Limites de la programmation dynamique

ÉCOLE POLYTECHNIQUE ParisTech

Le problème des Chaînes additives

Conception et mise en œuvre d'algorithmes

Programmation Dynamique - leçon 5-5

Benjamin Werner

Ecole Polytechnique

Le problème

ÉCOLE POLYTECHNIQUE ParisTech

On veut calculer $a^n = \underbrace{a \times a \times a \times \ldots \times a}_{n \text{ fois}}$

en faisant le minimum de multiplications

a^2 a^3 a^4 a^5 … a^n

Méthode naïve : $n-1$ multiplications

Partager les calculs

$$a \times a \times a \times a$$

$$a \times a \longrightarrow a^2$$

$$a^2 \times a^2 \longrightarrow a^4$$ 2 multiplications au lieu de 3

$$a^4 \times a^4 \longrightarrow a^8$$ 3 multiplications au lieu de 7

$$a^8 \times a^8 \longrightarrow a^{16}$$ 4 multiplications au lieu de 15

Diviser pour régner (pour multiplier)

ÉCOLE POLYTECHNIQUE ParisTech

Pour calculer a^n

si $n = 2 \cdot m$ (pair) : calculer $b = a^m$ le résultat est $b \times b$

si $n = 2 \cdot m + 1$ (impair) : calculer $b = a^m$ le résultat est $b \times b \times a$

si $n=1$ c'est fini

Code


```
A mult(A a, A b);

static A puissance(A a, int n){
    if (n == 1) return(a);
    A b = puissance(a, n / 2);
    A c = mult(b,b);
    if (n % 2 == 0) return(c);
    return(mult(c,a));
}
```

Analyse asymptotique

ÉCOLE POLYTECHNIQUE ParisTech

```
A mult(A a, A b);

static A puissance(A a, int n){
    if (n == 1) return(a);
    A b = puissance(a, n / 2);
    A c = mult(b,b);
    if (n % 2 == 0) return(c);
    return(mult(c,a));
}
```

c(*n*) =

c(*n*/2)

\+ 1

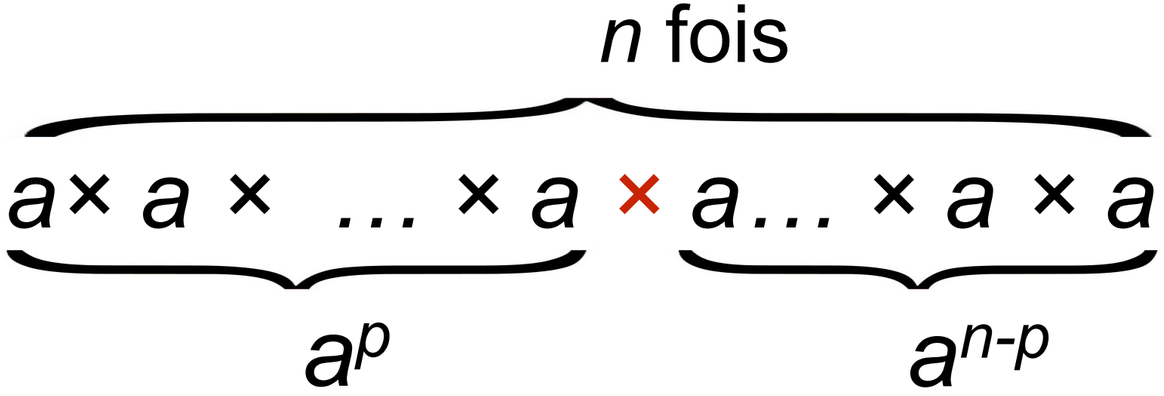
\+ 1

c(1) = 0

c(*n*) = 2 + c(*n*/2) c(*n*) = 2 $\log_2(n)$ (dans le pire cas)

(pires cas : 3 (cad. 11), 7 (111), 15 (1111), 31 (11111) …)

Essayons de trouver la *meilleure solution*



On s'est ramené à deux sous problèmes : p et $n-p$

Non !

Ca donnerait : $c(n) = \min_p(c(p) + c(n-p) + 1)$ et donc $c(n) = n - 1$

Mais alors on ne tient pas compte du partage des calculs entre a^p et a^{n-p}

Il ne faut *pas* utiliser la programmation dynamique ici