Exponentiation

Dernière mise à jour de ce document : 27 juin 2019.

Document réalisé avec LaTeX.

Plan

Extraits du programme NSI :

Extraits du programme NSI :

En Première : « [...] Quelques algorithmes classiques sont étudiés, *(ainsi que)* l'étude de leurs coûts respectifs [...] »

Extraits du programme NSI :

En Première : « [...] Quelques algorithmes classiques sont étudiés, *(ainsi que)* l'étude de leurs coûts respectifs [...] »

En Terminale :

Contenues	Capacités	Commentaires
Récursivité	Écrire un programme récursif. Analyser le fonctionnement d'un programme récursif.	Des exemples relevant de domaine variés sont à privilégier.

Pré-requis :

1. *Les notions de boucles, de fonctions.*
2. *L'écriture binaire d'un nombre en écriture décimale.*
3. *Fonctions récursives en Terminale ou bien on fera l'introduction d'une fonction récursive à cette occasion.*

Plan

Lien vers l'activité

Bilan de l'activité avec la notion de complexité (introduite à cette occasion-là ou réinvestie si faite précédemment), idem pour la notion de fonction récursives.

Plan

Lien vers un exercice sur des matrices avec Fibonacci et un exercice sur la récursivité de l'exponentiation

Quelques ouvertures à l'exponentiation rapide :

1. La méthode binaire est-elle toujours le meilleur algorithme ?

Quelques ouvertures à l'exponentiation rapide :

1. La méthode binaire est-elle toujours le meilleur algorithme ?

   Contre-exemple :

   y1 = a * a (a^2) 1 multiplication

Quelques ouvertures à l'exponentiation rapide :

1. La méthode binaire est-elle toujours le meilleur algorithme ?

   Contre-exemple :

   y1 = a * a (a^2) 1 multiplication

   y2 = y1 * y1 (a^4) 1 multiplication

Quelques ouvertures à l'exponentiation rapide :

1. La méthode binaire est-elle toujours le meilleur algorithme ?

   Contre-exemple :

y1 = a * a	(a^2)	1 multiplication
y2 = y1 * y1	(a^4)	1 multiplication
y3 = y2 * y2	(a^8)	1 multiplication

Quelques ouvertures à l'exponentiation rapide :

1. La méthode binaire est-elle toujours le meilleur algorithme ?

   Contre-exemple :

y1 = a * a	(a^2)	1 multiplication
y2 = y1 * y1	(a^4)	1 multiplication
y3 = y2 * y2	(a^8)	1 multiplication
a^{15} = a * y1 * y2 * y3		3 multiplications

Quelques ouvertures à l'exponentiation rapide :

1. La méthode binaire est-elle toujours le meilleur algorithme ?

   Contre-exemple :

y1 = a * a	(a^2)	1 multiplication
y2 = y1 * y1	(a^4)	1 multiplication
y3 = y2 * y2	(a^8)	1 multiplication
a^{15} = a * y1 * y2 * y3		3 multiplications

   soit au total 6 multiplications

mais 15 = 3*5 donc

mais 15 = 3*5 donc

Y1 = a * a (a^2) 1 multiplication

mais 15 = 3*5 donc

Y1 = a * a	(a^2)	1 multiplication
Y2 = Y1 * Y1	(a^4)	1 multiplication

mais 15 = 3*5 donc

Y1 = a * a	(a^2)	1 multiplication
Y2 = Y1 * Y1	(a^4)	1 multiplication
Y3 = Y2 * a	(a^5)	1 multiplication

mais 15 = 3*5 donc

Y1 = a * a	(a^2)	1 multiplication
Y2 = Y1 * Y1	(a^4)	1 multiplication
Y3 = Y2 * a	(a^5)	1 multiplication
a^{15} = Y3 * Y3 * Y3		2 multiplications

mais 15 = 3*5 donc

Y1 = a * a	(a^2)	1 multiplication
Y2 = Y1 * Y1	(a^4)	1 multiplication
Y3 = Y2 * a	(a^5)	1 multiplication
a^{15} = Y3 * Y3 * Y3		2 multiplications

soit au total 5 multiplications.

mais 15 = 3*5 donc

Y1 = a * a	(a^2)	1 multiplication
Y2 = Y1 * Y1	(a^4)	1 multiplication
Y3 = Y2 * a	(a^5)	1 multiplication
a^{15} = Y3 * Y3 * Y3		2 multiplications

soit au total 5 multiplications.

On utilise alors la méthode des facteurs, voir :

lien vers LIX de polytechnique

(notamment avec une exponentiation avec l'arbre de Knuth)

(Pour la méthode des facteurs : on trouvera un fonction récursive la codant dans le fichier `puissance_en_python.py`)

2. (a) Exponentiation modulaire :

   voir vidéo : vidéo sur Youtube

   algorithme : exponentiation rapide modulo (n)

   voir instruction python : `pow(a,n,m)`

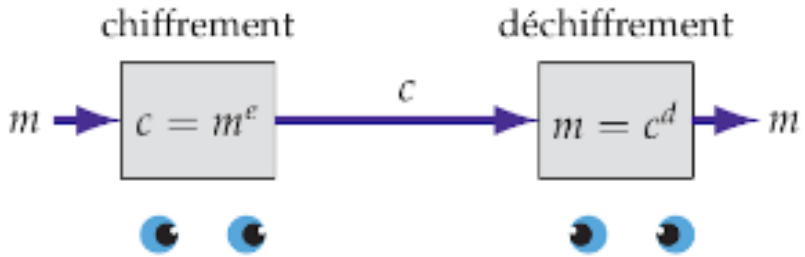
2. (a) Exponentiation modulaire :

   voir vidéo : vidéo sur Youtube

   algorithme : exponentiation rapide modulo (n)

   voir instruction python : `pow(a,n,m)`

   (b) Applications à la cryptographie : On utilise l'exponentiation modulaire pour chiffrer puis déchiffrer un message :



   Le modulo n et l'exposant de chiffrement e sont publics alors que d est l'exposant de déchiffrement privé.

   Les valeurs n , e , d sont reliées mathématiquement (vu en Mathématiques expertes en Terminale).

Extraits du programme NSI :

En Terminale :

Contenues	Capacités	Commentaires
Sécurisation des communications.	Décrire les principes de chiffrement symétrique (clef partagée) et asymétrique (avec clef privée/clef publique). [...]	[...]