1 Clase 1

1.1 Info de la materia

Mail del profesor. daniel.penazzi@unc.edu.ar
Temas a ver.

- Coloreo de grafos
- Flujos en network
- Matchings
- Códigos de correción de errores
- P-NP (Complejidad computacional)
- Inteligencia artifical

La materia tiene tres partes: teórico, práctico y proyecto de programación. Solo la parte práctica tiene promoción (se explica abajo). El final tiene parte teórica y parte práctica. La parte teórica es demostrar uno de tres teoremas dados a priori. La parte práctica tiene ejercicios de demostración o pensamiento y de resolución de problemas.

La parte práctica se promociona si se aprueban los dos parciales, con cualquer nota ≥ 4. De promocionarlo, la parte práctica del final no es necesaria.

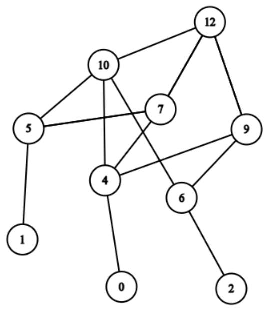
El proyecto de programación tiene dos partes. La primera es leer un grafo y cargar los datos al programa. La segunda es un problema de coloreo de grafos. La fecha de entrega de la parte uno es en dos o tres semanas a partir de hoy (13/03). La parte importante es la parte 2.

La biliografía está en el programa 2023.

1.2 Grafos

Definition 1 *Una grafo es una* 2*-upla* $G = (V, E)$ *con* V *un conjunto cualquiera (finito) y* $E \subseteq \{A \subseteq V : |A| = 2\}$

> **Nota.** La restricción de finitud es sólo de esta materia.



Los elementos de V se llaman vértices o nodos. Los elementos de E se llaman lados o aristas. Por convención, a menos que digamos lo contrario, es que $|V| = n$ y $|E| = m$.

Definition 2 *Un camino en un grafo $G = (V, E)$ es una sucesión de vértices $v_1, \ldots, v_r$, con $v_i \in V$ para todo i, tal que $\{v_j, v_{j+1}\} \in E$ para todo $1 \leq j < r$.*

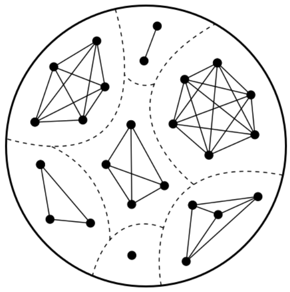
Dado un camino $v_1, \ldots, v_r$, si $v_1 = x, v_r = y$, decimos que es un camino de x a y. Para todo $G = (V, E)$ definimos la relación binaria

$$\sim \; := \left\{(x, y) \in V^2 : \text{ existe un camino de } x \text{ a } y \right\}$$

Es decir, $x \sim y$ denota la relación de que existe un camino entre x e y. Es trivial comproabar que $\sim$ es una relación de equivalencia. Cada clase de equivalencia $a/\!\sim$ con $a \in V$ se llama una componente conexa de G.

Definition 3 *Decimos que G es conexo si y solo si tiene una sola componente conexa. Es decir, si $|V \sim \;| = 1$.*

> El profesor no mencionó esto pero es lindo recordar (si alguien ha cursado lógica) que el conjunto de clases de equivalencia A/R de un conjunto a sobre una relación binaria R puede en sí mismo darse como un grupo de grafos disconexos. Por ejemplo, abajo se dan los grafos de un espacio cociente con siete clases de equivalencia; cada par de vértices unidos por un lado corresponde a dos elementos equivalentes.



Fíjense que de esto se sigue un dato curioso (aunque tal vez irrelevante): Si $G = (V, E)$ es un grafo conexo con n vértices, el grafo que describe la clase de equivalencia de V es K_n.

Definition 4 *Decimos que un grafo $H = (W, F)$ es un subgrafo de $G = (V, E)$ si $W \subseteq V, F \subseteq E$.*

A veces usamos $H \subseteq G$ para decir "H es un subgrafo de G", pero no debe entenderse por esto que H y G son conjuntos.

