Tema 1

1. Definición de Grafo

- **Grafo simple:**
  - Conjunto de vértices V y conjunto de aristas A.
  - Sin aristas múltiples, lazos, dirección ni pesos.

2. Variantes de Grafos

- **Multigrafo:** Permite aristas múltiples entre los mismos vértices.
- **Pseudografo:** Multigrafo que además permite lazos (aristas de un vértice consigo mismo).
- **Grafo dirigido (Digrafo):** Las aristas tienen sentido.
- **Grafo ponderado:** Las aristas tienen pesos.

---

3. Grado o Valencia de un Vértice

- $\delta(v)$: Número de aristas incidentes en el vértice v.
- **Lista de grados**: lista ordenada de grados de todos los vértices.
- **Grafo k-regular**: cuando todos los vértices tienen la **k** valencia
- **Lema del apretón de manos:** $\sum_{v\in V} \delta(v) = 2|A|$
- **En digrafos**: grado de entrada $\delta_e(v)$ y grado de salida $\delta_s(v)$.

Observaciones:

- $0 \leq \delta(v) \leq n-1$ para cualquier vértice v en un grafo simple.
- No pueden coexistir vértices con grados 0 y $n-1$ simultáneamente en el mismo grafo. (Ejemplo de la fiesta)

---

4. Ejemplos Importantes de Grafos

- **Grafo completo K_n:** Todos los vértices están conectados entre sí.
- **Grafo camino P_n:** Vértices forman una secuencia lineal.
- **Grafo ciclo C_n:** Vértices forman un ciclo cerrado.
- **Grafos bipartitos:** Se pueden dividir los vértices en dos grupos, sin aristas internas.
- **Grafos bipartitos completos $K_{n,m}$:** Cada vértice de un grupo conectado con todos los vértices del otro grupo.

---

5. Subgrafos

- **Subgrafo inducido por S:** Si S es un subconjunto de vértices, G(S) es el subgrafo formado por los vértices de S y las aristas entre ellos.
- **Subgrafo recubridor:** Cuando G' contiene todos los vértices del G original. (Tema 3)
- **Eliminación de un vértice:** Al eliminar un vértice, se eliminan también todas las aristas incidentes a dicho vértice.
- **Eliminación de una arista:** Al eliminar una arista, se elimina únicamente dicha conexión entre los vértices que unía.

---

6. Suma de Grafos

- Unión de conjuntos de vértices y aristas, más todas las aristas que conectan vértices de un grafo con los del otro.

---

7. Grafo Complementario

- Misma cantidad de vértices; dos vértices conectados en el complementario si no lo están en el original.
- Propiedad: $G \cup \bar{G} = K_n$

---

8. Isomorfismo de Grafos

- Dos grafos son isomorfos ($G_1 \cong G_2$) si existe una correspondencia biyectiva entre sus vértices que preserva las aristas.
- Condiciones NECESARIAS PERO NO SUFICIENTES para isomorfismo:
  - Mismo $|V|$ y mismo $|A|$
  - Misma lista de grados.
  - Mismo número de ciclos, vértices de corte y mismas componentes conexas.

Propiedad: Dos grafos son isomorfos <=> Sus complementarios lo son.

---

9. Formas de Representar un Grafo

- **Exhaustiva:** Listas explícitas de vértices y aristas.
- **Lista de adyacencias:** Cada vértice seguido de la lista de vértices adyacentes.
- **Matriz de adyacencias:** Matriz cuadrada simétrica; entrada $A_{ij} = 1$ si vértices i, j adyacentes; 0 si no.
- **Matriz de incidencias:** Filas representan vértices, columnas aristas; entrada indica incidencia.

Variantes en matrices:

- **Aristas múltiples**: Indican número de aristas entre vértices.
- **Lazos**: Indican doble número del lazo del vértice.
- **Digrafos**: La matriz de adyacencias ya no es simétrica.
- **Grafos ponderados**: Indican peso en lugar de 1.

Tema 2

1. Grafos Conexos

Definiciones:

- Camino: Secuencia de aristas en la que cada una es incidente en el siguiente vértice.
- Grafo conexo: Un grafo G es conexo si existe un camino entre cualquier par de vértices.

