Tesis de Licenciatura en Ciencias de la Computación

Sobre la thinness de árboles y otras clases de grafos

Tesistas: Agustín Sansone y Eric Brandwein
Directora: Flavia Bonomo
Codirectora: Carolina Lucía González

Departamento de Computación, FCEyN, Universidad de Buenos Aires

10 de marzo de 2022

Esquema

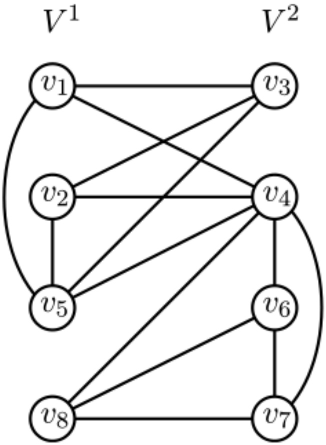
Introducción

Definición (k-thin)

Un grafo $G = (V, E)$ es k-thin si existe un orden $\sigma = v_1, \dots, v_n$ de V y una partición S de V en k clases tales que, para cada tripla (r, s, t) con $r < s < t$, si v_r y v_s pertenecen a la misma clase y $(v_t, v_r) \in E$, entonces $(v_t, v_s) \in E$. Un orden y partición que satisfacen estas propiedades es llamado solución consistente.

Definición (Thinness)

Mínimo k tal que G es k-thin.



Aplicaciones

Problemas NP-completos $\longrightarrow$ Solución polinomial!

- *Maximum weighted independet set* problem (C. Mannino et al., 2007)
- *Bounded coloring* problem (F. Bonomo et al., 2009)

Aplicaciones

Problemas NP-completos $\longrightarrow$ Solución polinomial!

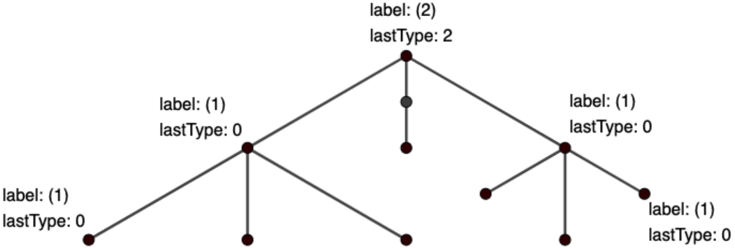
- *Maximum weighted independet set* problem (C. Mannino et al., 2007)
- *Bounded coloring* problem (F. Bonomo et al., 2009)

Estado del arte

- Problema general $\longrightarrow$ NP-completo (Y. Shitov, 2021)
- Problema dado un orden $\longrightarrow$ $\mathcal{O}(n^3)$ (F. Bonomo, D. De Estrada, 2019)
- Problema dada una partición $\longrightarrow$ NP-completo (F. Bonomo, D. De Estrada, 2019)

Thinness de árboles

Estructura del algoritmo

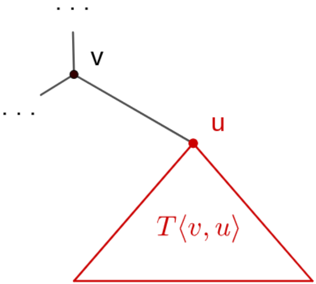


- *label(T)* $= (a_1, \ldots, a_p) \in \mathbb{N}^p$, donde $a_1 = \text{thin}(T)$.
- *lastType(T)* $= n \in [0, 3]$.

Árbol colgante $T\langle v, u\rangle$ (Dangling tree)

Definición

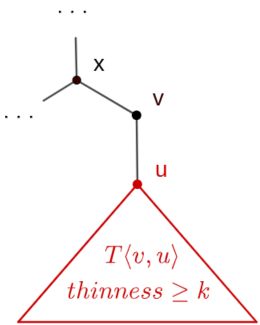
Sea T un árbol con vértices v y u adyacentes. El árbol colgante de v a u, $T\langle v, u\rangle$, es la componente de $T \setminus (u, v)$ que contiene a u.



k-vecino (k-neighbor)

Definición

Sean x y v vecinos en un árbol T. Decimos que v es un k-vecino de x si existe un vértice $u \neq x$ vecino a v tal que $\text{thin}(T\langle v, u\rangle) \geq k$.



