction d'une primitive de $\frac{1}{x}$.
3. En déduire que $S_n = O(\log n)$.
4. Peut-on affirmer que $S_n = \Theta(\log n)$?

Q 2. Soient m et n deux entiers positifs. On note d leur PGCD. Démontrer que $2^d - 1$ divise le PGCD de $2^m - 1$ et $2^n - 1$.

Q 3. Dessiner un graphe orienté d'ordre 9 ayant trois composantes fortement connexes, mais qui, sans tenir compte des orientations, ne possède qu'une seule composante connexe.

Q 4. On note Δ le plus grand diviseur commun à $a = 4301$ et $b = 1547$

1. Tracer les étapes de l'algorithme d'Euclide pour obtenir Δ .
2. Utiliser un algorithme d'Euclide étendu efficace vu en cours pour déterminer deux entiers relatifs u et v tel $au + bv = \Delta$.

```
1 typedef unsigned int uint;
2
3 int get( uint n, uint* t )
4 {
5     uint v;
6     v = t [ n / sizeof (uint) ];
7     v >>= n % sizeof (uint) ;
8     return v & 1;
9 }
```

boole.c

2. Préciser le temps de calcul en fonction de n .
3. Implanter l'algorithme en langage C.
4. Préciser la quantité de mémoire utilisée.

Q 6. Observer la source `boole.c` La fonction `get(n,t)` permet de consulter le **bit** d'indice n dans un tableau t d'entiers non signés. Ainsi, suivant l'architecture cible, $t[0]$ contient les 16, 32, ou 64 premiers bits.

Q 5. Le crible d'Eratostène permet de dresser la liste des nombres premiers inférieurs à un entier donné n .

1. Ecrire l'algorithme du crible d'Eratostène.

1. Ecrire une fonction `bit(n, t)` qui change l'état du bit d'indice n .
2. Comment utiliser ces fonctions dans l'implantation du crible ?
3. Quel est l'intérêt de cette approche ?