Observaciones:

- Si no es conexo, se dice disconexo.
- Los subgrafos conexos máximos de G se llaman componentes conexas.
- (si G es conexo entonces tiene una única componente conexa que es el propio G)

---

2. Conexión en la Matriz de Adyacencia

- $A[i][j] = 1$ si los vértices i y j son adyacentes; 0 en caso contrario.
- $A^k[i][j]$: Número de caminos de longitud k entre i y j. (Recordar Lab 2)

Propiedad:

- G es conexo $\Leftrightarrow$ Para toda pareja de vértices i, j, existe k tal que $A^k[i][j]! = 0$.

---

3. Algoritmo DFS (Búsqueda en profundidad)

Objetivo: Detectar componentes conexas. También es la misma estrategia para escapar de un laberinto

Procedimiento:

1. Establecer un orden en los vértices del grafo.
2. Selecciona un vértice v.
3. Explora los vértices en ese orden hasta que te ves obligado a retroceder.
4. Marca los vértices visitados.
5. Repetir desde otro vértice si quedan sin visitar.

Resultado: Se obtiene un subgrafo del grafo original G que contiene todos sus vértices, llamado "árbol de expansión" o "árbol de búsqueda en profundidad", correspondiente a una componente conexa.

Observaciones:

- Los subgrafos obtenidos mediante DFS dependen del orden elegido en los vértices.
- Si al terminar un recorrido DFS no se han visitado todos los vértices, es necesario iniciar otro DFS desde un vértice no visitado. Esto indica que el grafo es disconexo.

---

4. Conexión en Dígrafos

Tipos:

- Fuertemente conexo: Hay camino de i a j y de j a i. (Dos sentidos)
- Unilateralmente conexo: Hay camino de i a j o de j a i. (Un solo sentido)
- Débilmente conexo: Es conexo si nos olvidamos del sentido de las aristas.

Observación:

- Estas definiciones no son excluyentes; un mismo dígrafo puede ser, por ejemplo, débil y unilateralmente conexo pero no fuertemente conexo.

---

5. Algoritmo de Tarjan

Objetivo: Sirve para hallar las componentes fuertemente conexas de un dígrafo

Pasos:

1. Realizar DFS sobre G, creando una lista L (de inicio a final) en la que se van añadiendo los vértices conforme se quedan sin vecinos no visitados ("o cuando tengo que retroceder").
2. Calcular el grafo traspuesto G^T: invirtiendo sentido de las aristas en G.
3. Realizar DFS en G^T, usando el orden inverso de L (desde el final hasta el inicio).

Resultado: Componentes fuertemente conexas.

---

6. k-Conectividad por Vértices

Definiciones:

- Vértice de corte (V.D.C): Su eliminación desconecta G o lo trivializa.
- $k(G)$: Mínimo número (exacto) de vértices que hay que eliminar para desconectar G (o para que G sea trivial).
- Bloques de un grafo: son los mayores subgrafos que son 2-conexos (Recordar Lab 2)
- Caminos disjuntos: Dos caminos entre dos vértices x,y si no tienen vértices en común (excepto x,y)

Propiedades:

- Si G es disconexo, entonces $k(G) = 0$.
- G es r-conexo si $k(G) \geq r$: es decir, si elimino MENOS DE r vértices G sigue siendo conexo

Observación:

- No se debe decir “v y v son vdc’s”. En su lugar, se debe decir “{v , v } es un conjunto de corte”.

Teorema de Menger:

- G es r-conexo $\Leftrightarrow$ hay al menos r caminos disjuntos entre cada par de vértices.

---

7. k-Conectividad por Aristas

Definiciones:

- Arista puente: Su eliminación desconecta G o lo trivializa.
- $\lambda(G)$: Mínimo número de aristas necesarias para desconectar G.

Propiedades:

- Si G es disconexo, entonces $\lambda(G) = 0$.
- G es r-conexo por aristas si $\lambda(G) \geq r$.

Teorema de Menger (por aristas):

- G es r-linealmente conexo $\Leftrightarrow$ entre cada dos vértices siempre hay al menos r caminos disjuntos por aristas.

---

8. Teorema de Whitney

Relación:

$k(G) \leq \lambda(G) \leq \delta(G)$

Donde:

- $k(G)$: índice de conectividad por vértices.
- $\lambda(G)$: índice de conectividad por aristas.
- $\delta(G)$: mín(lista_grados(G))

Observación:

- Alta conectividad requiere grados altos, pero grados altos no aseguran alta conectividad.

