GrafosGreedy

Daniel Penazzi

17 de marzo de 2021

Tabla de Contenidos

Repaso de Definiciíon de Grafos

- Uds. vieron grafos en Discreta I
- Repasaremos algunos temas para estar seguros que los tenemos frescos.
- y para coordinar la notación.
- Recordemos que un grafo es un par ordenado $G = (V, E)$ donde
  - V es un conjunto cualquiera.
    - En esta materia siempre supondremos V finito.
  - E es un subconjunto del conjunto de subconjuntos de 2 elementos de V.
    - es decir $E \subseteq \{A \subseteq V : |A| = 2\}$

Notaciones

- Los elementos de V se llaman vértices o nodos. Usaremos preferentemente el primer nombre.
- Los elementos de E se llaman lados o aristas. Usaremos preferentemente el primer nombre.
- La cantidad de elementos de V, salvo que digamos otra cosa, se denotará por default como n.
- La cantidad de elementos de E, salvo que digamos otra cosa, se denotará por default como m.
- Un elemento $\{x, y\} \in E$ será abreviado como xy.
- x e y se llamarán los extremos del lado xy.

Lados

- Como xy denota el conjunto $\{x, y\}$ entonces es claro que $xy = yx$.
- Hay grafos en donde vamos a querer que el orden de los elementos de un lado importe, esos serán “grafos dirigidos”.
- La diferencia es que en vez de ser $E \subseteq \{A \subseteq V : |A| = 2\}$ tendremos $E \subseteq A \times A$.
- Pero ese tipo de grafos los veremos mas adelante, por ahora estaremos tomando grafos “no dirigidos”

Ejemplo de un grafo

- $G = (V, E)$ con:
  - $V = \{A, B, C, D, E\}$ (por lo tanto, $n = 5$)
  - $E = \{\{A, B\}, \{A, C\}, \{A, E\}, \{C, E\}\}$ (por lo tanto, $m = 4$)
  - o bien, en la notación que usaremos usualmente que hemos aclarado antes, $E = \{AB, AC, AE, CE\}$.
- No hay que confundir un grafo con la representación gráfica de un grafo.
- Podemos representar G dibujando 5 puntos, representando cada uno de los vértices, y dibujando lineas, ya sea rectas o curvas, uniendo dos vértices si forman un lado.
- La representación gráfica puede ser útil para grafos con pocos vértices o lados, pero a medida que el grafo se vuelve mas grande se vuelve casi inútil.

Subgrafos

- Dado un grafo $G = (V, E)$, un subgrafo de G es un grafo $H = (W, F)$ tal que $W \subseteq V$ y $F \subseteq E$.
- Observemos que pedimos que H sea en si mismo un grafo. No cualquier par (W, F) con $W \subseteq V$ y $F \subseteq E$ será un subgrafo porque necesitamos que si un lado está, los dos extremos del lado estén.
- En el ejemplo anterior $(\{A, B, C\}, \{AC\})$ es un subgrafo de G.
- $(\{A, B, C, D, E\}, \{AB, CE\})$ es otro subgrafo de G

Vecinos de un vértice

- Dado $x \in V$, los vértices que forman un lado con x se llaman los vécinos de x.
- El conjunto de vécinos se llama el "vecindario" y se denota por $\Gamma(x)$.
- Es decir $\Gamma(x) = \{y \in V : xy \in E\}$
- En el ejemplo anterior:
  - $\Gamma(A) = \{B, C, E\}$.
  - $\Gamma(B) = \{A\}$.
  - $\Gamma(C) = \{A, E\}$.
  - $\Gamma(D) = \emptyset$.
  - $\Gamma(E) = \{A, C\}$.

Grado de un vértice

- La cardinalidad de $\Gamma(x)$ se llama el grado de x, y la denotaremos por $d(x)$ (o $d_G(x)$ si queremos recalcar G, pej, si tenemos varios grafos con los mismos vértices).
- WARNING: en algunos libros se denota usando la letra griega delta: $\delta(x)$ pero δ nosotros la usaremos para otra cosa. (ver página siguiente).
- En el ejemplo anterior, tenemos:
  - $d(A) = 3$.
  - $d(B) = 1$.
  - $d(C) = 2$.
  - $d(D) = 0$.
  - $d(E) = 2$.