Índice de k-componente (k-component index)

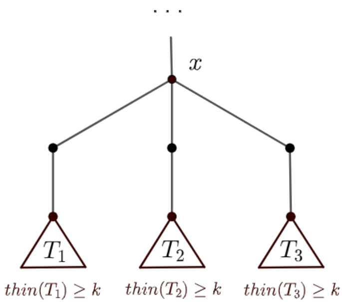
Definición

El índice de k-componente de x es la cantidad de k-vecinos de x, y lo notamos $D(x, k)$.

k-saturación

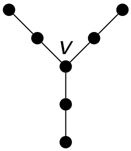
Teorema (k-saturación)

Sea T un árbol, entonces $\text{thin}(T) \geq k + 1$ si y sólo si $D(x, k) \geq 3$ para algún vértice x en T. Decimos que x está k-saturado.

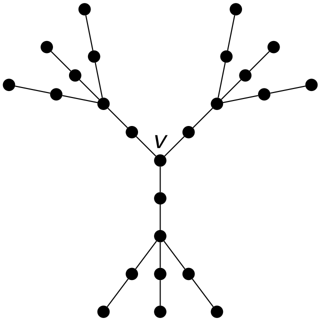


Árboles más chicos para cada thinness

$\mathrm{thin}(T) = 1$



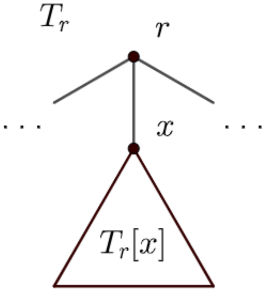
$\mathrm{thin}(T) = 2$



$\mathrm{thin}(T) = 3$

Árboles enraizados $T_r[x]$ (Rooted trees)

- T_r: árbol T enraizado en r.
- $T_r[x]$: subárbol de T_r enraizado en x.



Vértice k-crítico (k-critical vertex)

Definición

Sea T_r un árbol enraizado en r. Decimos que un vértice x en T_r es k-crítico si tiene exactamente dos hijos v_1 y v_2 que tienen por lo menos un hijo cada uno, u_1 y u_2 respectivamente, tal que $\mathrm{thin}(T_r[u_1]) = \mathrm{thin}(T_r[u_2]) = k$.

Lema

Un árbol T_r con thinness k tiene como máximo un vértice k-crítico.

Vértice k-crítico (k-critical vertex)

Definición

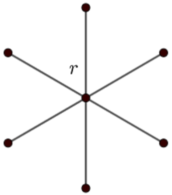
Sea T_r un árbol enraizado en r. Decimos que un vértice x en T_r es k-crítico si tiene exactamente dos hijos v_1 y v_2 que tienen por lo menos un hijo cada uno, u_1 y u_2 respectivamente, tal que $\mathrm{thin}(T_r[u_1]) = \mathrm{thin}(T_r[u_2]) = k$.

Lema

Un árbol T_r con thinness k tiene como máximo un vértice k-crítico.

Tipos de árboles - Tipo 0

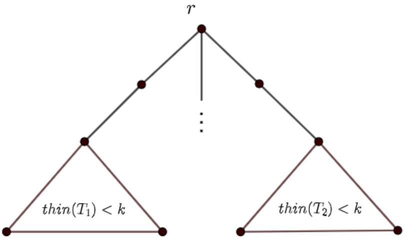
Decimos que T_r es un árbol de **tipo 0** si r es una hoja o si todos los hijos de r son hojas.



- $label(T_r) = (1)$
- $lastType(T_r) = 0$

Tipos de árboles - Tipo 1

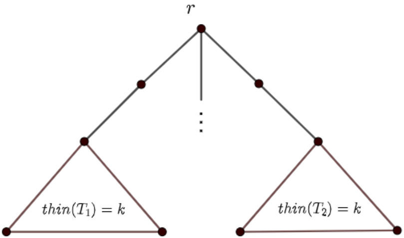
T_r es de **tipo 1** si no es de tipo 0 y no contiene ningún vértice k-crítico.