Tema 3

1. Definición de Árbol

- **Árbol**: Grafo *conexo* y *sin ciclos*.

---

2. Propiedades Fundamentales

1. Existe un **único camino** entre cualesquiera dos vértices.
2. Si se **añade una arista**, se forma un **ciclo**, y **deja de ser árbol**.
3. Si se **elimina una arista**, el grafo se parte en **dos componentes conexas**, cada una de ellas es un árbol.
4. Para **añadir un vértice** y que siga siendo un árbol, se debe conectar mediante **una única arista**.
5. Un árbol con $|V|$ vértices tiene $|A| = |V| - 1$ aristas.

---

3. Caracterizaciones Equivalentes

Las siguientes afirmaciones son equivalentes:

1. $T = (V, A)$ es un árbol.
2. Existe un único camino entre cada par de vértices de T.

3. T es conexo y $|A| = |V| - 1$.
4. T es acíclico y $|A| = |V| - 1$.

---

4. Bosque

- **Bosque**: Grafo *disconexo* y sin ciclos.
- Es decir, cada componente conexa de un bosque es un árbol.

---

5. Árbol Enraizado

- Árbol con un **vértice raíz** marcado.
- Los vértices se **organizan por niveles**. Ojo! Empezar a contar desde 0.
- Conceptos:
  - **Raíz**: vértice inicial desde donde se organiza el árbol.
  - **Padre / Hijo**: relación entre vértices conectados directamente desde un nivel superior (i) al siguiente (i+1).
  - **Ascendiente / Descendiente**: relación jerárquica entre vértices de distintos niveles.
  - **Hermanos**: vértices que tienen el mismo padre.
  - **Hoja**: vértice sin hijos (terminal).
  - **Altura**: distancia (en número de niveles) desde la raíz hasta la hoja más lejana (el mayor nivel que existe).

---

6. Árboles m-arios

- **Árbol m-ario**: todos los vértices internos tienen exactamente m hijos.
- **Árbol m-ario completo**: si además todas las hojas están en el mismo nivel.

Fórmulas:

1. Número de hojas:

$$x \leq m^h \quad \text{o equivalentemente} \quad h \geq \log_m x$$

   donde:

   - x: número de hojas
   - h: altura

2. Número total de vértices:

$$|V| = mi + 1$$

   donde:

   - i: número de vértices internos

3. En un árbol binario:

$$|V| = 2i + 1, \quad \text{con } i \text{ internos, } i + 1 \text{ hojas}$$

---

7. Árboles de Decisión

- Árbol enraizado donde:
  - Vértices internos = **comprobaciones o decisiones**
  - Aristas = **resultados de la comprobación**
  - Hojas = **conclusiones finales**

---

8. Árboles Recubridores

- **Subgrafo recubridor**: subgrafo que contiene todos los vértices del grafo original. (subgrafo conexo)
- Si es un **árbol**, se llama **árbol recubridor**. (+ sin ciclos)
- Si el grafo es disconexo, obtenemos un **bosque recubridor**. (las cc son arboles recubridores)

Algoritmos para construir árboles recubridores:

1. **DFS (Depth First Search)**
   - Usa **pila (LIFO)**.
   - Explora hasta el fondo antes de retroceder.
2. **BFS (Breadth First Search)**
   - Usa **cola (FIFO)**.
   - Explora por niveles.

- Observación: Ambos algoritmos son válidos también para grafos dirigidos.

8.1. Algoritmo BFS (Búsqueda en anchura)

Objetivo: Obtener caminos más cortos desde un vértice inicial (en G no ponderado), detectar componentes conexas y recorrer grafos por niveles.

Procedimiento:

1. Establecer un orden en los vértices del grafo.
2. Seleccionar un vértice inicial v.
3. Visitar todos los vértices vecinos inmediatos de v.
4. Continuar con los vecinos de los vecinos, y así sucesivamente, por niveles.
5. Marcar los vértices visitados.
6. Si quedan vértices sin visitar, reiniciar desde otro vértice no visitado.

Resultado: Se obtiene un subgrafo llamado “árbol de expansión” o “árbol de búsqueda en anchura” correspondiente a una componente conexa del grafo.