δ y Δ

- El menor de todos los grados de un grafo lo denotaremos por δ y al mayor de todos los grados por Δ.
- Es decir:
  - $\delta = \text{Min}\{d(x) : x \in V\}$
  - $\Delta = \text{Max}\{d(x) : x \in V\}$
- En el ejemplo anterior, $\delta = 0$, $\Delta = 3$.
- Un grafo que tenga $\delta = \Delta$ (es decir, todos los grados iguales) se llamará un grafo regular.
- o Δ-regular si queremos especificar el grado común a todos los vértices.
- Puesto que en el ejemplo $\delta = 0 \neq 3 = \Delta$, el ejemplo no es un grafo regular.

Cíclicos y completos

1. El grafo cíclico en n vértices, ($n > 3$) denotado por C_n, es el grafo:
   - Con vértices $\{1, 2, ..., n\}$.
   - Con lados $\{12, 23, ..., (n-1)n, n1\}$.
   - (mas generalmente podria ser cualquier conjunto con n elementos $\{x_1, ..., x_n\}$ y lados $\{x_1x_2, x_2x_3, ..., x_{n-1}x_n, x_nx_1\}$).
2. El grafo completo en n vértices, denotado por K_n, es el grafo:
   - Con vértices $\{1, 2, ..., n\}$.
   - Con lados $\{ij : i, j \in \{1, 2, ..., n\}, i < j\}$.
   - (mas generalmente podria ser cualquier conjunto con n elementos $\{x_1, ..., x_n\}$ y lados $\{x_ix_j : i, j \in \{1, 2, ..., n\}, i < j\}$)

Mas sobre ciclicos y completos.

- C_n y K_n tienen ambos n vértices, (aunque C_1 y C_2 no están definidos y K_1 y K_2 si) pero C_n tiene n lados mientras que K_n tiene $\binom{n}{2} = \frac{n(n-1)}{2}$ lados.
- Observar que $C_3 = K_3$ y su representación gráfica es un triángulo.
- C_n se llaman cíclicos porque su representación gráfica es un ciclo de n puntos.
- Pej, como dijimos, C_3 es un triangulo, C_4 un cuadrado, C_5 un pentágono, etc.
- $d_{C_n}(x) = 2$ para todo vértice de C_n, mientras que $d_{K_n}(x) = n - 1$ para todo vértice de K_n.
- Por lo tanto ambos son grafos regulares.
- (C_n es 2-regular y K_n es $(n-1)$-regular).

Componentes conexas

- Un camino entre 2 vértices x, y es una sucesión de vértices $x_1, ..., x_r$ tales que:
  1. $x_1 = x$
  2. $x_r = y$.
  3. $x_i x_{i+1} \in E \ \forall i \in \{1, 2, ..., r-1\}$.
- Es trivial ver que la relación:
  - "$x \sim y$ sii existe un camino entre x e y"
- es una relación de equivalencia.
- Por lo tanto el grafo G se parte en clases de equivalencia de esa relación de equivalencia.
- Esas partes se llaman las componentes conexas de G.

Grafos conexos

- Un grafo se dice conexo si tiene una sola componente conexa.
- Por ejemplo, C_n y K_n son conexos.
- Pero el ejemplo de grafo que habiamos dado al principio no es conexo.
- Tiene dos componentes conexas: $\{A, B, C, E\}$ y $\{D\}$.
- Recordemos que un arbol es un grafo conexo sin ciclos. (es decir, que no tiene como subgrafo a un C_k).

Componentes conexas.

- ¿Cómo determinar las componentes conexas?
- En realidad no estoy seguro si esto lo vieron en Discreta I, pero seguro lo vieron en Algoritmos II.
- De hecho, vieron al menos dos algoritmos que hacen esto.

