Matemática Discreta

Clase 22: Caminos más cortos y colorabilidad

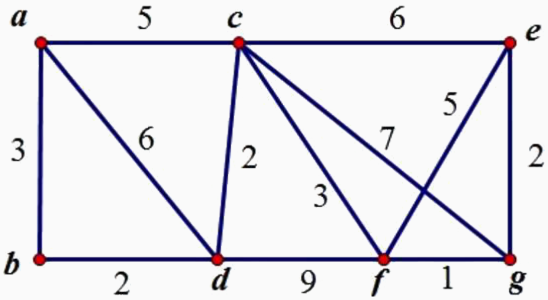
Federico Olmedo y Alejandro Hevia

Departamento de Ciencias de la Computación
Universidad de Chile

Caminos más cortos sobre grafos

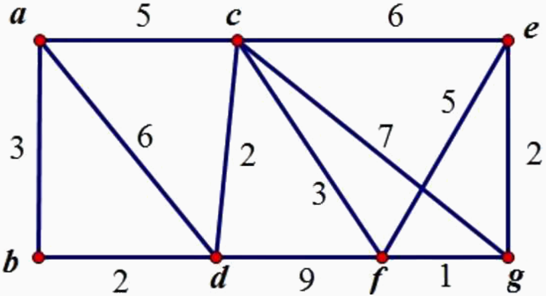
Camino más corto entre dos ciudades

Suponga que el siguiente grafo representa una red de transporte entre distintas ciudades de Chile, donde cada arista está etiquetada con la distancia entre las ciudades que conecta.



Camino más corto entre dos ciudades

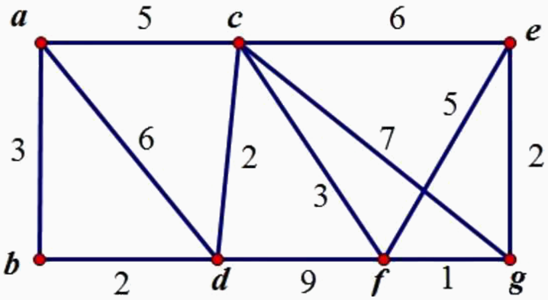
Suponga que el siguiente grafo representa una red de transporte entre distintas ciudades de Chile, donde cada arista está etiquetada con la distancia entre las ciudades que conecta.



Pregunta: ¿Cuál es el la ruta más corta para ir desde a hasta g?

Camino más corto entre dos ciudades

Suponga que el siguiente grafo representa una red de transporte entre distintas ciudades de Chile, donde cada arista está etiquetada con la distancia entre las ciudades que conecta.



Pregunta: ¿Cuál es el la ruta más corta para ir desde a hasta g?

El grafo de arriba se conoce como un grafos con pesos.

Grafos con pesos

- Un grafo con pesos es un grafo donde cada arista tiene asociado un número real no-negativo, su peso. Se formaliza a través de una terna $G = (V, E, w)$ donde $w \colon E \to \mathbb{R}_{\geq 0}$ es la función que le asigna pesos (positivos) a cada arista del grafo.

Grafos con pesos

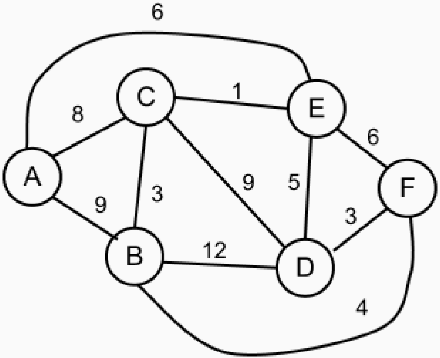
- Un grafo con pesos es un grafo donde cada arista tiene asociado un número real no-negativo, su peso. Se formaliza a través de una terna $G = (V, E, w)$ donde $w\colon E \to \mathbb{R}_{\geq 0}$ es la función que le asigna pesos (positivos) a cada arista del grafo.
- El largo o longitud de un camino es la suma de los pesos de las aristas en el camino.

Grafos con pesos

- Un grafo con pesos es un grafo donde cada arista tiene asociado un número real no-negativo, su peso. Se formaliza a través de una terna $G = (V, E, w)$ donde $w\colon E \to \mathbb{R}_{\geq 0}$ es la función que le asigna pesos (positivos) a cada arista del grafo.
- El largo o longitud de un camino es la suma de los pesos de las aristas en el camino.
- Un camino más corto entre un par de vértices es un camino de longitud mínima entre los mismos. (Observe que dicho camino no es necesariamente único, y que todos ellos van a ser necesariamente simples.)

Aplicación: Envío de mensajes en redes de computadoras

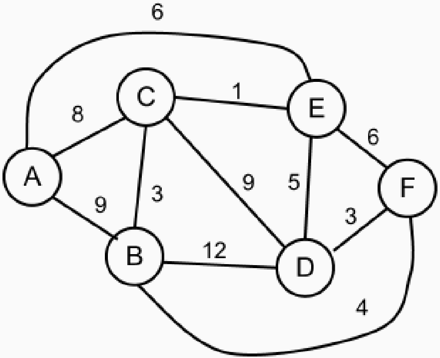
Suponga que el siguiente grafo representa una red de computadoras, donde cada arista contiene el *delay* que presenta cada conexión.



