



Tecnológico
de Monterrey

TC2038

Análisis y diseño de algoritmos avanzados

Dr. Frumencio Olivas Alvarez

frumen@tec.mx

Contenido

- ☐ Problemas de coloreo de grafos:
 - ☐ Decisión.
 - ☐ Optimización.
 - ☐ Permutación.

Problema de coloreo de grafos

Consiste en añadir color a cada nodo de un grafo, de tal manera que ningún nodo adyacente tenga el mismo color.

Existen variantes (todas con la misma restricción), tales como:

- Buscar la cantidad mínima de colores para un grafo.
- Saber si un grafo se puede pintar o no con un número determinado de colores.
- Descubrir todas las posibles formas de pintar un grafo.

Historia



Aplicaciones

- ❖ Colorear mapas.
- ❖ Asignación de registros en compiladores (register allocation).
- ❖ Solución al rompecabezas Sudoku.
- ❖ Planificar actividades, tareas o exámenes (scheduling problems).

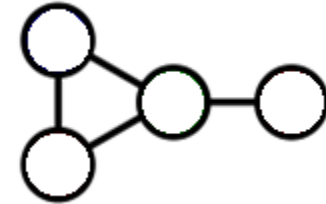
Problema de **decisión** de coloreo de grafos

Reside en saber si un grafo se puede colorear o no, con una cantidad definida de colores. Tomando en cuenta la restricción de que ningún par de nodos adyacentes deben tener el mismo color.

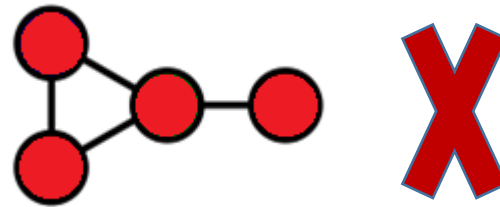
Se debe proporcionar, por lo menos, un grafo y el número máximo de colores permitidos. La función solo devuelve un valor Booleano dependiendo si se puede o no.

Ejemplos

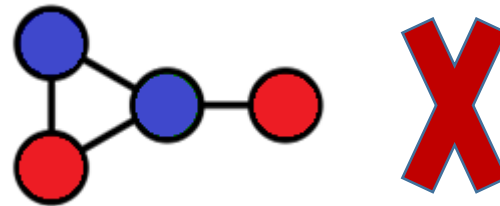
Se puede colorear el siguiente grafo:



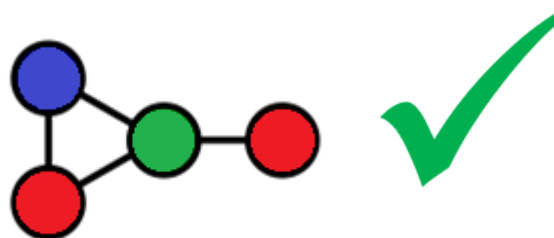
Con un solo color ¿?



Con dos colores ¿?



Con tres colores ¿?



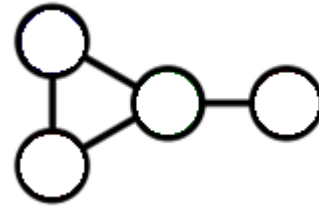
Problema de **optimización** de coloreo de grafos

Encontrar la cantidad mínima de colores necesarios para colorear un grafo, cumpliendo con la restricción de que ningún par de nodos adyacentes puede tener el mismo color.

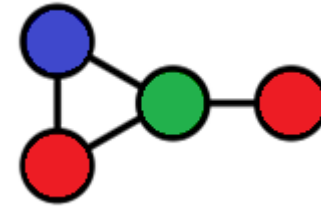
Se debe proporcionar solamente el grafo, y devuelve la cantidad mínima de colores para el grafo dado.

Ejemplos

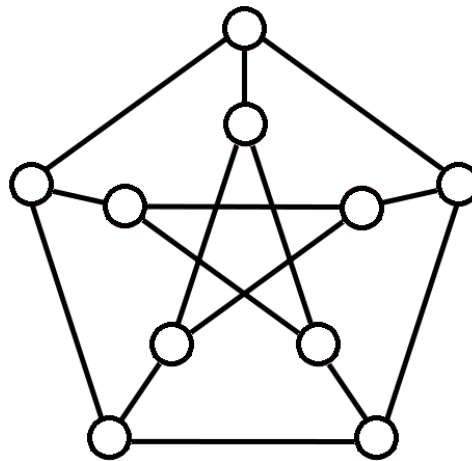
Cuantos colores ¿?