Observaciones:

- Se utiliza una cola (FIFO) para gestionar los vértices pendientes de explorar.
- Ideal para encontrar caminos más cortos en grafos no ponderados.
- El recorrido depende del orden de los vértices.

8.2. Aplicaciones de DFS y BFS

Aplicaciones del DFS (Depth First Search):

- Determinar si un grafo es conexo / Detectar **componentes conexas** en grafos disconexos.
- Hallar **componentes fuertemente conexas** en digrafos (TARJAN).
- Encontrar **vértices de corte**: V es VDC <=> DFS con raiz en V tiene más de un hijo.
- Recorrido útil para **escapar de un laberinto** (explora un camino hasta el final y retrocede si es necesario).

Aplicaciones del BFS (Breadth First Search):

- Encontrar **caminos más cortos** en grafos no ponderados.
- Resolver problemas tipo **laberinto** encontrando la salida por el CAMINO MÁS CORTO.

Los 2 sirven para:

- Hallar componentes conexas => Conseguir bosques recubridores

Observación: Si al aplicar DFS o BFS no se alcanzan todos los vértices, el grafo no es conexo. Los vértices alcanzados forman una **componente conexa**, y se puede aplicar el algoritmo nuevamente para encontrar las demás.

---

9. Árboles Recubridores en Grafos Ponderados

Tipos:

1. **Árbol recubridor de peso mínimo**:
   - Minimiza la suma total de pesos.
   - Algoritmo clásico: **Kruskal**: Añadimos las aristas ordenadas en orden decreciente de peso siempre que no generen ciclos hasta que sea conexo (algoritmo voraz).
   - **No garantiza caminos más cortos entre vértices.**
2. **Árbol recubridor de camino mínimo**:
   - Desde una **raíz dada**, da los caminos más cortos al resto.
   - Algoritmo: **Dijkstra**
   - **No garantiza caminos mínimos entre cualquier par de vértices.** añadir clase

---

10. Distancias en Grafos

- **Distancia** $d(u, v)$: es la longitud del camino más corto que une los vértices u y v.
  - En grafos simples (no ponderados), la longitud del camino es el número de aristas.
  - En grafos ponderados, la longitud de un camino es la suma de los pesos de sus aristas.
- **Excentricidad** de un vértice $v \in V$:
- **Radio**: $\min_{v \in V} e(v)$
- **Diámetro**: $\max_{v \in V} e(v)$

Tema 4: Transversalidad

Definiciones

- **Recorrido euleriano**: recorre todas las aristas exactamente una vez, puede empezar y terminar en vértices distintos.
- **Circuito euleriano**: recorrido euleriano cerrado que empieza y termina en el mismo vértice.
- **Grafo semi-euleriano**: grafo conexo que admite un recorrido euleriano.
- **Grafo euleriano**: grafo conexo que admite un circuito euleriano.

Teoremas (CONDICIONES NECESARIAS Y SUFICIENTES)

- **Teorema (carácter euleriano):**

  Un grafo (o multigrafo) conexo es **euleriano si y sólo si** todos sus vértices tienen **grado par**.

- **Teorema (carácter semi-euleriano):**

Un grafo (o multigrafo) conexo es **semi-euleriano si y sólo si** todos sus vértices tienen grado par **excepto dos**.

Observación:

Si en G todos los vértices tienen grado par, cada uno está contenido en al menos un ciclo.

Algoritmo para encontrar circuito euleriano EN UN G EULERIANO

1. Sea v un vértice cualquiera y $C = \{v\}$
2. Mientras haya aristas sin visitar:
   - Escoger un vértice $u \in C$ con aristas incidentes sin visitar.
   - Formar un ciclo D comenzando en u.
   - Marcar aristas de D como visitadas.
   - Reemplazar u por D en C
3. Resultado: C es el circuito euleriano.

Observaciones:

- Nótese que tanto en D como en C es un {} donde se comienza y termina con el mismo vértice
- Se puede escoger el mismo u varias veces.