- $label(T_r) = (k)$
- $lastType(T_r) = 1$

Tipos de árboles - Tipo 2

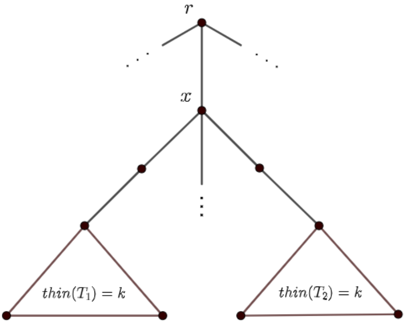
T_r es de **tipo 2** si r es su único vértice k-crítico.



- $label(T_r) = (k)$
- $lastType(T_r) = 2$

Tipos de árboles - Tipo 3

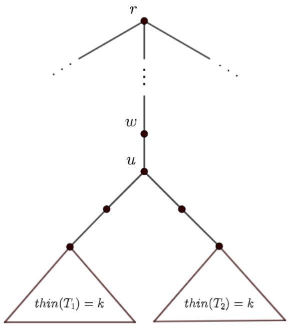
T_r es de **tipo 3** si existe un hijo de r en T_r que es k-crítico.



- $label(T_r) = (k)$
- $lastType(T_r) = 3$

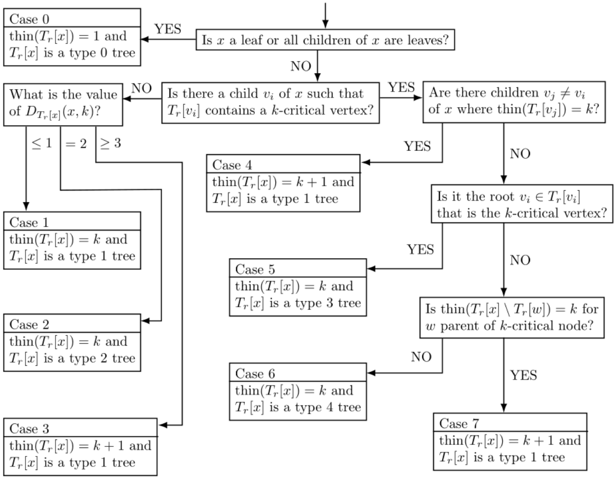
Tipos de árboles - Tipo 4

T_r es de **tipo 4** si hay un vértice k-crítico u que no es ni r ni un hijo de r.



- $label(T_r) = k \oplus label(T_r \setminus T_r[w])$
- $lastType(T_r) = lastType(T_r \setminus T_r[w])$

Árbol de decisión



Complejidad $\mathcal{O}(n\log(n))$

Cada vértice es visitado como máximo $|label(r)| \leq \text{thin}(T_r)$ veces:

1. en la llamada recursiva a su padre.
2. en la llamada recursiva a su abuelo.

Como $\text{thin}(T) \in \mathcal{O}(\log(n)) \implies$ máximo $\mathcal{O}(n\log(n))$ accesos a vértices.

Complejidad $\mathcal{O}(n\log(n))$

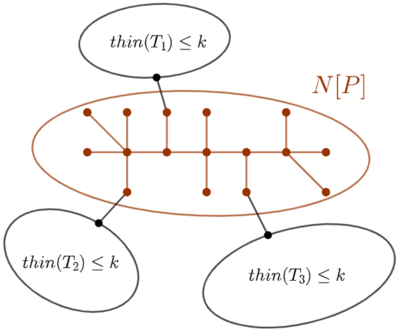
Cada vértice es visitado como máximo $|label(r)| \leq \text{thin}(T_r)$ veces:

1. en la llamada recursiva a su padre.
2. en la llamada recursiva a su abuelo.

Como $\text{thin}(T) \in \mathcal{O}(\log(n)) \implies$ máximo $\mathcal{O}(n\log(n))$ accesos a vértices.

Lema (Path Layout Lemma)

Sea T un árbol. Si existe un camino $P = (x_1, \ldots, x_p)$ en T tal que cada componente conexa de $T \setminus N[P]$ tiene thinness menor o igual a k, entonces $\text{thin}(T) \leq k + 1$. Más aún, dadas las soluciones consistentes de las componentes en máximo k clases podemos en tiempo lineal calcular una solución consistente para T en máximo $k + 1$ clases.



Solución consistente

1. Encontrar un camino P tal que $\text{thin}(T_r \setminus N[P]) < k$.
2. Llamar esta función recursivamente en todos los árboles de $T_r \setminus N[P]$.
3. Usar el Path Layout Lemma para encontrar una solución.