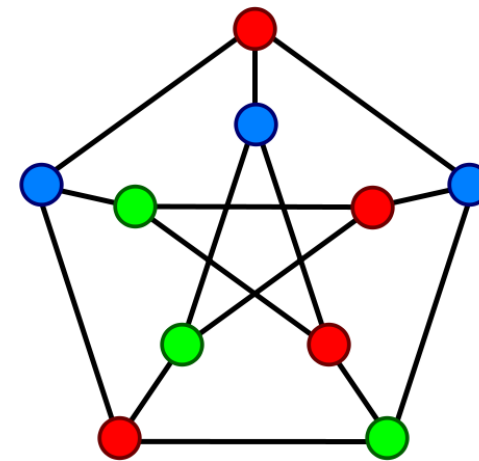
3



Cuantos colores ¿?

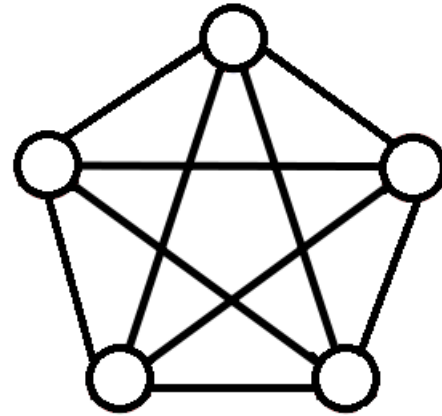


3

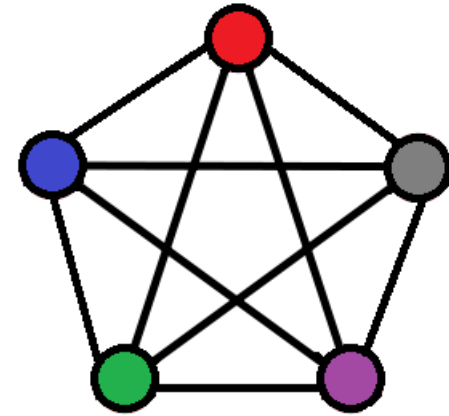


Ejemplos

Cuántos colores ¿?



5



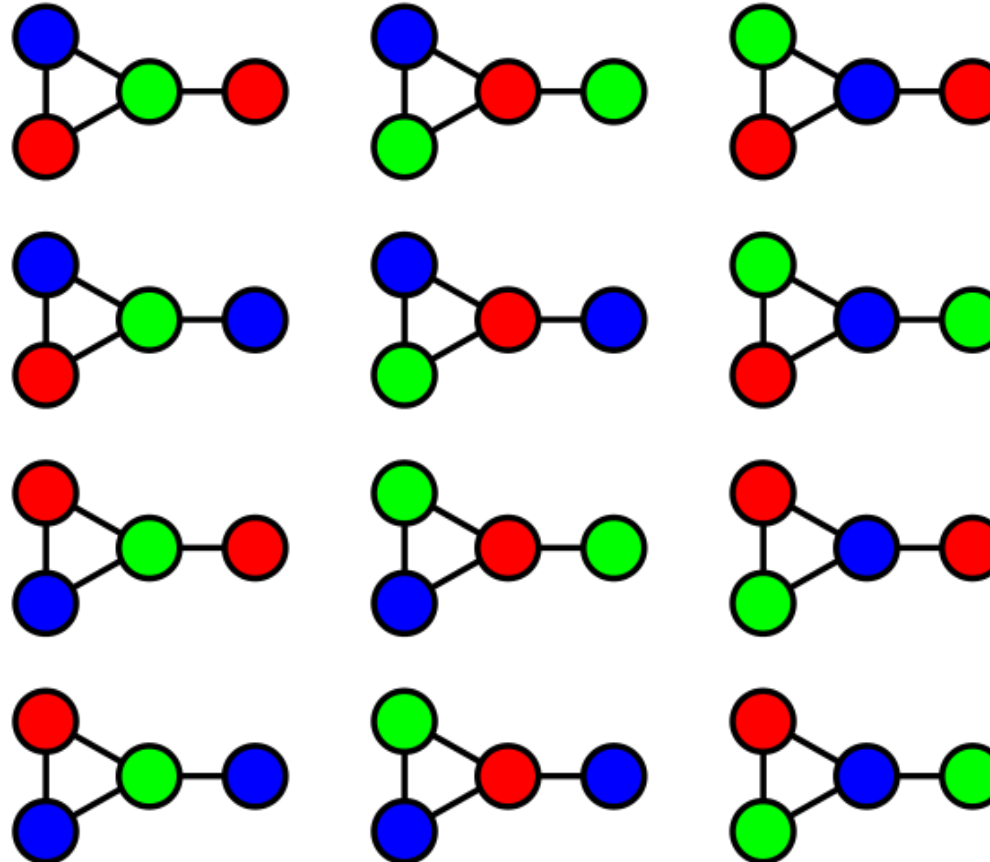
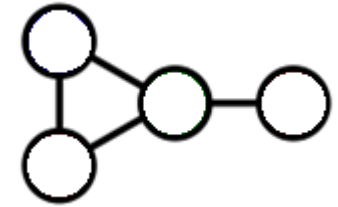
Problema de **permutación** de coloreo de grafos