Algoritmo para encontrar recorrido euleriano EN UN G SEMI-EULERIANO

1. Identificar los dos vértices de **grado impar**, llamémoslos u y v.
2. Añadir un nuevo vértice ficticio w, y conectarlo con dos nuevas aristas: $w - u$ y $w - v$.
3. Esto forma un nuevo grafo G', **euleriano**, ya que ahora todos los vértices tienen grado par.
4. Aplicar el algoritmo de circuito euleriano sobre G', comenzando en w. Obtenemos el circuito.
5. Eliminamos el vértice añadido del circuito. Lo que obtenemos es un recorrido euleriano para G.

Carácter euleriano en dígrafos

Sea G un dígrafo débilmente conexo. Entonces:

G es euleriano si y solo si todos sus vértices v cumplen:

$$\delta_s(v) = \delta_e(v)$$

G admite un recorrido euleriano si y solo **si cumplen la condicion anterior excepto 2 vertices**:

Uno con:

$$\delta_s(v) = \delta_e(v) + 1$$

Otro con:

$$\delta_e(v) = \delta_s(v) + 1$$

Esto implica que el recorrido euleriano comienza en el vértice con mayor grado de salida y termina en el vértice con mayor grado de entrada.

---

2. Grafos Hamiltonianos

Definiciones

- **Camino hamiltoniano**: pasa por todos los vértices exactamente una vez.
- **Ciclo hamiltoniano**: camino hamiltoniano cerrado (empieza y termina en el mismo vértice).
- **Grafo semi-hamiltoniano**: grafo que admite un camino hamiltoniano.
- **Grafo hamiltoniano**: grafo que admite un ciclo hamiltoniano.

Observación:

Si G es hamiltoniano => también es semi-hamiltoniano. (NO PASA CON EULER)

Cuidado

- Determinar si un grafo es hamiltoniano es **NP-completo**.
- No existen condiciones necesarias y suficientes simples ni un algoritmo eficiente general para encontrar un ciclo hamiltoniano.

Condiciones necesarias para que G SEA HAMILTONIANO

- No puede haber un vértice con **grado 1**.
- Si un vértice tiene **grado 2**, sus aristas deben estar en el ciclo.
- No pueden haber **vértice de corte**.
- Si al eliminar c vértices quedan más de c componentes conexas $\rightarrow$ NO ES HAMILTONIANO.

Condiciones suficientes para que G SEA HAMILTONIANO

Sea G un grafo simple conexo con $n \geq 3$ vértices:

Condición de Dirac:

Si todos los vértices tienen grado $\geq n/2$, entonces G tiene ciclo hamiltoniano.

Condición de Ore:

Si la suma de grados de cada par de vértices no adyacentes u, v es $\geq n$, entonces G tiene ciclo hamiltoniano.

Nota: Estas son condiciones **suficientes pero no necesarias**. El C5 por ejemplo es un grafo hamiltoniano que no verifica ninguna de esas condiciones.

Condiciones para Caminos Hamiltonianos

Necesarias:

- Si hay **vértice de grado 1**, debe ser extremo del camino.
- No más de dos vértices con grado 1.
- Si hay vértice de corte cuya eliminación provoca más de 2 componentes $\rightarrow$ no hay camino.
- Si al eliminar c vértices hay más de $c + 1$ componentes $\rightarrow$ no hay camino.

Solo una suficiente (Dirac):

Si G es simple, conexo, $n \geq 3$, y todos los vértices tienen grado $\geq (n-1)/2$, entonces **admite camino hamiltoniano**.

Nota: Es solo una condición **suficiente pero no necesaria**. El C5 - 1 arista por ejemplo es un grafo hamiltoniano que no verifica ninguna de esas condiciones.

Tema 5 Parte 1: Coloreado

1.1 Definición:

Una k-coloración es una coloración que utiliza exactamente colores distintos.

Una *k-coloración por vértices* es una función $c : V \to \mathbb{N}$ tal que $c(u) \neq c(v)$ si u y v son adyacentes.

1.2 Número cromático:

$\chi(G)$ = número cromático = mínimo número de colores en una coloración válida de vértices.

1.3 Propiedades:

- $\chi(G) \leq |V|$
- Si G' es subgrafo de G, entonces $\chi(G') \leq \chi(G)$
- Si G no es conexo, $\chi(G) = \max\{\chi(G_1), \dots, \chi(G_c)\}$ con $G_1, \dots, G_c$ componentes conexas.
- $\chi(G) = 1 \iff G$ sin aristas.
- $\chi(G) = 2 \iff G$ es bipartito.

