Graphes et Nombres.

- Pour le prbl de Simon nous avions

$$(\mathbb{F}_2^n, \oplus) = G.$$

sous esp vect de dim k $= H$.

- provenance : $f : G \longrightarrow X = G/H$.

$$\begin{array}{ccc} \underline{x} & \longmapsto & f(\underline{x}) \\ \cap & & \cap \\ \mathbb{F}_2^n & & \mathbb{F}_2^n/H. \end{array}$$

$f(\underline{x}) = f(\underline{y})$ ssi $\underline{y} = \underline{x} \oplus \underline{h}$, $\underline{h} \in H$.

en d'autre termes ssi $\underline{y} - \underline{x} \in H$.

- Généralisation du prbl de Simon aux graphes.

et on l'appelle le prbl du sous-graphe caché

Pbl général du Simon ou du sous groupe caché :

$$f : G \longrightarrow G/H .$$

G commutatif

H s-groupe inconnu ou "caché"

$$g \longmapsto f(g)$$

propriété : $f(g_1) = f(g_2)$ ssi $g_1 g_2^{-1} \in H$

Not Multiplicative

Not add $g_1 + (-g_2) \in H$

problème : trouver un alg qui permet de reconstruire le H en posant un # min de queries à l'oracle $\xrightarrow[g]{}$ | Oracle. | $\longrightarrow f(g)$

L'alg Quant va la fois passée se généralise T.B à ce pbl du sous groupe caché .

Nouveau Problème : Recherche de la période d'une fonction arithmétique.

$$f : \mathbb{Z} \longrightarrow \mathbb{Z}$$
$$x \longmapsto f(x)$$

- **propriété** : $f(x) = f(x+r) \quad \forall x \in \mathbb{Z}$.
  pour $r \in \mathbb{Z} \setminus \{0\}$.

  c.c.d f possède période r.

- **problème** : calculer r grâce à un algo efficace.

1) Nos fcts d'intérêt ont une période énorme.
   (r possède peut-être 400 décimales).

2) L'équation $f(x) = f(x+r)$ on ne sait pas la résoudre en temps polynomial dans la "taille de r" c.c.d en temps poly $(\log_2 r)$.
   $\log_{10} r$.

- En fait ceci est un problème de s-grape caché.

$$\begin{cases} G = \mathbb{Z}. \quad ; \quad H = r\mathbb{Z}. \text{ mult de } r. \\ f(x) = f(y) \quad \text{ssi} \quad x - y \in r\mathbb{Z}. \end{cases}$$

exactement un "prob de Simon".

On peut espérer utiliser un algo quantique "à la Simon" pour calculer la période r en temps pol $\text{poly}(\log_2 r)$.

Le seul problème à priori est que G est infini et nous devons le "tronquer" pour utiliser un nombre fini de qubits.

$G = \mathbb{Z}$ mais $\frac{\mathbb{Z}}{M\mathbb{Z}} = \{0, 1, 2 \dots, M-1\}$ avec $M \gg r$.

Problème est que $H = r\mathbb{Z}$ est un s-grape seulement si $r \mid M$

Comment choisir M puisque r est inconnu ??

On va juste prendre M assez grand devant r sans essayer de rechercher si "$(r \text{ div } M)$". et on va voir que ce n'est finalement pas si grave.

↳ Travailler un peu avec l'Alg de Shor

Alg pour la recherche de la période d'une fct arithmétique.

La factorisation des entiers va se faire en appliquant cet algorithme à des fcts spéciales qui dépendent de l'entier N à factoriser.

La factorisation des entiers vue comme un problème de recherche de période d'une fct arithmétique.

- N entier peut être factorisé de façon unique en produit de # premiers. (Euclide)

$$N = p_1^{e_1} p_2^{e_2} \cdots p_K^{e_K}.$$

- difficile, algs classiques connus sont tous sous-exponentiels, les meilleurs algo :

$$O\left(\exp\left[\left(\frac{64}{9} b\right)^{1/3}\right] (\log b)^{2/3}\right) \quad \checkmark$$

avec $b = \log_2 N$. → (Longueur de N ou taille de N)

- Algo quantique (Shor) $O(b^3)$.

Algorithme classique qui fait intervenir la recherche de la période d'une fct arithmétique

a. Choisir aléatoirement uniformément $a \in \{2, \dots, N-1\}$

et on calcule (via Euclide) $d \equiv PGCD(a, N)$

— complexité est $O(b^3)$ où $b = \log_2 N$. —

b. Si $d > 1$ on a un facteur de N, on divise N par d et on retourne en a.

— Ceci a une proba très faible et "n'arrivera pas" —

c. Si $d = 1$ (a et N sont premiers entre eux) et on calcule le plus petit entier r t.q

$$a^r = 1 \mod N.$$

→ compl classique supérieur pour algs connus

Cet r s'appelle l'ordre de a mod N $= \mathrm{Ord}_N(a)$

d. Si r est impair → output FAIL. (retourne à a.)

e. Si r est pair alors on note que

$$a^r - 1 = (a^{r/2} - 1)(a^{r/2} + 1).$$

Suite du point c :

$$a^r - 1 = (a^{r/2} - 1)(a^{r/2} + 1)$$

si r est pair.

Notez que $a^r - 1 = 0 \mod N$ c.a.d que $a^r - 1$ est un multiple de N. Si bien que N div $a^r - 1$.

e1 N div $a^{r/2} - 1$. ⇝ impossible car alors $a^{r/2} = 1 \mod N$ mais r est le plus petit entier t.q $a^r = 1 \mod N$

e2 N div $a^{r/2} + 1$

⇝ possible pour a bas → FAIL et retourne à a

e3 N partage fact Non-triviaux avec $a^{r/2} - 1$ et $a^{r/2} + 1$

$d_\pm = PGCD(a^{r/2} \pm 1, N)$ sont non triviaux tous les deux > 1.

↳ deux facts de N on divise N et on retourne au pt a.

Remarque

Notez par $N = p \cdot q$, p et q premiers

alors $d_{\pm} = p$ et q et l'alg se termine tout de suite en e3.

- Pour finir avec cet algo il faut préciser sa

$$\text{prob}(\text{succès}) = 1 - \text{prob}(\text{échec})$$

pour $a \in \{2, \dots, N-1\}$ aléatoire.

Théorème en théorie des nombres (Rabin et Miller 1974).

$\text{prob}(\text{échec}) \leq \frac{1}{4}$ et $\text{prob}(\text{succès}) \geq \frac{3}{4}$.

Conséquent on peut amplifier cette probabilité en recommençant $O(\ln \epsilon)$ pour avoir

$$\text{prob}(\text{succès}) \geq 1 - \epsilon.$$

- Grâce à l'alg de Shor. Nous allons résoudre

le calcul de r t.q

$$a^r = 1 \mod N$$

r le plus petit possible.

- Vue plus large ici :

$$f_a : \mathbb{Z} \to \mathbb{Z}.$$

$$x \mapsto f_a(x) = a^x \mod N.$$

ici $f_a(x) = f_a(x+r)$ car en effet

$$a^x = a^{x+r} \iff 1 = a^r \mod N.$$

- Calculer $r = \mathrm{Ord}_N(a)$ $\iff$ Calculer la période de l'application arithmétique $f_a(x) = a^x \mod N$.