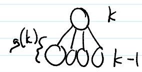
Algos de branchement

4 novembre 2021 10:21

Technique de design d'algo FPT qui consiste à brancher sur un # limité de cas possibles.
Pour avoir un algo FPT, il faut que :



- le # d'appels récursifs est borné par une fct $g(k)$
- le param k doit réduire à chaque appel récursif
  $\Rightarrow$ hauteur de l'arbre bornée par une fct $h(k)$

Avec ces 2 conditions, le # de noeuds est $O(g(k)^{h(k)})$

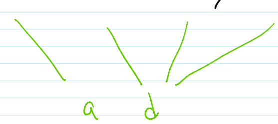
- s'il y a une solution, un des noeuds de l'arbre de récursion la trouve

Vertex-cover
$g(k) = 2$
$h(k) = k$
$\Rightarrow O(2^k)$

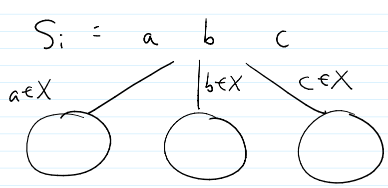
3-HITTING SET

Entrée : ensembles $S = \{S_1, S_2, \dots, S_m\}$ de taille 3 chacun
univers $U = \{u_1, \dots, u_n\}$
Param : k = taille de solution voulue
Sortie : $X \subseteq U$ (univers) tel que $|X| \leq k$
et tel que $\forall S_i,\ X \cap S_i \neq \emptyset$, ou null si un tel X n'existe pas

S_1	S_2	S_3	S_4	
abc	ade	dfg	deh	$k=2$



On prend $S_i \in S$ arbitraire

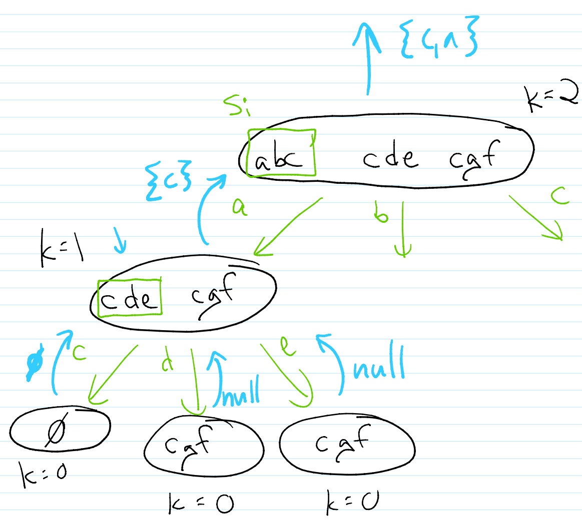


Toute solution X doit contenir a ou b ou c

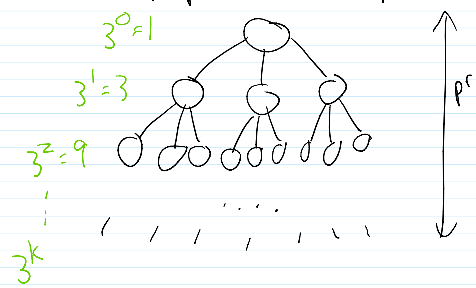
3hitset($S = \{S_1, \dots, S_m\}$, k)
  si $S = \emptyset$, return $\emptyset$
  si $k = 0$, return null

  S_i = ensemble de S
  Soit $S_i = \{a, b, c\}$
  $S_a = S \setminus \{S_j \in S : a \in S_j\}$
  X_a = 3hitset(S_a, $k-1$)
  $S_b = S \setminus \{S_j \in S : b \in S_j\}$
  X_b = 3hitset(S_b, $k-1$)
  $S_c = S \setminus \{S_j \in S : c \in S_j\}$
  X_c = 3hitset(S_c, $k-1$)

  si $X_a = X_b = X_c$ = null, return null
  si $X_a \neq$ null, return $X_a \cup \{a\}$
  si $X_b \neq$ null, return $X_b \cup \{b\}$
  return $X_c \cup \{c\}$



Cet algo crée un arbre récursif où chaque noeud a 3 enfants, et sa prof. est bornée par k