Complejidad temporal y espacial $\mathcal{O}(n \log(n))$

Solución consistente

1. Encontrar un camino P tal que $\text{thin}(T_r \setminus N[P]) < k$.
2. Llamar esta función recursivamente en todos los árboles de $T_r \setminus N[P]$.
3. Usar el Path Layout Lemma para encontrar una solución.

Complejidad temporal y espacial $\mathcal{O}(n \log(n))$

Cotas de thinness en árboles

Teorema

La thinness de un árbol T con n vértices es $\mathcal{O}(\log(n))$. De hecho $\text{thin}(T) \leq \log_3(n+2)$.

Teorema

Un árbol no trivial de thinness k tiene al menos $\frac{3^{k-1}+3}{2}$ hojas. En particular, la thinness de un árbol con ℓ hojas es al menos $\log_3(6\ell - 9)$.

Teorema

Sea T un árbol, $thin(T) - lmimw(T) \leq 1$

Cotas de thinness en árboles

Teorema

La thinness de un árbol T con n vértices es $\mathcal{O}(\log(n))$. De hecho $\text{thin}(T) \leq \log_3(n+2)$.

Teorema

Un árbol no trivial de thinness k tiene al menos $\frac{3^{k-1}+3}{2}$ hojas. En particular, la thinness de un árbol con ℓ hojas es al menos $\log_3(6\ell - 9)$.

Teorema

Sea T un árbol, $thin(T) - lmimw(T) \leq 1$

Cotas de thinness en árboles

Teorema

La thinness de un árbol T con n vértices es $\mathcal{O}(\log(n))$. De hecho $\operatorname{thin}(T) \leq \log_3(n+2)$.

Teorema

Un árbol no trivial de thinness k tiene al menos $\frac{3^{k-1}+3}{2}$ hojas. En particular, la thinness de un árbol con ℓ hojas es al menos $\log_3(6\ell - 9)$.

Teorema

Sea T un árbol, $thin(T) - lmimw(T) \leq 1$

Cotas de thinness en árboles

Teorema

La thinness de un árbol T con n vértices es $\mathcal{O}(\log(n))$. De hecho $\text{thin}(T) \leq \log_3(n+2)$.

Teorema

Un árbol no trivial de thinness k tiene al menos $\frac{3^{k-1}+3}{2}$ hojas. En particular, la thinness de un árbol con ℓ hojas es al menos $\log_3(6\ell - 9)$.

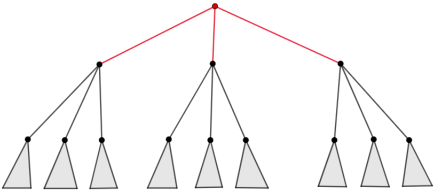
Teorema

Sea T un árbol, $thin(T) - lmimw(T) \leq 1$

Thinness en árboles m-arios completos

Teorema

Sea un natural $m \geq 3$ y T un árbol m-ario completo de altura h, entonces $\text{thin}(T) = \lceil \frac{h+1}{2} \rceil$.



Teorema

Sea T un árbol binario completo de altura h, $\text{thin}(T) = \lceil \frac{h+1}{3} \rceil$

Más cotas

Teorema

Sea T un árbol y d su diámetro, entonces $\operatorname{thin}(T) \leq \lceil \frac{d+1}{4} \rceil$. *Además, si el grado máximo de los vértices en T es como mucho 3, entonces* $\operatorname{thin}(T) \leq \lceil \frac{d+3}{6} \rceil$.

Teorema

Sea T un árbol con t casi-hojas con $t \geq 2$, entonces $thin(T) \leq t - 1$.

Más cotas

Teorema

Sea T un árbol y d su diámetro, entonces $\text{thin}(T) \leq \lceil \frac{d+1}{4} \rceil$. *Además, si el grado máximo de los vértices en T es como mucho* 3, *entonces* $\text{thin}(T) \leq \lceil \frac{d+3}{6} \rceil$.

Teorema

Sea T un árbol con t casi-hojas con $t \geq 2$, entonces $thin(T) \leq t - 1$.

Más cotas

Teorema

Sea T un árbol y d su diámetro, entonces $\mathrm{thin}(T) \leq \left\lceil \frac{d+1}{4} \right\rceil$. *Además, si el grado máximo de los vértices en T es como mucho* 3, *entonces* $\mathrm{thin}(T) \leq \left\lceil \frac{d+3}{6} \right\rceil$.

