Algorithmique Algébrique

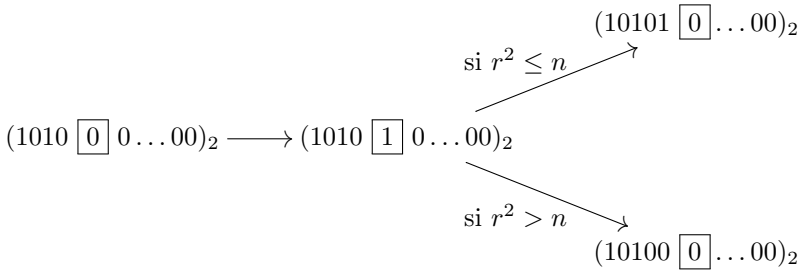
TD2 Exercice 8.

On propose l'algorithme suivant :

1. Supposons que n s'écrit en binaire $(n_k n_{k-1} \dots n_1 n_0)_2$ avec $n_k = 1$ (on a $2^k \leq n < 2^{k+1}$).
2. On pose pour première approximation de $\lfloor \sqrt{n} \rfloor$, $r = (10 \dots 00)_2$, un 1 suivi de $l = \lfloor k/2 \rfloor$ zéros ($r = 2^{\lfloor k/2 \rfloor}$).
3. On parcourt ensuite les chiffres de r de gauche à droite, à chaque étape on remplace le chiffre courant par un 1, on calcule r^2, si $r^2 \leq n$ on garde le 1 et on passe au chiffre suivant. Si $r^2 > n$ on remet un zéro et on passe au chiffre suivant.

   Autrement dit, à la i-ème étape on regarde le chiffre en position i (après le 1 en partant de la gauche), si on change le zéro en un 1 et que $r^2 \leq n$ on garde le 1 et on passe au chiffre suivant. Si $r^2 > n$ on remet un zéro et on passe au chiffre suivant.

   Exemple à l'étape 4 :



4. A la l-ème étape on fait de même et on arrête l'algorithme. Alors $r = \lfloor \sqrt{n} \rfloor$.

On peut maintenant calculer la complexité de cet algorithme, la détermination du bit de plus haut poids et le calcul de $\lfloor k/2 \rfloor$ ne compte pas. A chaque étape de l'algorithme on fait une comparaison entre r^2 et n, ce qui nécessite de calculer r^2. On peut majorer brutalement $r \leq n$, le calcul de r^2 est donc en $O(\log^2(n))$. On fait ce calcul $l = \lfloor k/2 \rfloor$ fois donc la complexité de l'algorithme total est en

$$O(l \log^2(n)) = O(k \log^2(n)) = O(\log^3(n)).$$

Remarque. *Si la majoration précédente vous semble brutale, on peut essayer* $r \leq 2^{\lfloor k/2 \rfloor + 1} \leq 2^{k/2+1}$ *donc le coût de la multiplication est en* $O(\log^2(2^{k/2+1})) = O((k/2+1)^2 \log^2(2)) = O(k^2) = O(\log^2(n))$. *On obtient le même résultat car le* O *est glouton et mange toute les constantes..*

Remarque. *Si* $2^k \leq n < 2^{k+1}$ *alors* $2^{k/2} \leq \sqrt{n} < 2^{(k+1)/2}$, *par les propriété de la partie entière,* $2^{\lfloor k/2 \rfloor} \leq 2^{k/2} \leq \sqrt{n} < 2^{(k+1)/2} \leq 2^{\lfloor k/2 \rfloor + 1}$ *donc* $2^{\lfloor k/2 \rfloor} \leq \lfloor \sqrt{n} \rfloor < 2^{\lfloor k/2 \rfloor + 1}$ *(si un nombre est compris entre deux entiers, sa partie entière aussi).*