Grafos II: Aplicaciones y características

Matemáticas Discretas
(TC1003)

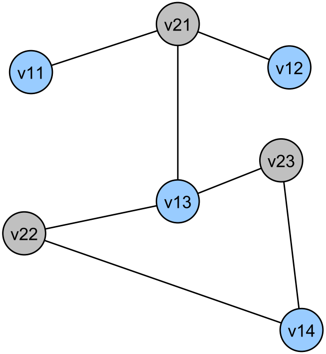
M.C. Xavier Sánchez Díaz
mail@tec.mx

Outline

Grafos bipartito

Grafos especiales

Un grafo $G = (V, E)$ se dice que es bipartito si $V = V_1 \cup V_2$ tal que no existen ejes que **interconecten** V_1 o V_2

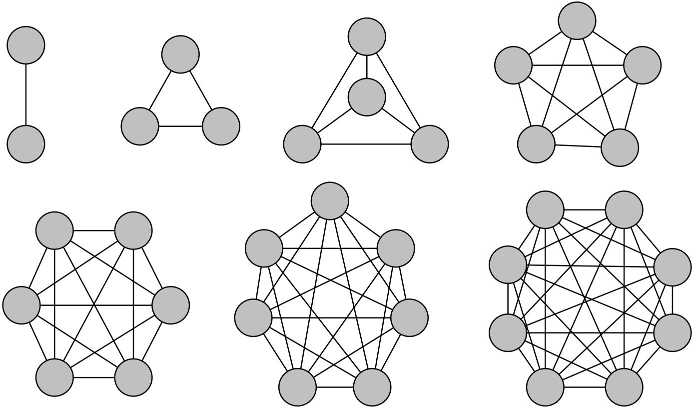


En otras palabras, si todos los ejes que salen de $v_i \in V_1$ llegan a $v_j \in V_2$ y viceversa.

Grafos K

Grafos Especiales

Un grafo **completo** K_n es un grafo con n vértices y con todos los ejes posibles, que también es $n-1$-regular.

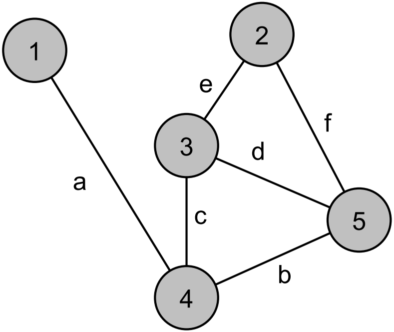


Matriz de adyacencia

Representaciones con matrices

Una matriz de adyacencia $n \times m$ puede representar en su celda $A_{i,j}$ si existe un eje entre el vértice i y el vértice j:

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{bmatrix}$$

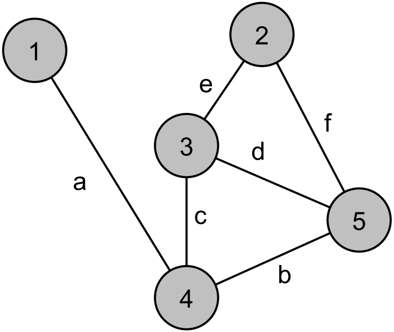


Matriz de incidencia

Representaciones con matrices

Una matriz de incidencia $n \times m$ puede representar en sus celdas $T_{i,k} = T_{j,k}$ si el eje k tiene como extremos a los vértices i y j:

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}$$

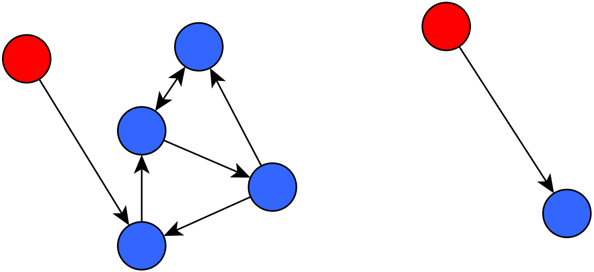


Componentes fuertemente conectados

Operaciones con grafos

En un **grafo direccionado**, dos vértices u y v están fuertemente conectados si existe una **caminata** de u a v y de v a u.

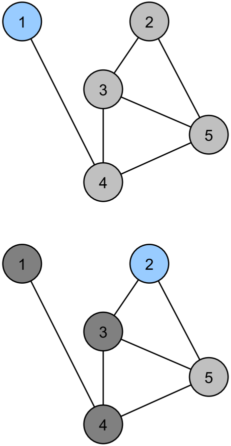
Podemos agrupar los **componentes fuertemente conectados** para reducir el grafo.