Observe que no todo $W \subseteq V, F \subseteq E$ satisfacen que (W, F) es un grafo. Por ejemplo, si $F = \emptyset$ tenemos $F \subseteq E$, pero F no cumple la propiedad de que todos sus elementos sean conjuntos con cardinalidad 2.

Definition 5 (Densidad) *Decimos que un grafo es denso si $m = O(n^2)$. Decimos que un grafo es raro si $m = O(n)$.*

Random fact. Recuerde que "raro" no sólo significa "inusual" sino que es el antónimo de "denso". La etimología inglesa es más interesante: La palabra *Weird* (raro) significaba, en la edad media, destino. Por eso, en la balada medieval *True Thomas*, se lee "Weird shall never daunton me": El destino nunca ha de asustarme. Se debe a una antigua leyenda nórdica en la cual las *Weird sisters*, diosas terribles, tejían el destino de los hombres. Si le da curiosidad:

https://en.wikipedia.org/wiki/Three$_{Witches}$

Definition 6 *Dado $G = (V, E)$, si $x \in V$, $\Gamma(x) := \{y \in V : \{x, y\} \in E\}$ se llama el vecindario de x.*

Si $y \in \Gamma(x)$, decimos que y es un vecino de x. El grado de x, denotado $d(x)$, es la cantidad de vecinos de x; es decir, $d(x) = |\Gamma(x)|$. Usamos $\delta = \min \{d(x) : x \in V\}$ y $\Delta = \max \{d(x) : x \in V\}$. Si $\delta = \Delta$ se dice que G es regular. Por ejemplo, los grafos cíclicos y los completos son regulares.

1.3 Repaso de BFS y DFS

A completar.

1.4 Los grafos K_n y C_n

- K_n : El grafo completo en n vértices se define

$$K_n = (\{1, 2, \ldots, n\}, \{\{x, y\} : x, y \in \{1, 2, \ldots, n\}\})$$

  Es el grafo de n elementos donde todos los vértices están conectados unos con otros. Resulta que que $m = \binom{n}{2}$. Lo cual implica que $m = O(n^2)$.

- C_n: El grafo cíclico

$$C_n = (\ 1, 2, \ldots, n\ , \{12, 23, 34, \ldots, (n-1)n, n1\})$$

  Una observación es que $C_3 = K_3$; pero de allí en adelante difieren.

1.5 Coloreo de grafos

Definition 7 *Un coloreo propio de $G = (V, E)$ con k colores es una función*

$$C : V \mapsto A$$

con $|A| = k$ y tal que $xy \in A \Rightarrow C(x) \neq C(y)$.

Intuitivamente, un coloreo asigna k propiedades a los vértices de modo tal que ningún par de grafos adyacentes cumple la misma propiedad.

> **Nota.** Hace unos meses escrbí un algoritmo de coloreo en C. Es el segundo algoritmo dado en esta entrada:
>
> https://slopezpereyra.github.io/2023-10-29-Hamiltonian/
>
> No prometo que sea muy prolijo o esté bien explicado; ya en general uno es tonto y encima de tonto no sabe de grafos. Pero tal vez a alguien le sirva, qué se yo.