Teorema

Sea T un árbol con t casi-hojas con $t \geq 2$, entonces $thin(T) \leq t - 1$.

Thinness en otras clases

CR_n - Grafos corona (crown graphs)

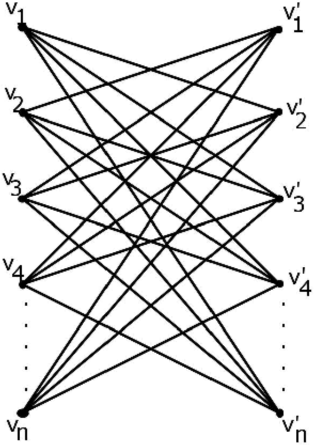


Figura: CR_n

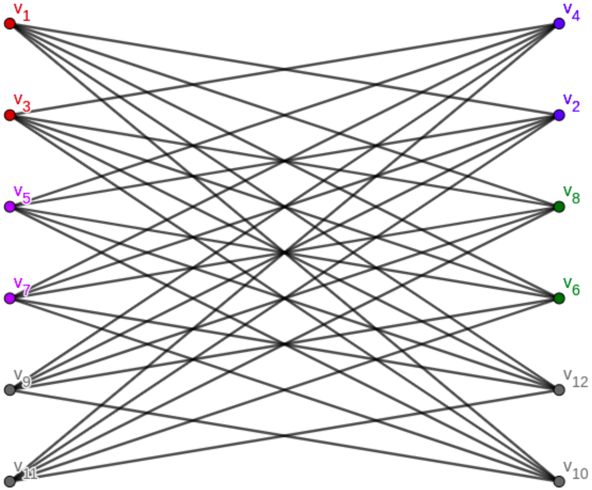
CR_n - Grafos corona (crown graphs)

- Cota conocida: $thin(CR_n) \geq \frac{n}{2}$ (F. Bonomo et al., 2020)
- $thin(CR_n) = max(1, n-1)$

CR_n - Grafos corona (crown graphs)

- Cota conocida: $thin(CR_n) \geq \frac{n}{2}$ (F. Bonomo et al., 2020)
- $thin(CR_n) = max(1, n-1)$

Ejemplo *solución consistente* con $n-1$ pilas



GR_r - Grafos grilla (grid graphs)

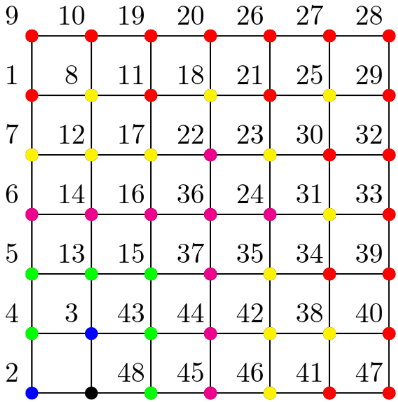


Figura: GR_7 (con solución consistente)

GR_r - Grafos grilla (grid graphs)

- Cota conocida: $\frac{r}{4} \leq thin(GR_r) \leq r + 1$ (C. Manninno et al, 2007)
- $thin(GR_r) \leq \lceil \frac{2r}{3} \rceil$

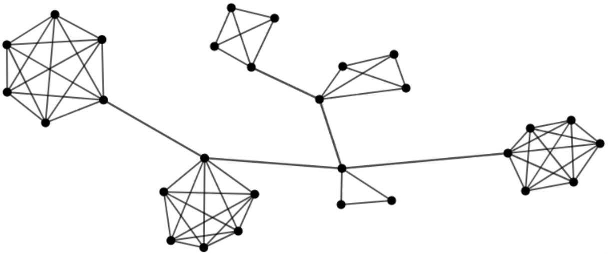
GR_r - Grafos grilla (grid graphs)

- Cota conocida: $\frac{r}{4} \leq thin(GR_r) \leq r + 1$ (C. Manninno et al, 2007)
- $thin(GR_r) \leq \lceil \frac{2r}{3} \rceil$