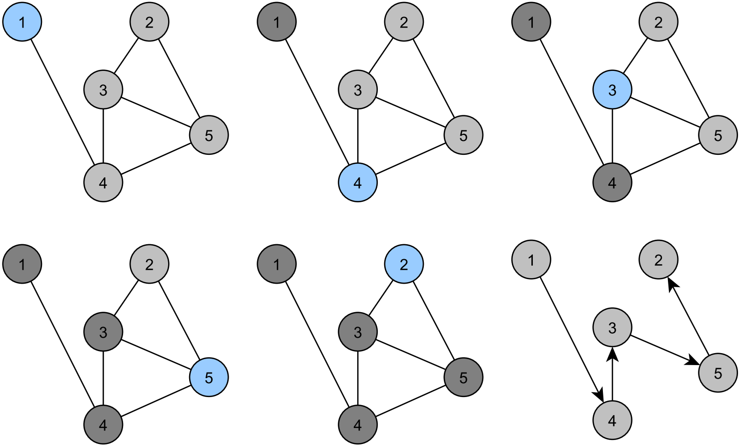
Búsqueda

Operaciones con grafos

Podemos hacer una búsqueda en un grafo para marcar los vértices encontrados y generar una secuencia. Empezando en el 1, podemos buscar por profundidad:

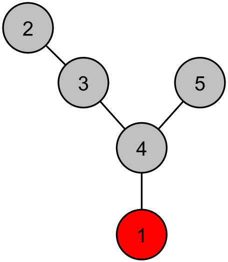


O bien podemos buscar por anchura:



Árboles

El **orden** de los nodos de un grafo dan pie a una jerarquía, lo cual es usualmente representado con un grafo acíclico que conocemos como árbol.

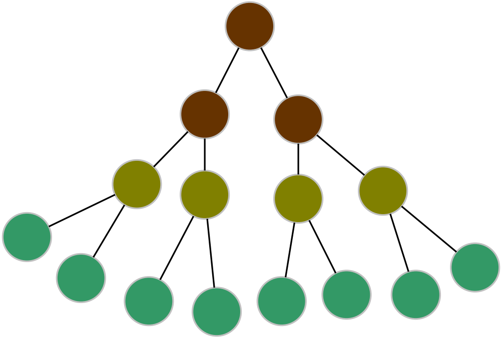


Un **árbol** es una estructura de datos *ordenada*, donde el nodo raíz es padre de algunos otros nodos hijos. Los nodos *finales*, los que no tienen descendencia, se les conoce como nodos hoja, y suelen representarse con la raíz hasta arriba...

Definición Recursiva

Árboles

. . . así. Un árbol puede ser definido de manera **recursiva**, considerando que tiene la misma estructura replicada múltiples veces.



En un árbol binario, cada vértice padre tiene dos nodos hijos—uno izquierdo y uno derecho—que a su vez tienen cada uno dos nodos hijos. . .

Aplicaciones de alta complejidad

Como ya vimos, muchas situaciones problema pueden ser representadas con grafos. Sin embargo, existen algunos problemas *clásicos* que suelen estudiarse (y que no son parte del índice analítico pero es bueno que conozcan).

- *Max-flow*
- *Min-cut*
- *Max-flow Min-cut*
- *Minimum spanning tree*
- *Eulerian tour*
- *Chinese Postman*

- *Hamiltonian cycle*
- *Traveling Salesman*
- *Graph-coloring*
- *Constraint Satisfaction*
- *K-satisfiability*

Todos los de la derecha son de la clase $\mathcal{NP}$*-complete*[1]. Si alguien encuentra cómo resolverlos de manera óptima, por favor envíeme un correo.

[1] Véase `https://en.wikipedia.org/wiki/List_of_NP-complete_problems`

Un problema de satisfacción de restricciones se define como una tripleta $P = (X, D, C)$ donde

- X es un conjunto de variables,
- D es un conjunto de dominios de dichas variables (los valores que pueden tomar), y
- C es un conjunto de restricciones

en donde la solución es verdadera o falsa dependiendo de la existencia de un mapeo $f\colon X_i \to D_i, \forall i \in X, i \in D$ en el que ninguna restricción $c \in C$ sea violada.