EPFL - Cours “Information, Calcul et Communication”

Exemples d’exercices - Semaine 3

1 Au temps des Egyptiens, deuxième partie

Proposez une version récursive de l’algorithme de las semaine préccédente :

devinette
entrée : a, b *deux entiers naturels non nuls* sortie : $a \times b$

$x \longleftarrow a$
$y \longleftarrow b$
$z \longleftarrow 0$
Tant que $y \geq 1$
  Si y *est pair*
  $x \longleftarrow 2\,x$
  $y \longleftarrow y/2$
  Sinon
  $z \longleftarrow z + x$
  $y \longleftarrow y - 1$
sortir : z

2 Taille de liste

Soit L une liste d’entiers (pas forcément ordonnée et avec de possibles répétitions), comme par exemple $\{19, 31, 15, 21\}$.

On cherche ici à écrire un algorithme **taille**(L) qui retourne le nombre d’éléments d’une liste L donnée en entrée.

On suppose que l’on dispose d’un autre algorithme **a_element**(L, i) qui nous dit (vrai ou faux) s’il existe un i$^{\text{e}}$ élément dans la liste : il répond donc “ vrai ” si i est inférieur ou égal à la taille de la liste et “ faux ” sinon. [1]

1. Comment utiliser l’algorithme **a_element**(L, i) pour calculer la taille de L en un temps linéaire (par rapport à cette taille) ?

   Ecrivez un algorithme pour le faire.

2. En vous inspirant de la recherche dichotomique (cf. cours), pourrait-on avoir une complexité moindre que linéaire ?

   Si oui, quelle serait cette complexité et expliquez intuitivement comment procéder.

   Pour les plus motivés : essayez d’écrire un tel algorithme.

1. Notez qu’il n’est pas évident de toujours avoir un tel algorithme (sans avoir **taille**, bien sûr !). En realité, cela dépend de la représentation effective de L. Mais nous faisons ici l’hypothèse qu’un tel algorithme existe pour L.