Responde a la pregunta ¿Cuántas maneras diferentes existen de colorear un mismo grafo con un número determinado de colores?

Es similar al problema de optimización, solo que aquí no se detiene cuando encuentra el número menor de colores, si no que sigue buscando distintas permutaciones (o combinaciones) posibles.

Ejemplos

De cuantas maneras se puede colorear el grafo:



Pseudocódigo del problema de **decisión**

Entradas: matriz del **grafo**, vector de **color**, **m** número de colores, nodo **v** actual y **n** número de nodos.

Caso base: si el nodo actual es igual a número de nodos:

Retornar verdadero.

Por cada color **c** desde 1 hasta **m** colores:

Si es valido utilizar el color **c**:

Guardar **c** en color en posición del nodo actual.

Si la recursión con nodo actual más 1 retorna verdadero:

Retornar verdadero.

Guardar 0 en color en posición del nodo actual.

Retornar falso.

Pseudocódigo del problema de **optimización**

Entradas: matriz del **grafo**.

Crear vector color inicializado en -1 y vector asignados de booleanos.

Asignar el color 1 al primer nodo en color.

Por cada nodo i de 1 hasta $n-1$:

 Inicializar todos los valores de asignados en falso.

 Por cada nodo j de 0 hasta $n-1$:

 Si el grafo en $[i,j]$ contiene un 1:

 Si color en $[j]$ es diferente de -1:

 Guardar verdadero en asignados en la posición color en $[j]$.

 Por cada color c de 1 hasta n :

 Si asignados en $[c]$ es falso:

 Guardar el valor de c en color en $[i]$.

 Detener el ciclo.

Pseudocódigo del problema de **permutación**

Entradas: matriz del **grafo**, vector de **color**, **m** número de colores, nodo **v** actual y **n** número de nodos.

Caso base: si el nodo actual es igual a número de nodos:

- Imprimir los valores de color.

- Finalizamos la ejecución.

Por cada color **c** desde 1 hasta **m** colores:

- Si es valido utilizar el color **c**:

 - Guardar **c** en color en posición del nodo actual.

 - Recursión con nodo actual más 1.

 - Guardar 0 en color en posición del nodo actual.

Pseudocódigo

validar uso de un color

Entradas: matriz del **grafo**, vector de **color**, color **c** para validar y nodo **v** actual.

Por cada nodo **u** del grafo:

 Si el grafo en $[v,u]$ tiene un 1:

 Si color en $[u]$ es igual al color **c** buscado:

 Retornar falso.

Retornar verdadero.

Actividad 3.4

Implementación de los algoritmos que resuelven los tres diferentes problemas de coloreo de grafos vistos en esta clase.

Dudas ¿?