Definition 8 *El número cromático de un grafo $G = (V, E)$ es*

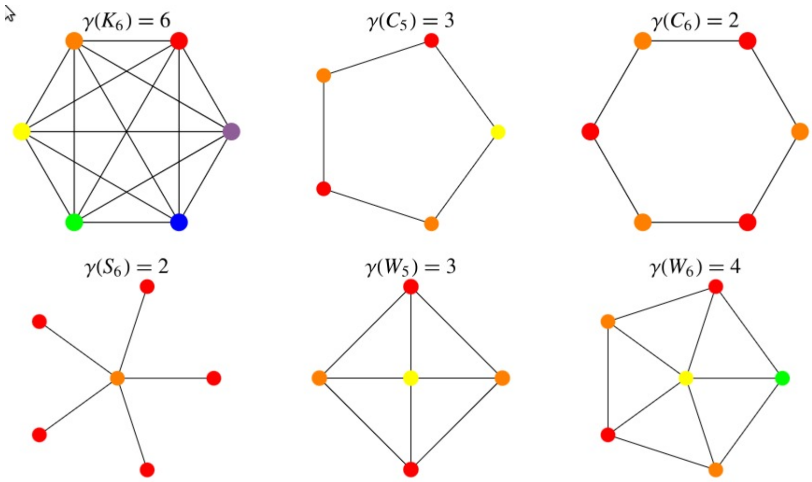
$$\chi(G) = \min_k (\exists \text{ coloreo propio de } G \text{ con } k \text{ colores})$$

No se conoce un algoritmo polinomial que calcule $\chi(G)$. El proyecto será dar un algoritmo polinomial que se aproxime a χ.

1.6 Un algoritmo greedy de coloreo

Damos un algoritmo que colorea un grafo G con vértices $v_1, \ldots, v_n$ y colores $c_1, c_2, \ldots, c_n$. Para que el algoritmo funcione, los colores y los vértices deben tener un orden (en nuestro caso dado por los subíndices).

> **Invariante del algoritmo.** Los coloreos parciales son propios. Es decir, a medida que se va coloreando iterativamente el grafo, en cada paso el coloreo resultante debe ser propio.
>
> **Pasos del algoritmo**.
>
> *(1)* $C(v_1) = c_1$.
>
> *(2)* $C(v_k)$ = mínimo color que mantenga un coloreo propio (que satisfaga el invariante).



1.7 Acotando χ

Generalmente nos interesa encontar $\chi(G)$ dado un grafo $G = (V, E)$. Damos unas pautas y observaciones generales para acotar $\chi(G)$ y así facilitar su hallazgo.

Lemma 1 *Si existe un coloreo propio de $G = (V, E)$ con k colores, entonces $\chi(G) \leq k$.*

> **Proof.** Es trivial por definición de χ (χ es el mínimo k en el conjunto de los coloreos posibles de G con k colores).

El lema significa que para acotar χ por arriba solo basta dar un coloreo con k colores.

Lemma 2 *Si H es un subgrafo de G, $\chi(H) \leq \chi(G)$.*

Este lema nos dice que podemos acotar χ por abajo si encontramos un subgrafo de G cuyo número cromático es conocido. Es fácil ver que $\chi(K_r) = r$ para todo $r > 1$. Es menos directo pero en clase se demostró que

$$\chi(C_r) = \begin{cases} 2 & r \mod 2 \equiv 0 \\ 3 & r \mod 2 \equiv 1 \end{cases}$$

Esto, en combinación con el último lema, nos dice que podemos acotar $\chi(G)$ por abajo simplemente observando si G contiene algún C_r o K_r como subgrafo. En el caso C_r, la cota inferior dada en el caso par, con $\chi(C_r) = 2$, es trivial (todo grafo necesita al menos dos colores). Por eso nos quedamos con el siguiente teorema:

Theorem 1 *Sea $G = (V, E)$ un grafo. Si $C_r \subseteq G$ con r impar entonces $\chi(G) \geq 3$. Si $K_r \subseteq G$ entonces $\chi(G) \geq r$.*

Theorem 2 *Sea $G = (V, E)$ un grafo. Entonces*

$$\chi(G) = \max\{\chi(C) : C \textit{ es componente conexa de } G\}$$

Theorem 3 *Sea $G = (V, E)$ un grafo. Entonces G contiene un ciclo impar si y solo si $\chi(G) \geq 3$.*

Proof. ($\Rightarrow$) Trivial.

