# resumen

Lautaro Bachmann

## Contents

## 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 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 V y F E.

Observemos que pedimos que H sea en si mismo un grafo. No cualquier par (W, F) con W V y F E será un subgrafo

### Vecinos de un vértice

Dado x 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 Γ(x).

Es decir Γ(x) = V : xy

### Grado de un vértice

La cardinalidad de Γ(x) se llama el **grado** de x, y la denotaremos por d(x) (o dG(x)

**WARNING:**
en algunos libros se denota usando la letra griega delta: (x)

**y**

**El menor de todos los grados**
de un grafo lo denotaremos por y al

**mayor de todos los grados**
por .

= : x = : x

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

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

### Cíclicos y completos

**grafo cíclico**
en n vértices, (n > 3) denotado por Cn, es el grafo:

…, y lados x2x3, …,

**grafo completo**
en n vértices, denotado por Kn, es el grafo:

..., y lados : i, j 2, ..., i <

Cn y Kn tienen ambos n vértices, pero Cn tiene n lados mientras que Kn tiene

= lados. 2

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

dCn(x) = 2 para todo vértice de Cn, mientras que dKn(x) = n 1 para todo vértice de Kn.

Por lo tanto ambos son grafos regulares.

es 2-regular y Kn es (n 1)-regular).

## Componentes conexas

**camino**
entre 2 vértices x, y es una sucesión de vértices x1, ..., xr tales que:

x1 = x

xr = y. xixi+1 E 2, ..., r

"x 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.

**componentes conexas**

## Grafos conexos

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

Cn y Kn 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 Ci a la componente conexa que encuentra BFS(x).

Hacer W = (vértices de Ci).

Si W = V, return C1, C2, ..., Ci.

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

## DFS y BFS

#### breve repaso

a partir de un vértice raiz, los algoritmos van buscando nuevos vértices, buscando vecinos de vértices que ya han sido agregados. 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 = 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 = 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 $\Gamma(p)$ 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

### Calculando (G)

### Algoritmo de fuerza bruta

### Algoritmo Greedy

### Idea de Greedy

### Greedy

### Complejidad de Greedy