Determinación de las componentes conexas

- Son DFS y BFS.
- De hecho, hay un poco de ambivalencia en que queremos decir por “DFS”y “BFS”
- El algoritmo básico de DFS o BFS lo que hace es, dado un vértice x, encontrar todos los vértices de la componente conexa de x.
- Asi que si llamamos DFS o BFS a eso, no encuentra todas las componentes conexas.
- Pero ese algoritmo seria mas preciso llamarlo DFS(x) o BFS(x).
- Y el algoritmo DFS o BFS seria: (concretemos para BFS, pej):

Determinación de las componentes conexas

(abajo en vez de BFS puede usarse DFS)

1. Tomar $W = \emptyset$, $i = 1$.
2. Tomar un vértice cualquiera x de V.
3. Correr BFS(x).
4. LLamarle C_i a la componente conexa que encuentra BFS(x).
5. Hacer $W = W \cup$ (vértices de C_i).
6. Si $W = V$, **return** $C_1, C_2, ..., C_i$.
7. Si no, hacer $i = i + 1$, tomar un vértice $x \notin W$ y repetir [3].

DFS y BFS

- Uds ya vieron DFS y BFS en Algoritmos II, asi que no los veremos aca asi que si no los recuerdan bien vuelvan a verlos.
- En especial BFS va a ser usado mas adelante en forma crucial en un algoritmo muy importante.
- Sólo daremos un breve repaso de ambos.
- En ambos casos, a partir de un vértice raiz, los algoritmos van buscando nuevos vértices, buscando vecinos de vértices que ya han sido agregados.
- Los algoritmos pueden ser implementados simplemente buscando los vértices, o ademas agregando los lados entre un vértice que ya estaba agregado y los vértices que ese vértice agregó.

DFS y BFS

- En ambos casos, si se hace lo segundo, las componentes conexas que se obtengan serán árboles, por la forma que tienen ambos algoritmos.
- Esto es porque en ambos casos no se agrega un vértice que ya estaba agregado, asi que no se generan ciclos.
- La diferencia entre ambos es cómo se buscan vecinos, y qué tipo de estructura de datos usan.
- DFS agrega de a un vécino por vez y usa una pila.
- BFS agrega todos los vecinos juntos y usa una cola.

BFS(x):

- Crear una cola con x como único elemento.
- Tomar $C = \{x\}$.
- WHILE (la cola no sea vacia)
  - Tomar p=el primer elemento de la cola.
  - Borrar p de la cola.
  - IF existen vértices de $\Gamma(p)$ que no esten en C:
    - Agregar todos los elementos de $\Gamma(p)$ que no estén en C a la cola y a C.
  - ENDWHILE
- **return** C.

DFS(x):

- Crear una pila con x como único elemento.
- Tomar $C = \{x\}$.
- WHILE (la pila no sea vacia)
  - Tomar p=el primer elemento de la pila.
  - IF existe algún vértice de $\Gamma(p)$ que no esté en C:
    - Tomar un $q \in \Gamma(p) - C$.
    - Hacer $C = C \cup \{q\}$.
    - Agregar q a la pila.
  - ELSE:
    - Borrar p de la pila.
  - ENDWHILE
- **return** C.

Complejidad

- Deberian haber visto en Algoritmos II que la complejidad tanto de DFS como de BFS es $O(m)$.
- (por supuesto, en realidad depende de como se implementen. Hay formas malas de implementarlos que harian que la complejidad fuese peor).
- Una variación obvia de DFS o BFS hace que en vez de retornar las componentes conexas, se retorne simplemente el número de componentes conexas, si es de interes.
- En el proyecto que van a tener que entregar van a tener que implementar al menos uno de estos algoritmos.

