Examen 1 2018 Q5

Quelle est la complexité de l'algorithme suivant :

algo5

entrée : *Entier positif n*
sortie : *valeur*

Si $n \leq 2$
  Sortir : 3
Sinon
  Sortir : $4 \log(\textbf{algo5}(n-2)) + 1$

Examen 1 2018 Q15

Cette question est la suite de celle (Q13) posée la semaine passée que je reproduis donc ici (suite sur le transparent suivant).

On s'intéresse ici à raccourcir les répétitions de trois ou plus valeurs identiques successives ; par exemple à produire la liste $(6,6,4,4,12,4,6)$ à partir de la liste $(6,6,4,4,4,12,4,6)$ en supprimant le 4 en cinquième position car il est présent trois fois consécutives.

À noter que :

- les seules valeurs supprimées sont celles qui sont répétées successivement trois fois ou plus (l'une à la suite de l'autre) ; on ne garde alors que deux de ces valeurs (cf la valeur 4 ci-dessus) ;
- toute valeur présente une ou deux fois successivement est préservée, et l'on conserve l'ordre de la liste ;
- en sortie on ne peut donc pas avoir plus de deux valeurs identiques consécutives.

Examen 1 2018 Q15 (2)

On s'intéresse à une solution récursive séparant le premier élément et le reste de la liste.

Pour cela, il serait plus pratique que l'algorithme retourne également le nombre de répétitions successives de la première valeur dans la liste résultat ;

par exemple à partir de la liste $(6,6,6,6,4,4,4,12,4,6)$, l'algorithme récursif doit produire la sortie $(6,6,4,4,12,4,6)$ et 2, car il y a 2 fois la valeur 6 au début de la liste résultat.

A noter que cette information supplémentaire sera donc nécessairement inférieure ou égale à 2.

Ecrivez un algorithme récursif résolvant le problème de cette façon.

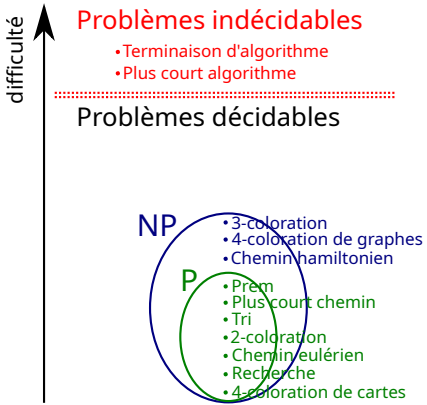
Examen 1 2018 Q15 (solution)

Examen 1 2018 Q16

Déterminez la complexité de votre algorithme (question précédente).
Justifiez votre réponse.

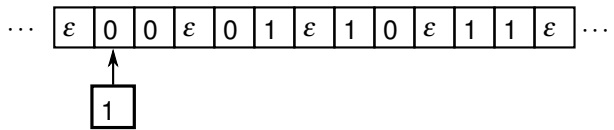
Leçon I.3 (calculabitilé) – Points clés

- qu'est-ce que *une* machine de Turing et **la** machine de Turing universelle
- problème = une question (portant sur ses entrées) et (une infinités d') instances
- il y a beaucoup de problèmes non calculables
  exemple : problème de l'arrêt
- P et NP
- on ne sait pas si NP est inclu dans P
- la 2- et la 4-coloration de graphes planaires sont dans P
  la 3-coloration de graphes est dans NP
- et d'autres...



Leçon I.3 (calculabitilé) – Machines de Turing

	0	1	ε
1	$(3, \varepsilon, +)$	$(4, \varepsilon, +)$	$(2, \varepsilon, +)$
2	$(3, \varepsilon, +)$	$(4, \varepsilon, +)$	$(6, \varepsilon, -)$
3	$(1, \varepsilon, +)$	$(1, \varepsilon, +)$	$(1, \varepsilon, +)$
4	$(5, \varepsilon, -)$	$(3, \varepsilon, +)$	$(1, \varepsilon, +)$
5	$(5, \varepsilon, +)$	$(5, \varepsilon, +)$	$(7, 1, -)$
6	$(6, \varepsilon, +)$	$(6, \varepsilon, +)$	$(7, 0, -)$



Leçon I.3 (calculabitilé) – P et NP

Si un algorithme résoud un problème X (dont la taille de l'entrée est n) en $\Theta(n^2 \log n)$, que peut on dire ?

Si un algorithme permet de vérifier en $\Theta(n^2 \log n)$ qu'une donnée de taille n est bien une instance positive (« solution ») à un problème Y, moyennant une donnée supplémentaire (certificat) de taille en $\Theta(\log n)$, que peut on dire ?

Un tel problème est-il dans P ?
Peut-il être dans P ?

Leçon I.3 (calculabitilé) – Complexité des problème

Un algorithme possible pour résoudre un problème Z est le suivant :

1. transformer les données de Z en un graphe
2. trouver un chemin hamiltonien dans ce graphe
3. retransformer ce chemin dans les données de Z, puis conclure

avec à chaque fois des transformations qui sont en $\Theta(n)$
(n étant la taille de l'entrée du problème Z).

Que peut-on dire de Z ?

1. rien
2. qu'il est dans P
3. qu'il est dans NP
4. qu'il n'est pas dans P
5. (qu'il est NP-difficile)