($\Leftarrow$) Damos una prueba algorítmica. La prueba es simple, pero debe tenerse presente que usamos **tres** grafos diferentes en ella: el grafo $G = (V, E)$ del teorema, una componente conexa $C \subseteq G$, y un grafo $\mathcal{B} \subseteq C$ resultante de correr BFS de a partir de un vértice de C. El algoritmo determina si $\chi(G) = 2$ y de no serlo construye un ciclo impar. Usamos la notación $n(z)$ si $z \in V$ para denotar el nivel de z en $\mathcal{B}$. Dividimos la prueba en tres partes.

(1) Como $\chi(G) = \max\{\chi(C) : C \text{ es componente conexa de } G\}$, tenemos que

$$\chi(G) \geq 3 \Rightarrow \exists \text{ c.c. } C \text{ tal que } \chi(C) \geq 3$$

Elegimos $x \in C$ y corremos BFS a partir de él, generando un subgrafo $\mathcal{B} \subseteq C$ con todos los vértices de C (pero no todos los lados).

(2) Coloreamos cada vértice de C como sigue: como sigue: $c(z)$ será el nivel BFS de z módulo 2. Esto equivale a hacer $c(x) = 0$ y cada vez que y agrega a z a la cola de BFS, hacer $c(z) = 1 - c(y)$.

Esto da un coloreo con 2 colores, pero podría no ser propio (por ejemplo, dos vecinos de la raíz que son vecinos entre sí tendrían ambos color 1). Por lo tanto verificamos si es propio o no.

(3) Si es propio tenemos un coloreo propio de dos colores y $\chi(G) = 2$. Si es impropio, construiremos un ciclo impar. En particular, si no es propio, existen $u, v \in V$ tales que $c(u) = c(v)$ y $uv \in E$. Pero si $c(u) = c(v)$, entonces

$$n(u) \equiv n(v) \mod 2$$

Pero en un árbol siempre existe un camino de la raíz a cualquier vértice arbitrario. Entonces, en $\mathcal{B}$ existe un camino desde x hasta u, y existe un camino desde x hasta v. En particular, existe un vértice z que está en ambos caminos y a partir del cual los caminos se separan (note que puede ser x). Pero en C, uv forman un lado. Luego, en C tenemos el ciclo z, u, v, z.

¿Cuántos lados tiene el ciclo? Vea que de z a u hay $n(u) - n(z)$ lados. De u a v un solo lado, y de v a z otra vez $n(v) - n(z)$. En total, hay

$$n(u) + n(v) - 2n(z) + 1$$

lados. Pero $n(u) \equiv n(v) \mod 2$ y su suma es par. Luego la suma anterior es impar. Luego existe un ciclo impar en G.

Theorem 4 *Para todo gafo $G = (V, E)$, $\chi(G) \leq \Delta + 1$.*

Proof. Greedy siempre colorea con $\Delta + 1$ colores o menos. Tome cualquier orden $v_1 \dots v_n$ sobre V. Para colorear un vértice cualquiera excepto v_1, se miran los vecinos anteriores. El peor caso posible es que todos los vecinos sean vecinos anteriores y todos tengan colores distintos. En ese caso, greedy no considera ninguno de esos colores; es decir, $d(x)$ colores. Como $d(x) \leq \Delta$, entonces greedy, al colorear x, elimina a lo sumo Δ colores. Por lo tanto, en $\{1, 2, \dots, \Delta, \Delta + 1\}$ habrá al menos un color disponible. ■

Una pregunta natural es si hay grafos que alcancen la cota. La respuesta es sí: a saber,

- $\chi(K_n) = n$ y $\Delta = n - 1$.
- $\chi(C_{2r+1}) = 3$ y $\Delta = 2$.

Theorem 5 (Brooks) *Sea $G = (V, E)$ un grafo conexo distinto de un K_n y un C_{2r+1}. Entonces $\chi(G) \leq \Delta$.*

Proof. Complicada, una hora y media o dos de hacer. Triste.

