

Resumen filmina 1 MD2

Lautaro Bachmann

Contents

Grafos	2
grafo	2
Notaciones	2
Subgrafos	2
Vecinos de un vértice	3
Grado de un vértice	3
y	3
Cíclicos y completos	3
Componentes conexas	4
Grafos conexos	4
Determinación de las componentes conexas	5
DFS y BFS	5
BFS(x):	5
DFS(x):	6
Complejidad	6
Coloreos propios	6
número cromático	6
Calculando $\chi(G)$	7
ayuda útil para probar [2]	7
$\chi(G)$ para algunos grafos	7
Algoritmo de fuerza bruta	8
Algoritmo Greedy	8

Grafos

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 $E \subseteq \mathcal{P}(V)$:

Notaciones

elementos de V

elementos de E

cantidad de elementos de V ,

cantidad de elementos de E ,

Un elemento E

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

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 $\Gamma(x) = \{y \in V : xy\}$

Grado de un vértice

La cardinalidad de $\Gamma(x)$ se llama el **grado** de x , y la denotaremos por $d(x)$ (o $d_G(x)$)

WARNING:

en algunos libros se denota usando la letra griega delta: $\delta(x)$

y

El menor de todos los grados

de un grafo lo denotaremos por δ y al

mayor de todos los grados

por Δ .

$$\delta = \min_{x \in V} d(x) \quad \Delta = \max_{x \in V} d(x)$$

Un grafo que tenga $\delta = \Delta$ (es decir, todos los grados iguales) se llamará un

grafo regular.

o r -regular si queremos especificar el grado común a todos los vértices.

Cíclicos y completos

grafo cíclico

en n vértices, ($n > 3$) denotado por C_n , es el grafo:

..., y lados $x_1x_2, x_2x_3, \dots, x_{n-1}x_n, x_nx_1$, ...,

grafo completo

en n vértices, denotado por K_n , es el grafo:

..., y lados : $i, j \in \{1, 2, \dots, n\}, i < j$

C_n y K_n tienen ambos n vértices, pero C_n tiene n lados mientras que K_n tiene $\frac{n(n-1)}{2}$ lados.

C_n se llaman cíclicos porque su representación gráfica es un ciclo de n puntos.

$d_{C_n}(x) = 2$ 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**camino**

entre 2 vértices x, y es una sucesión de vértices x_1, \dots, x_r tales que:

$x_1 = x$

$x_r = y, x_i x_{i+1} \in E, i = 1, \dots, r-1$

“ x y y están conectados por un camino entre x y 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 .

componentes conexas**Grafos conexos**

Un grafo se dice conexo si tiene una sola componente conexa.

C_n y K_n son conexos.

arbol

es un grafo conexo sin ciclos.

Determinación de las componentes conexas

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 .

algoritmo

(abajo en vez de BFS puede usarse DFS)

Tomar $W = i = 1$.

Tomar un vértice cualquiera x de V .

Correr $BFS(x)$.

LLamarle C_i a la componente conexa que encuentra $BFS(x)$.

Hacer $W = (\text{vértices de } C_i)$.

Si $W = V$, return C_1, C_2, \dots, C_i .

Si no, hacer $i = i + 1$, tomar un vértice $x \in W$ y repetir [3].

DFS y BFS

breve repaso

a partir de un vértice raíz, los algoritmos van buscando nuevos vértices, buscando vecinos de vértices que ya han sido agregados. DFS agrega de a un vecino 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 $C = \text{WHILE}$ (la cola no sea vacía)

Tomar $p = \text{el primer elemento de la cola}$. Borrar p de la cola. IF existen vértices de $\Gamma(p)$ que no estén 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 = WHILE (la pila no sea vacía)

Tomar p=el primer elemento de la pila. IF existe algún vértice de $\Gamma(p)$ que no esté en C:

Tomar un q $\Gamma(p)$ C. Hacer C = C Agregar q a la pila.

ELSE:

Borrar p de la pila.

ENDWHILE

return C.

Complejidad

la complejidad tanto de DFS como de BFS es $O(m)$.

Coloreos propios

Un coloreo (de los vértices) es una función cualquiera $c : V \rightarrow S$ donde S es un conjunto finito.

Un coloreo es propio si $\forall xy \in E, c(x) \neq c(y)$ (extremos con distinto color)

Si la cardinalidad de S es k diremos que el coloreo tiene k colores. En general usaremos $S = \{1, \dots, k\}$ para denotar los colores.

Un grafo que tiene un coloreo propio con k colores se dice k-coloreable.

número cromático

$\chi(G) = k$: un coloreo propio con k colores de

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 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.

ayuda útil para probar [2]

Si H es un subgrafo de G , entonces $\chi(H) \leq \chi(G)$.

Entonces si encontramos un subgrafo H de G para el cual sepamos que $\chi(H) = k$ habremos probado [2].

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

1 Llegar al absurdo puede ser bastante difícil, 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.

$\chi(G)$ para algunos grafos

En general, dado que para cualquier grafo G podemos darle un color distinto a todos los vértices, tenemos la desigualdad $\chi(G) \leq n$. $\chi(K_n) = n$ si quieren probar que $\chi(G) \leq r$ basta con ver que existe un K_{r+1} subgrafo de G . $\chi(G) = 1$ si y solo si $E = \emptyset$ así que para cualquier grafo que tenga al menos un lado, $\chi(G) \geq 2$. $\chi(C_{2r}) = 2$ pues podemos colorear $c(i) = (i \bmod 2)$ con $\chi(C_{2r+1}) = 3$ pues tendríamos que $2r + 1$ y 1 tendrían color 1, absurdo pues forman lado. Podemos colorear: $c(i) = (i \bmod 2)$ si $i < 2r + 1$ y $c(2r + 1) = 3$.

los ciclos impares tienen número cromático igual a 3.

cualquier grafo que tenga como subgrafo a un ciclo impar debe tener número cromático mayor o igual que 3.

Algoritmo de fuerza bruta

simplemente tomar todos los coloreos posibles con los colores 1, ..., n y calcular cuales de esos coloreo son propios, y ver de entre esos quien tiene la menor cantidad de colores.

Este algoritmo calcula (G) pero:

Hay nn posibles coloreos. Chequear que un coloreo es propio es $O(m)$.

el algoritmo tiene complejidad $O(nm)$ asi que no es útil salvo para n muy chicos.

Algoritmo Greedy

El algoritmo Greedy requiere como input no sólo un grafo G sino un **orden** de los vértices.

para extraer el mayor beneficio posible de Greedy conviene poder llamarlo varias veces cambiando el orden.

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.

Greedy

Input: Grafo G y orden de los vértices x_1, x_2, \dots, x_n .

$c(x_1) = 0$ Para $i > 1$, asumiendo que los vértices x_1, x_2, \dots , ya han sido coloreados, colorear x_i con:

$c(x_i) = 0 : k \text{ c } \dots$,

estamos usando la notación usual de $c(A) = : a$

Es decir, x_i recibe el menor color que sea distinto del color de todos los vecinos anteriores a x_i .

Complejidad de Greedy

la complejidad de Greedy es $O(d(x_1) + d(x_2) + \dots + d(x_n))$.

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.