Coloreos propios

- Un coloreo (de los vértices) es una función cualquiera $c : V \to S$ donde S es un conjunto finito.
- Un coloreo es propio si $xy \in E \Rightarrow c(x) \neq c(y)$ (extremos con distinto color)
- Si la cardinalidad de S es k diremos que el coloreo tiene k colores. En general usaremos $S = \{0, 1, ..., k-1\}$ para denotar los colores.
- Un grafo que tiene un coloreo propio con k colores se dice k-coloreable.
- El número cromático es

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

Calculando $\chi(G)$

- Si uno dice que $\chi(G) = k$, por la definición misma de este número, hay que hacer dos cosas para probarlo:
  1. Dar un coloreo propio de G con k colores. (y obviamente probar que es propio).
     - Esto prueba la parte del "$\exists$ un coloreo propio con k colores de G"
  2. Probar que no existe ningún coloreo propio con $k-1$ colores de G.
     - Esto prueba que k es el mínimo.
- Es un error inexplicable (porque repetimos una y otra vez que no lo hagan, pero lo hacen igual) pero común que muchos alumnos cuando se les pide probar que $\chi(G)$ es igual a pej 5, lo que hacen es dar un coloreo propio con 5 colores y nada mas.
- Es decir, prueban [1] arriba pero no [2].
- Este es un error serio con alto descuento de puntos.

Calculando $\chi(G)$

- En general probar [1] es la parte mas fácil de un ejercicio, aunque en algunos casos puede ser la mas dificil si no pueden encontrar un coloreo.
- Probar [2], salvo en algunos casos especiales, puede ser la parte mas complicada.
- Una ayuda útil para probar [2] es la siguiente observación obvia:
  - Si H es un subgrafo de G, entonces $\chi(H) \leq \chi(G)$.
    - Prueba: pues si necesito r colores para colorear los vértices de H, no puedo colorear a todos los vértices de G con menos de r colores, porque tendria un coloreo con menos de r colores de los vértices de H al restringir el coloreo de G a los vértices de H.
- Entonces si encontramos un subgrafo H de G para el cual sepamos que $\chi(H) = k$ habremos probado [2].

Calculando $\chi(G)$

- Lamentablemente en muchos casos no es posible o no es fácil encontrar un H adecuado con esa propiedad.
- Entonces hay que hacer una prueba por contradicción: se asume que existe un coloreo propio con $k - 1$ colores y deduciendo cosas, se llega a un absurdo.
- Hay 2 problemas aca:
  1. Llegar al absurdo puede ser bastante dificil, teniendo que contemplar varios casos, pej.
  2. Para poder hacer la prueba por contradicción, hay que asumir que existe un coloreo propio con $k - 1$ colores.
     - Eso significa que uds. NO TIENEN CONTROL sobre ese coloreo.
     - Sólo saben que hay uno, y deben deducir cosas sobre ese coloreo a partir de la estructura del grafo.
     - Un error típico es que los alumnos CONSTRUYEN un coloreo con $k - 1$ colores y prueba que ese coloreo que construyeron no es propio.
     - Esto obviamente no prueba nada mas que ese coloreo no es adecuado, pero no demuestra que no pueda existir otro coloreo.

$\chi(G)$ para algunos grafos

- En general, dado que para cualquier grafo G podemos darle un color distinto a todos los vértices, tenemos la desigualdad $\chi(G) \leq n$.
- Obviamente $\chi(K_n) = n$ pues al estar todos los vértices unidos con todos los otros vértices, entonces todos los vértices deben tener colores distintos.
- Entonces si quieren probar que $r \leq \chi(G)$ basta con ver que existe un K_r subgrafo de G.
- Por ejemplo, en el ejemplo dado al principio, dado que el subgrafo $(\{A, C, E\}, \{AC, AE, CE\})$ es un K_3, concluimos que $3 \leq \chi(G)$.
- Como es trivial dar un coloreo propio con 3 colores de G, concluimos $\chi(G) = 3$.
- Sin embargo, la vuelta NO VALE: puede ocurrir que $r \leq \chi(G)$ pero que no exista ningún subgrafo K_r de G. (luego veremos un ejemplo).