Theorem 6 (Baby Brooks o Brooks para bebés) *Si $G = (V, E)$ un grafo conexo y no regular, entonces $\chi(G) \leq \Delta$.*

Proof. Es un caso particular del teorema de Brooks. Corramos BFS(x) con $d(x) = \delta$. Como G es conexo, obtenemos todos los vértices. Pues los vértices son agregados iterativamente al árbol del BFS, existe un orden dado precisamente por el orden de inserción al árbol.

Daremos un orden (que no es el anterior) tal que Greedy colorea G con a lo sumo Δ colores. En particular, es el orden inverso al anterior. Es decir que el último elemento del orden es la raíz x.

Como siempre el color del primer elemento es 1. Sea $z \neq v_1$ (en el orden del Greedy). El peor caso posible, que z tenga como vecinos a todos los anteriores, es imposible aquí, porque en el orden de DFS todo $y \neq x$ es insertado en el árbol por un vértice que ya estaba en el árbol. Y ese vértice debe ser un vecino. Es decir, en el orden de inserción del BFS, todo vértice tiene un vecino que es anterior en el orden. Entonces, en el orden inverso, todo $y \neq x$ tiene al menos un vecino posterior. Entonces, en el peor de los casos, tiene $d(y) - 1$ vecinos anteriores. Por lo tanto, greedy elimina a lo sumo $d(y) - 1 \leq \Delta - 1$ colores. $\therefore$ Siempre puede colorear a y con un color en $\{1, 2, \dots, \Delta\}$.

Cuando llega a x, Greedy elimina a lo sumo $d(x) = \delta$ colores. Y como G no es regular, $\delta < \Delta$. $\therefore$ Existe al menos un color para x.

1.8 ¿Para qué acotar χ? Para encontar χ!

En general no es fácil mostrar que $\chi(G) = \varphi$ de manera directa. Uno puede mostrar entonces que $\varphi_l \leq \chi(G) \leq \varphi_u$ usando las acotaciones vistas antes. Si sucede que $\varphi_l = \varphi_u$ qué bonito hemos encontrado $\chi(G)$. Si este no es el caso de igual modo estamos restringiendo el espacio de soluciones posibles. Por ejemplo, si $3 \leq \chi(G) \leq 4$, tenemos que $\chi(G) = 3$ o $\chi(G) = 4$, y a veces es fácil ver (y demostrar) cuál de los casos es correcto.

1.9 Limitaciones de Greedy

Ejemplo de Greedy mal. Sea $G = (V, E)$ un grafo tal que $n = 2i$ es par y

$$E = \{xy : x \text{ impar}, y \text{ par}, y \neq x + 1\}$$

El grafo es claramente bipartito (un par nunca se conecta con un par, un impar nunca con un impar) y tiene un coloreo de 2 colores. Sin embargo, al correr Greedy sobre este grafo, obtenemos un coloreo sub-óptimo.

Se deja al lector verificar los coloreos sub-óptimos que resultan si $n = 2, 4, 6$. Se presenta la siguiente hipótesis inductiva haciendo inducción sobre i:

> *Hipótesis inductiva.* Para todo $j \leq k$, Greedy colorea el grafo con $c(2j-1) = c(2j) = j$ colores.
>
> *Caso inductivo.* Si $j \leq k$ por HI tenemos $c(2j - 1) = c(2j) = j$. Si $j = k + 1$, tenemos que $2(k + 1) - 1$ es impar y forma lados con todos los pares anteriores. Luego greedy no puede asignarle el color de ningún par anterior. Es decir, no puede asignar ningún color entre 1 y k. Luego greedy lo colorea con $k + 1$. El mismo razonamiento muestra que el vértice $2(k + 1)$ se colorea con $i + 1$.
>
> Por lo tanto, greedy colorea G con $\frac{n}{2}$ colores.