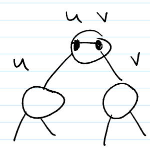


$O(3^k)$ noeuds

Chaque appel prend un temps $O(|S|) = O(m)$

$\Rightarrow$ Complexité: $O(3^k \cdot m)$

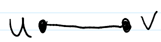
Vertex-Cover



$O(2^k)$

Autre idée de branchement $\rightarrow O(1{,}618^k)$

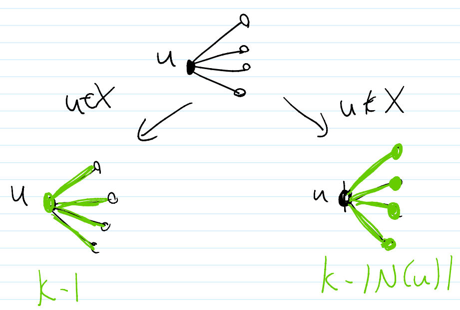
Soit $uv \in E$.



2 cas possibles pour une solution X

- $u \in X$ $\quad k-1$
- $u \notin X$ $\quad k$

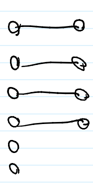
$\llcorner$ si $u \notin X$, alors tous ses voisins sont dans X



- Pour réduire le + possible, on choisit u de degré max

- On voudrait que $k-|N(u)|$ soit mieux que $k-1$
- S'il $\exists u \in V$ t.q. $|N(u)| \geq 2$, on n'a pas ce pb.
- Il y a un pb si $|N(u)| \leq 1 \quad \forall u \in V$

Si c'est le cas, vertex-cover est facile car G est



Dans ce cas, le vertex-cover est $|E|$

vcFPT2 ($G=(V,E)$, k)
 si $|E|=0$, return $\emptyset$

vcFPT2 ($G=(V,\mathcal{E})$, k)

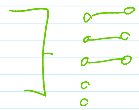
si $|E|=0$, return $\emptyset$

si $k=0$, return null

si $|N(u)| \leq 1 \ \forall u \in V$

si $|E|>k$, return null

return {un sommet par arête}



// à partir d'ici, $\exists u$ t.q. $|N(u)| \geq 2$

Soit u de degré maximum

$G_u = G - u$

$X_u = \text{vcFPT2}(G_u, k-1)$

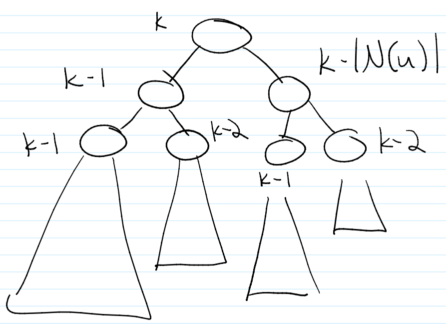
$G_{Nu} = G - N(u)$

$X_{Nu} = \text{vcFPT2}(G_{Nu}, k-|N(u)|)$

si $X_u \neq$ null, return $X_u \cup \{u\}$

si $X_{Nu} \neq$ null, return $X_{Nu} \cup N(u)$

return null



on peut supposer que $k-|N(u)| \leq k-2$ car $|N(u)| \geq 2$

Soit $f(k)$ le # de noeuds de l'arbre récursif

On a

$$f(k) = f(k-1) + f(k-2)$$

Ceci donne que le # de noeuds est $O(1.619^k)$

On ne sait pas $f(k)$ c'est quoi, mais c'est "sûrement" exponentiel.

$f(k) = \alpha^k$ on cherche le α

On a

$$f(k) = f(k-1) + f(k-2)$$

$$\alpha^k = \alpha^{k-1} + \alpha^{k-2} \qquad / \alpha^{k-2}$$

$$\alpha^2 = \alpha^1 + 1$$

$$\alpha^2 - \alpha^1 - 1 = 0$$

$$\alpha \in \{-0{,}61803.., \underline{1{,}618...}\}$$

$$f(k) = c_1 \cdot 1{,}618^k + c_2 \cdot (-0{,}6)^k$$

$\alpha \in \{-0.61803..., \underline{1.618...}\}$

$f(k) = c_1 \cdot 1.618^n + c_2 \cdot (-0.6)$

$\hookrightarrow \alpha = 1.618... < 1.619$

$\Rightarrow f(k) = 1.619^k$

$\Rightarrow \#\text{rounds} \in O(1.619^k)$