

Introduction à l'informatique théorique – 2005 devoir 11

Alban Guyon

May 24, 2022

1 Questions

1 Pour représenter une liste de nombres entiers en un seul nombre entier on utilise la notation suivante:

$$[a_1, a_2, a_3, \dots, a_n, 0, 0, 0, \dots] = 2^{a_1} * 3^{a_2} * 5^{a_3} * \dots * p^{a_n} \quad (1)$$

où les bases des exposants sont les nombres premiers.

Du fait que chaque entier positif a une factorisation unique en nombres premiers, chaque liste sera associée à un nombre unique supérieur ou égal à 1.

Une autre chose à noter est que toutes les listes sont infinies: une liste avec n éléments à réelement n éléments et une infinité de 0 qui suivent. En effet, les 0 à la fin d'une liste sont ignorés. Par exemple, la liste $[2, 3, 4, 3, 2, 0, 0]$ est équivalente à $[2, 3, 4, 3, 2]$

Dans le pseudo-code, on utilise la notation P_i pour représenter le i -ème nombre premier. Voici quelques fonctions sur les listes dans cette nouvelle représentation:

Algorithm 1 Vide

```
procedure VIDE()
   $r_0 \leftarrow 1$ 
```

Algorithm 2 EstVide?

```
procedure ESTVIDE?( $r_1$ )
  if  $r_1 = 1$  then
     $r_0 \leftarrow \text{TRUE}$ 
  else
     $r_0 \leftarrow \text{FALSE}$ 
```

Algorithm 3 Dans?

```
procedure DANS?( $r_1, r_2$ )
   $r_1 \leftarrow \text{FALSE}$ 
  for  $i \leftarrow 0$  to  $r_1$  do ▷ on utilise  $r_1$  pour la limite max car taille de la liste est forcément inférieur
    if  $r_1 \bmod P_i^{r_2} = 0$  and not  $r_0$  then
       $N \leftarrow n / P_i^{r_2}$  ▷ on enleve  $r_1$  fois le même facteur premier
      if  $N \bmod (P_i^{r_2}) \neq 0$  then ▷ on s'assure qu'il n'y a pas d'autre diviseurs
         $r_1 \leftarrow \text{TRUE}$ 
```

Algorithm 4 Card

```
procedure CARD( $r_1$ )
   $r_0 \leftarrow 0$ 
  for  $i \leftarrow 0$  to  $r_1$  do
    if  $n \bmod (P_i^{r_1}) = 0$  then
       $r_0 \leftarrow i + 1$  ▷ l'indice du dernier element non nul + 1 est la taille
```

Algorithm 5 Ajouter

IND

```
procedure AJOUTER( $r_2, r_1$ )  
   $SIZE \leftarrow Card(r_1)$   
   $r_0 \leftarrow r_1 * P_{SIZE}^{r_2}$ 
```

Pour la fonction retirer, je vais d'abord définir les macros auxiliaires $IndexDe(r_2, r_1)$ et $index(r_2, r_1)$:
 $IndexDe(r_2, r_1)$ retourne l'indice de l'élément r_2 dans la liste r_1 .
 $index(r_2, r_1)$ retourne l'élément à l'indice r_2 dans la liste r_1

Algorithm 6 IndexDe

```
procedure INDEXDE( $r_2, r_1$ )  
   $r_0 \leftarrow -1$  ▷ on initialise l'indice à -1 qui signifie que l'élément n'est pas dans la liste  
   $FOUND \leftarrow FALSE$   
  for  $i \leftarrow 0$  to  $r_1$  do  
    if not FOUND then  
       $COPY \leftarrow r_1$   
       $FLAG \leftarrow TRUE$  ▷ Pouvait-on enlever  $r_2$  instaces de  $P_i$ ?  
      for  $j \leftarrow 0$  to  $r_2$  do  
        if  $COPY \bmod P_j = 0$  then  
           $COPY \leftarrow COPY / P_j$   
        else  
           $FLAG \leftarrow FALSE$   
      if  $FLAG$  and  $COPY \bmod P_i \neq 0$  then ▷ Si  $P_i$  n'est plus un facteur de  $COPY$   
         $FOUND \leftarrow TRUE$   
         $r_0 \leftarrow i$ 
```

Algorithm 7 Index

```
procedure INDEX( $r_2, r_1$ )  
   $r_0 \leftarrow 0$   
   $COPY \leftarrow r_1$   
  for  $i \leftarrow 0$  to  $r_1$  do ▷ On compte le nombre de fois que l'on peut enlever  $P_{r_2}$  de  $r_1$   
    if  $COPY \bmod P_{r_2} = 0$  then  
       $COPY \leftarrow COPY / P_{r_2}$   
     $r_0 \leftarrow r_0 + 1$ 
```

On peut maintenant utiliser ces macros dans la macro Retirer:

Algorithm 8 Retirer

```
procedure RETIRER( $r_2, r_1$ )  
   $IND \leftarrow IndexDe(r_2, r_1)$   
   $COPY \leftarrow r_1 / P_{IND}^{r_2}$  ▷ On met l'élément que l'on veut enlever à 0  
  for  $i \leftarrow 0$  to  $r_1$  do ▷ On décale toutes les valeurs suivantes un indice à gauche  
     $VAL \leftarrow Index(i + 1, COPY)$   
     $COPY \leftarrow COPY / P_{i+1}^{VAL}$   
     $COPY \leftarrow COPY * P_i^{VAL}$   
   $r_0 \leftarrow COPY$ 
```

Algorithm 9 Inter

```
procedure INTER( $r_1, r_2$ )  
   $r_0 \leftarrow 1$   
  for  $i \leftarrow 0$  to  $r_1$  do  
     $VAL \leftarrow Index(i, r_1)$   
    if  $Dans?(r_1, VAL)$  and not  $Dans?(L, VAL)$  then ▷ Si l'élément est dans les deux listes et on l'a pas déjà écrit  
       $r_0 \leftarrow Ajouter(VAL, r_0)$ 
```