1.10 Hill clmbing

Hill climbing es un algoritmo de optimización matemática. Dada una función $f : \mathbb{R}^n \to \mathbb{R}$ que se quiere maximizar, el algoritmo toma un punto arbitrario $\vec{x} \in \mathcal{D}_f$ y evalúa $f(\vec{x})$. Luego se hace $\Delta\vec{x} = (x_1, \ldots, x_i + \epsilon, \ldots, x_n)$ con $\epsilon \in R$ e i arbitario y se vuelve a testear. Si la función incrementa, se hace $\vec{x} = \Delta\vec{x}$: si no, se hace la modificación inversa. Este proceso se repite iterativamente hasta que un criterio es satisfecho.

El principal problema del algoritmo de Hill climbing es que encuentra solo óptimos locales. Para lidiar con este problema, se usa *simulated annealing*. Si $\Delta\vec{x}$ va mejor se lo acepta; pero si $\Delta\vec{x}$ va peor se otorga una cierta probabilidad de aceptarlo de todos modos. La probabilidad de aceptar va reduciendo con el tiempo. La idea es permitir la salida de optimos locales con el tiempo.

En coloreo de grafos, Hill climbing toma la siguiente forma. Se prueba un orden de V arbitrario y se lo colorea con Greedy. Luego hacer una pequeña permutación y se prueba de nuevo, etc.

Existe un modo de asegurarnos que la mutación $\Delta\vec{x}$ de $\vec{x}$ nunca empeore (puede permanecer igual). El problema es que para hacer esto, reducimos el espacio de búsqueda; es decir, las permutaciones que exploraremos (las que no pueden empeorar el rendimiento) son muy pocas, y puede ser que ninguna mejore el rendimiento.

Una forma de lidiar con esto es elegir varios órdenes iniciales al azar y aplicar esta idea a cada una de ellas.

Theorem 7 (VIT) *Sea $G = (V, E)$ un grafo con un coloreo propio c con r colores $c_1, \ldots, c_r$. Sea $V_{c_i} := \{x \in V : c(x) = c_i\}$. Sea P una permutación de $c_1, \ldots, c_r$. Es claro que $P : \{c_1, \ldots, c_r\} \mapsto \{c_1, \ldots, c_r\}$ es una biyección.*

Ordenemos los vértices poniendo primero los vértices de $V_{P(c_1)}$; luego los de $V_{P(c_2)}$, etc. Es decir, ordenamos los vértices por bloques de color. Entonces Greedy con ese orden usa a lo sumo r colores.

Proof. Hacemos inducción sobre r.

Caso base $i = 1$. Los vértices de $V_{P(c_i)}$ tienen el mismo color $P(c_i)$. Porque c es un coloreo propio, y los vértices en cada $V_{P(c_i)}$ no pueden formar lados entre sí. Pues todos mis primeros vértices (los de $V_{P(c_1)}$) no tienen lados entre sí, Greedy los colorea a todos con el color 1 y usa un color. ■

$HI(i) \Rightarrow HI(i+1)$: Sea $x \in V_{P(c_1)} \cup \ldots \cup V_{P(c_{i+1})}$. Si x está en alguno de los primeros i bloques, Greedy lo colorea con uno de entre i colores por hipótesis inductiva.

Si $x \in V_{P(c_{i+1})}$, el caso en que Greedy lo colorea con un color menor o igual a $i + 1$ no viola lo que queremos demostrar. Pero asumamos que lo colorea con el color $i + 2$. Entonces ningún color menor estaba disponible; entonces x es vecino de algún z de color $i + 1$. Pero todos los vértices de $V_{P(c_i)}$ están coloreados el color c_i. Luego $z \in V_{P(c_{i+1})}$. Pero si tanto $x, z \in V_{P(c_{i+1})}$ entonces $c(x) = c(z) = P(c_{i+1})$, lo cual implica que c no es propio. $(\perp)$

2 Prácticos

2.1 Práctico 1

Problem 1 *Dar el algoritmo más rápido posible que resuelva el problema siguiente:*

Dado un input (T, n, m) con T un árbol, n el número de vértices, m el número de lados, dar $\chi(G)$.