$\chi(G)$ para algunos grafos

- $\chi(G) = 1$ si y solo si $E = \emptyset$ asi que para cualquier grafo que tenga al menos un lado, $\chi(G) \geq 2$.
- $\chi(C_{2r}) = 2$ pues podemos colorear $c(i) = (i \mod 2)$ (es decir $c(i) = 0$ si i es par y $c(i) = 1$ si i es impar) y eso es un coloreo propio porque vértices consecutivos en el ciclo tendrán colores distintos.
- Pero no podemos hacer lo mismo con $\chi(C_{2r+1})$ pues tendriamos que $2r + 1$ y 1 tendrían color 1, absurdo pues forman lado.
- Podemos colorear: $c(i) = (i \mod 2)$ si $i < 2r + 1$ y $c(2r + 1) = 2$.
- Ese es un coloreo propio pues $2r + 1$ tiene color distinto del resto de los vértices asi que no hay problema con el, y los demas vértices consecutivos tienen colores distintos.
- Esto demuestra que $\chi(C_{2r+1}) \leq 3$. Pero todavia no probamos que $3 \leq \chi(C_{2r+1})$ pues sólo hemos probado que un coloreo específico que dimos con 2 colores no es propio. Podria haber otro.

$\chi(C_{2r+1}) = 3$

- Para ver que $3 \leq \chi(C_{2r+1})$ supongamos que no es cierto. Esto implica que existe un coloreo propio c con 2 colores.
- Sea $A = c(1)$.
- Como $12 \in E$, entonces $c(2) \neq c(1)$. Sea $B = c(2)$. Entonces acabamos de ver que $B \neq A$ y como estamos suponiendo que es un coloreo con 2 colores, entonces A y B son esos 2 colores.
- Como $23 \in E$, entonces $c(3) \neq c(2)$. Como $c(2) = B$ y sólo hay 2 colores, concluimos que necesariamente $c(3) = A$.
- Como $34 \in E$, entonces $c(4) \neq c(3)$. Como $c(3) = A$ y sólo hay 2 colores, concluimos que necesariamente $c(4) = B$.
- Continuando de esta forma y probando por inducción, concluimos que $c(i) = A$ si i es impar (y $c(i) = B$ si i es par) lo cual es un absurdo pues 1 y $2r + 1$ forman lado.

- Entonces hemos probado que los ciclos impares (i.e., con un número impar de vértices) tienen número cromático igual a 3.
- En particular, $\chi(C_5) = 3$.
- Pero C_5 no tiene como subgrafo ningún K_3, asi que esto es un ejemplo de que se puede tener $\chi(G) \geq r$ sin tener un K_r como subgrafo.
- Que los ciclos impares tengan número cromático igual a 3 significa que cualquier grafo que tenga como subgrafo a un ciclo impar debe tener número cromático mayor o igual que 3.
- En algunos ejercicios esto les permitira calcular rapidamente $\chi(G)$: sólo tienen que dar un coloreo propio con 3 colores y encontrar algún ciclo impar.

Algoritmo de fuerza bruta

- Un algoritmo para encontrar $\chi(G)$ es simplemente tomar todos los coloreos posibles con los colores $\{0, 1, ..., n-1\}$ y calcular cuales de esos coloreo son propios, y ver de entre esos quien tiene la menor cantidad de colores.
- Este algoritmo calcula $\chi(G)$ pero:
  1. Hay n^n posibles coloreos.
  2. Chequear que un coloreo es propio es $O(m)$.
- por lo tanto el algoritmo tiene complejidad $O(n^n m)$ asi que no es útil salvo para n muy chicos.
- Ahora veremos un algoritmo que no calcula $\chi(G)$ pero al menos da un coloreo propio en tiempo polinomial.

Algoritmo Greedy

- El algoritmo Greedy requiere como input no sólo un grafo G sino un orden de los vértices.
- Algunas implementaciones en vez de requerir un orden simplemente usan algún orden cualquiera, pero para extraer el mayor beneficio posible de Greedy conviene poder llamarlo varias veces cambiando el orden.
- De hecho, luego probaremos que si bien Greedy no necesariamente colorea G con $\chi(G)$ colores, existe un orden de los vértices tal que Greedy, con ese orden, colorea G con $\chi(G)$ colores, asi que el orden es importante.
- Esto no nos da un algoritmo polinomial para calcular $\chi(G)$ porque hay $n!$ ordenes posibles.