1.4 Casos particulares:

- **Grafo completo** K_n: $\chi(K_n) = n$
- **Ciclo** C_n:
  - Si n par: $\chi(C_n) = 2$
  - Si n impar: $\chi(C_n) = 3$
- **Triangulo**: Se necesitan 3 colorse para colorear un triangulo. ### 1.5 Observación:
- Una coloración por vértices induce una partición del conjunto de vértices en conjuntos independientes, donde los vértices de cada conjunto tienen el mismo color y no son adyacentes entre sí.

1.5 Algoritmo voraz (coloración de vértices)

1.5.1 Procedimiento:

1. Fijar orden de los vértices.
2. Asignar a cada vértice el primer color disponible.

1.5.2 Observaciones:

- Podemos obtener distintas coloraciones (y distinto número de colores) con distintos órdenes para el conjunto de vértices.
- Da una coloracion, pero no garantiza usar el mínimo número de colores. A no ser que ejecutemos el algoritmo $|V|!$ veces y nos quedemos con el menor número de colores.

1.6 Caracterización de grafos bipartitos

1.6.1 Teoremas equivalentes:

G es bipartito $\iff \chi(G) = 2 \iff G$ no tiene ciclos impares

1.6.2 [Friki] Demostración estructural (niveles):

- Representar G por niveles.
- Vértices en niveles pares $\leftrightarrow$ color 1
- Vértices en niveles impares $\leftrightarrow$ color 2

1.7 Cotas para el número cromático

1.7.1 Teoremas:

- $\chi(G) \leq \Delta + 1$, donde Δ es el grado máximo de G.
- **Teorema de Brooks**: Si G no es un grafo completo ni un ciclo impar, entonces $\chi(G) \leq \Delta$

2. Coloreado de aristas

2.1 Definición:

Una *k-coloración por aristas* es una función $c : E \to \mathbb{N}$ tal que $c(e) \neq c(e')$ si e y e' son incidentes.

2.2 Índice cromático

2.2.1 Definición:

$\chi_1(G)$ = índice cromático = mínimo número de colores necesarios para una coloración por aristas.

2.2.2 Cotas:

Teorema de Vizing: $\Delta \leq \chi_1(G) \leq \Delta + 1$

2.2.3 Casos particulares:

- $\chi_1(K_n) = \begin{cases} n-1, & \text{si } n \text{ par} \\ n, & \text{si } n \text{ impar} \end{cases}$
- Para grafos bipartitos: $\chi_1(G) = \Delta$

2.3 Algoritmo voraz (coloración por aristas)

2.3.1 Procedimiento:

1. Establecer un orden en las aristas.
2. Asignar a cada arista el primer color disponible.

2.3.2 Observaciones:

- Aplican las mismas que en el caso del Algoritmo de Coloreado de Vertices pero con las aristas.

3. Grafo de línea $L(G)$

3.1 Definición:

Sea $G = (V, E)$ un grafo.

El *grafo de línea* de G, denotado por $L(G) = (V', E')$, es un grafo donde:

- Los vértices de $L(G)$ son las aristas de G: $V' = E$.
- Dos vértices $e, e' \in V'$ son adyacentes en $L(G)$ si y solo si las aristas e y e' son incidentes en un mismo vértice de G.

Es decir, $e \sim e' \iff e \cap e' \neq \emptyset$ en G.

3.2 Propiedades:

- Una coloración por aristas de G es equivalente a una coloración por vértices de su grafo de línea $L(G)$.
- El índice cromático de G es igual al número cromático de $L(G)$:

$$\chi_1(G) = \chi(L(G))$$

- Nos viene a decir que el problema de encontrar una coloracion por vertices es equivalente al problema de las aristas

Tema 5 Parte 2: Emparejamiento

Definiciones clave

- **Emparejamiento**: Subconjunto de aristas de G no incidentes entre sí.
- **Emparejamiento maximal**: Emparejamiento con el mayor número posible de aristas. (El mayor número de asignaciones posibles).
- **Emparejamiento completo**: Cuando "todos los trabajadores tienen una tarea asignada"

Condición y Teorema de Hall

- **Condición de Hall**:
  - P es cualquier subconjunto de X (trabajadores).

  $$\forall P \subseteq X, \quad |T(P)| \geq |P|$$

  donde $T(P) = \{y \in Y \mid \exists x \in P \text{ tal que } (x, y) \in E\}$
- **Teorema de Hall**:
  - Un grafo bipartito $G = (X \cup Y, E)$ tiene un emparejamiento completo si y solo si cumple la condición de Hall.
- **Observación**:
  - Comprobar la condición de Hall directamente requiere $2^{|X|}$ comprobaciones. INVIABLE!!