El número cromático de un árbol siempre es dos. Se puede probar por inducción. El caso $n = 2$ es trivial. Asumamos vale para $k \in \mathbb{N}$ arbitrario. Sea $T = (V, E)$ un árbol con $k + 1$ vértices (y por lo tanto k lados). Sea $x \in V$ un vértice tal que $d(x) = 1$ y sea y el único $y \in V$ tal que $\{x, y\} \in E$. (Si lo desea, demuestre que si T es un árbol entonces necesariamente existe un vértice de grado 1; es trivial por def de árbol.)

Definamos $T' \subseteq T$ el sub-árbol cuyos vertices son $V - \{x\}$ y cuyos lados son $E - \{x, y\}$. Pues T' tiene k vértices, $\chi(T') = 2$. Ahora bien, aplicando cualquier coloreo propio de dos colores en $\{0, 1\}$ a los vértices de T', si hacemos $c(x) = 1 - c(y)$ obtenemos un coloreo propio de T.

$\therefore \chi(T) = 2$ para todo árbol T.

Por lo tanto, el algoritmo más rápido que devuelve el número cromático de un árbol es:

Algoritmo.

(1) Output 2.

Problem 2 (Ejercicio 6)

G contiene un C_{2r+1} y por lo tanto $\chi(G) \geq 3$. Es fácil dar un coloreo de cuatro colores para G_{2r+1}; utilice los tres colores necesarios para el C_{2r+1} contenido en $x_0, \ldots, x_{2r}$ y construya el resto de los valores desde allí; se llega por necesidad a cuatro colores. Entonces $\chi(G) \leq 4$. Mostraremos que $\chi(G) \neq 3$.

Asuma que existe un coloreo propio de tres colores. El C_{2r+1} contenido en $x_0, \ldots, x_{2r}$ debe colorearse con tres colores; digamos, con 0 en los índices pares, 1 en los impares, y 2 en el caso x_{2r}.

De esto se sigue que $c(y_i) = 1, c(z_i) = 2$ donde i es par (podríamos invertir esta asignación pero da lo mismo); $c(y_i) = 0, c(z_i) = 2$ si i impar; excepto en el caso $c(y_{2r}) = 0, c(z_{2r}) = 1$.

Habiendo coloreado cada triángulo x_i, y_i, z_i, queda un solo color posible para cada w_i; a saber, $c(w_i) = 0$ si i par; $c(w_i) = 1$ si i impar, y el caso especial $c(w_{2r}) = 2$.

Cada w_i se conecta con p. Pero los w_i ya tienen tres colores en total. (Y observe que para todo r, G_r siempre tendrá al menos tres w_i) Entonces no hay forma de dar un coloreo para p que sea propio.

Pues $\chi(G) \leq 4$ y $\chi(G) \neq 3$ tenemos que $\chi(G) = 4$.

Problem 3 *Dado n natural, sea G_n el grafo con vértices $1, 2, \ldots, n$ y cuyos lados son $\{i, j\}$ tales que $(i, j) = 1$ (coprimos). Calcular $\chi(G_{100})$.*

Es fácil notar que 1 y los z números primos en $2, 3, \ldots, n$ forman un K_{n+1} y por lo tanto $\chi(G_n) \geq z + 1$. Damos un coloreo propio con $z + 1$ colores; a saber,

$$c(i) = \begin{cases} 1 & i = 1 \\ \min_p \{p \text{ primo y } p \mid i\} & i > 1 \end{cases}$$

Que es propio se sigue de que $c(i) = c(j) \Rightarrow \gcd(i, j) = 1$ lo cual implica que no comparten ningún primo en su factorización; y en particular, no comparten el mínimo. Todo número no primo se expresa con el color del menor primo que está en su descomposición; es decir, no utiliza un color propiamente nuevo. Luego los únicos colores usados son los que identifican a los z primos y al 1. Es decir, es un coloreo propio de $z + 1$ colores.

Conclusión. $\chi(100) = 25 + 1 = 26$ (Hay 25 primos entre 1 y 100).