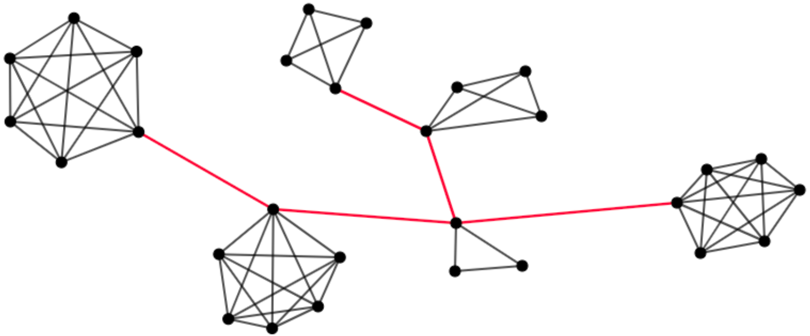
Heurísticas

Teorema

Dado un árbol de cliques G, existe algoritmo $\mathcal{O}(n \log n)$ que retorna una solución consistente usando k clases garantizando que $k - thin(G) \leq 1$



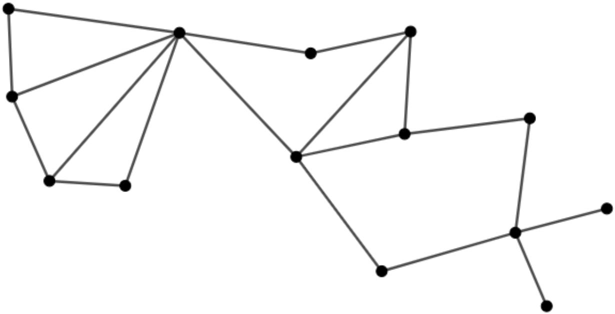
Heurísticas



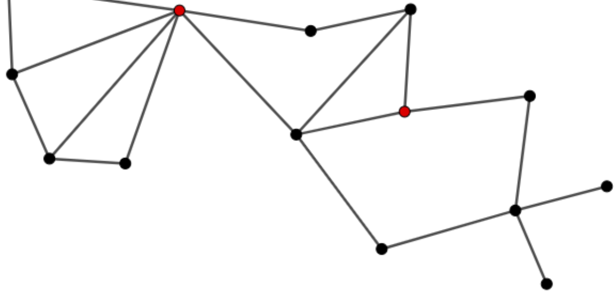
Heurísticas

Teorema

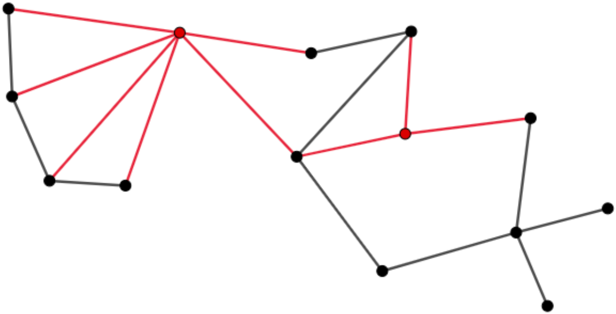
Dado un grafo G y un conjunto de vértices S en $V(G)$, G es $(|S| + thin(G - S))$-thin



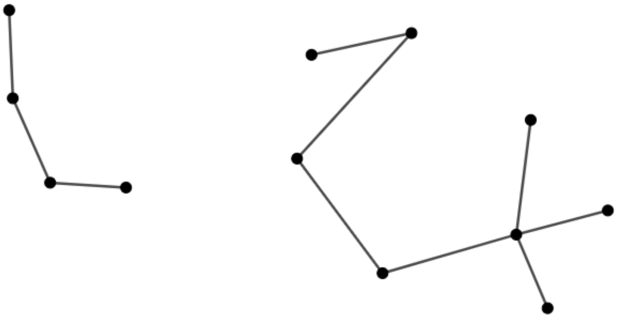
Heurísticas



Heurísticas



Heurísticas



Trabajo futuro

Trabajo futuro

- Caracterizar grafos G tal que tienen un subconjunto de vértices S tales que $G - S$ es un bosque y $|S| = \mathcal{O}(min(log(n), d))$
- Lo mismo para el árbol de cliques
- Investigar calcular thinness para árboles en tiempo lineal
- Explorar usar las ideas del *k-component index theorem* para otras clases de grafos
- Pensar si se puede usar alguna idea para *proper, independent* thinness en árboles

¡Muchas gracias!

¿Preguntas?