Caminos alternados

- **Definición**:
  - Dado un emparejamiento M, un **camino alternado** es una secuencia de aristas donde:
    * Las posiciones impares (1ra, 3ra, ...) no están en M
    * Las pares (2da, 4ta, ...) sí están en M
    * El camino tiene longitud impar.
- **Ventaja**:
  - Si se encuentra un camino alternado, al intercambiar las aristas en M por las del camino se obtiene un nuevo emparejamiento con una arista más.

Árbol de caminos alternados

- **Construcción**:
  1. Nivel 0: trabajador libre x_0
  2. Nivel 1: tareas adyacentes a x_0
  3. Nivel 2: trabajadores emparejados con las tareas del nivel anterior
  4. Nivel 3: tareas adyacentes a los trabajadores del nivel anterior
  5. Y así sucesivamente...
- **Propiedad**:
  - Si el emparejamiento M no es maximal, entonces **siempre existe** un camino alternado en G.

- **Criterio**:
  - Si hay una **hoja en un nivel impar** del árbol, entonces existe un camino alternado.

Algoritmo de mejora de emparejamiento

1. Comenzar con un emparejamiento inicial M.
2. Buscamos un camino alternado para M construyendo el **árbol de camino alternado**

- Si se encuentra un camino alternado:
  - Intercambiar aristas para obtener M' con una arista más.
  - Volver al paso 2.
- Si no se encuentra camino alternado:
  - M es EMPAREJAMIENTO MAXIMAL.
  - Si además $|M| = |X|$, entonces TAMBIÉN ES COMPLETO.

Tema 6: Planaridad

Inmersiones y Grafos Planos

- **Inmersión**: Representación de un grafo donde **no se cruzan las aristas**.
- **Grafo plano**: Un grafo que **admite una inmersión**.

Propiedades de los grafos planos

- Las **caras** de una inmersión pueden ser **interiores** o **exteriores**.
- Las **aristas puente**: Conectan caras con otras caras o con nada.
- Las **aristas frontera** cumplen que:
  - Forman **ciclos**.
  - Si una arista pertenece a un ciclo   es arista frontera.
  - Bordean **dos caras**.
- En un G plano con $|E| > 2$, **cada cara tiene al menos tres aristas frontera**. (Pensar en el triangulo)

Fórmulas:

- Si G tiene c caras y a aristas: $3c \leq 2a$
- Si **además** cada cara tiene al menos b aristas frontera: $bc \leq 2a$
- Si **G no tiene triangulos**: $4c \leq 2a$

Fórmula de Euler

- Para un G **plano conexo** con v vértices, a aristas y c caras:

$$v + c = a + 2$$

- Para un G **plano conexo** con d **componentes conexas**:

$$v + c = a + d + 1$$

Un test de Planaridad (NECESARIA PERO NO SUFICIENTE)

- Para un G **conexo** con $v > 2$, si G es plano, entonces:

$$a \leq 3v - 6$$

- Consecuencia:
  - Si $a > 3v - 6$ **No es plano**.
  - Si $a \leq 3v - 6$ **Solo sabemos que PUEDE ser plano**. Por ejemplo, el K3,3 satisface la condición pero NO es plano.

Grafos Planos Maximales

- Un G plano es **maximal** si **añadir cualquier arista lo hace no plano**.
- **Propiedades**:
  - Cada cara está rodeada por **exactamente 3 aristas** (caras triangulares).
  - No tienen **aristas puente**.
  - Siempre son **conexos**.
  - Cumplen: $a = 3v - 6$

Teorema de Kuratowski

- **Un grafo es plano no contiene una subdivisión de K_5 o $K_{3,3}$.**

Teorema de Wagner

- **Un grafo es plano no contiene un subgrafo que pueda contraerse a K_5 o $K_{3,3}$.**