Idea de Greedy

- La idea de Greedy consiste de dos partes:
  1. Ir coloreando los vértices de G uno por uno, en el orden dado, manteniendo siempre el invariante que el coloreo parcial que se va obteniendo es propio.
  2. Darle a cada vértice al momento de colorearlo el menor color posible que se le pueda dar manteniendo el invariante de que el coloreo es propio.

1 nos asegura que cuando Greedy termina, el coloreo que da es propio.

2 es lo que le da el nombre al algoritmo: usar siempre el menor color posible.

- Como muchas cosas en la vida, hacer algo que optimiza el presente no necesariamente optimiza el futuro, y luego veremos ejemplos de que Greedy no siempre obtiene $\chi(G)$.

Greedy

- Input: Grafo G y orden de los vértices $x_1, x_2, ..., x_n$.
- $c(x_1) = 0$
- Para $i > 1$, asumiendo que los vértices $x_1, x_2, \ldots, x_{i-1}$ ya han sido coloreados, colorear x_i con:
  - $c(x_i) = \min\{k \geq 0 : k \notin c\left(\{x_1, \ldots, x_{i-1}\} \cap \Gamma(x_i)\right)\}$
- Arriba estamos usando la notación usual de $c(A) = \{c(a) : a \in A\}$.
- Es decir, x_i recibe el menor color que sea distinto del color de todos los vecinos anteriores a x_i.

Greedy, ejemplo

- $G =$ el grafo del ejemplo del principio, con el orden alfabetico.
- $c(A) = 0$.
- Como $AB \in E$, B no puede tener el color de A, coloreamos B con el menor color posible distinto de 0: $c(B) = 1$.
- Como $AC \in E$, C no puede tener el color de A. Pero A es el único vécino de C en el conjunto $\{A, B\}$ asi que Greedy le da $c(C) = 1$.
- Como D no tiene vecinos, $c(D) = 0$.
- Como $\Gamma(E) = \{A, C\}$, entonces E no puede tener ni el color 0 ni el 1, Greedy le da $c(E) = 2$.
- En este caso Greedy coloreó con 3 colores, y como el grafo tiene un K_3 entonces $\chi(G) = 3$.

Greedy, ejemplo II

- Tomemos C_6 pero con el orden $1, 4, 2, 3, 5, 6$.
- $c(1) = 0$.
- 4 no es vecino de 1, por lo tanto, Greedy tambien le da el color 0.
- 2 es vecino de 1, Greedy hace $c(2) = 1$.
- 3 tiene vecinos a 2 (de color 1) y a 4 (de color 0) asi que Greedy le da el color 2.
- 5 tiene vecino anterior solamente a 4, asi que Greedy le da el color 1.
- 6 tiene vecinos a 1 (color 0) y a 5 (color 1) asi que Greedy hace $c(6) = 2$.
- Conclusión: Greedy colorea C_6, en ese orden, con 3 colores.
- Pero sabemos que $\chi(C_6) = 2$ asi que este es un ejemplo de que Greedy puede colorear con mas de $\chi(G)$ colores.

Complejidad de Greedy

- Para colorear el vertice x_i Greedy debe chequear $d(x_i)$ vecinos de x para ver sus colores.
- Por lo tanto la complejidad de Greedy es $O(d(x_1) + d(x_2) + \cdots + d(x_n))$.
- (en realidad el d_1 esta de mas porque colorear x_1 es $O(1)$ pero no importa).
- Por el lema del apretón de manos que vieron en Discreta I, la suma de todos los grados es igual a $2m$.
- Por lo tanto la complejidad de Greedy es $O(2m) = O(m)$, polinomial.
- (el lema del apreton de manos es facil de probar: al sumar sobre todos los grados, estamos contando cada lado xy dos veces: una vez cuando pasamos por x y otra cuando pasamos por y).