

TL;DR. Nous avons choisi l'algorithme de divisions successives, sur lequel nous lançons plusieurs threads qui testent la division avec des valeurs de départ décalées.

Déroulement

Nous commençons par lire tous les entiers à factoriser et nous les plaçons dans un heap (dans l'ordre inversé, avec le plus petit au-dessus). Ensuite, tant que le heap n'est pas vide, nous enlevons le plus petit entier du heap et nous lançons dessus les n threads à notre disposition. Dès qu'ils trouvent un facteur, ils l'ajoutent au heap et divisent la valeur à factoriser.

Une fois que les n threads ont tous atteint la racine, cela signifie que le nombre restant est premier. Nous essayons de diviser chaque nombre dans le heap par ce facteur premier. Si aucun n'est divisible, ce nombre est la solution. Sinon nous continuons.

Voici le pseudocode pour le thread principal :

```
1 read all input integers
2 push them to heap
3
4 while heap is not empty:
5
6     get toFact from heap
7
8     for i in 1..n:
9         init thread i with (start=1+i, step=n)
10    end
11
12    for i in 1..n:
13        join thread i
14    end
15
16    divide heap by toFact
17
18    if no division possible:
19        toFact is the solution
20        print it and exit
21    end
22
23 end
24
25 (if the loop ends input is invalid)
```

Remarquons que si la boucle `while` se finit, l'input est incorrect, car cela signifie qu'aucun facteur premier ne divise exactement un nombre exactement une fois.