Pregunta: ¿Cuál es la manera más rápida de enviar un mensaje de A a F?

Aplicación: Envío de mensajes en redes de computadoras

Suponga que el siguiente grafo representa una red de computadoras, donde cada arista contiene el *delay* que presenta cada conexión.



Pregunta: ¿Cuál es la manera más rápida de enviar un mensaje de A a F?

Está dada por un camino más corto entre A y F.

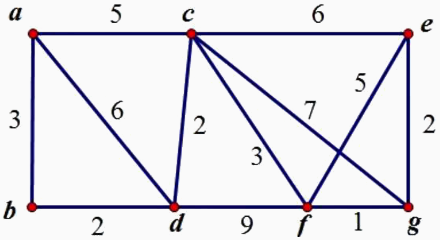
Algoritmo de Dijkstra

El algoritmo de Dijkstra permite calcular el camino más corto desde un origen hacia los nodos restantes de un grafo.

Algoritmo de Dijkstra

El algoritmo de Dijkstra permite calcular el camino más corto desde un origen hacia los nodos restantes de un grafo.

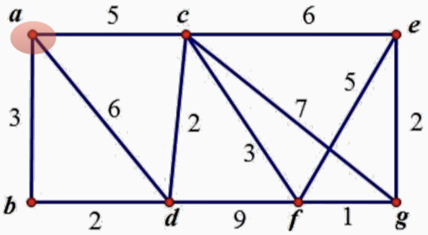
Ejemplo: Calculemos el camino más corto desde *a* hacia el resto de los nodos.



V	a	b	c	d	e	f	g
a							

El algoritmo de Dijkstra permite calcular el camino más corto desde un origen hacia los nodos restantes de un grafo.

Ejemplo: Calculemos el camino más corto desde *a* hacia el resto de los nodos.

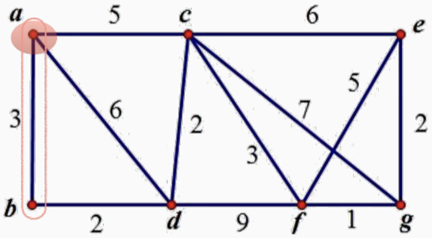


V	a	b	c	d	e	f	g
a	0_a						

Algoritmo de Dijkstra

El algoritmo de Dijkstra permite calcular el camino más corto desde un origen hacia los nodos restantes de un grafo.

Ejemplo: Calculemos el camino más corto desde *a* hacia el resto de los nodos.

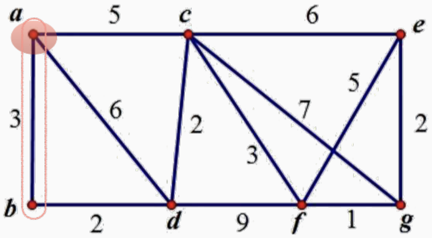


V	a	b	c	d	e	f	g
a	0_a						

Algoritmo de Dijkstra

El algoritmo de Dijkstra permite calcular el camino más corto desde un origen hacia los nodos restantes de un grafo.

Ejemplo: Calculemos el camino más corto desde *a* hacia el resto de los nodos.

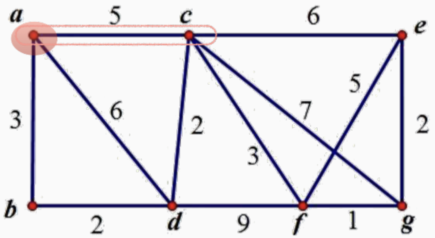


V	a	b	c	d	e	f	g
a	0_a	3_a					

Algoritmo de Dijkstra

El algoritmo de Dijkstra permite calcular el camino más corto desde un origen hacia los nodos restantes de un grafo.

Ejemplo: Calculemos el camino más corto desde *a* hacia el resto de los nodos.

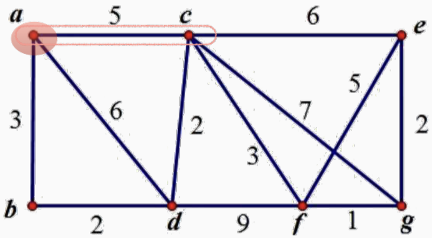


V	a	b	c	d	e	f	g
a	0_a	3_a					

Algoritmo de Dijkstra

El algoritmo de Dijkstra permite calcular el camino más corto desde un origen hacia los nodos restantes de un grafo.

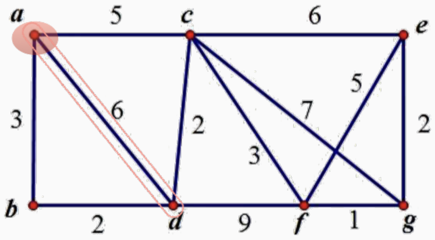
Ejemplo: Calculemos el camino más corto desde a hacia el resto de los nodos.



V	a	b	c	d	e	f	g
a	0_a	3_a	5_a				

El algoritmo de Dijkstra permite calcular el camino más corto desde un origen hacia los nodos restantes de un grafo.

Ejemplo: Calculemos el camino más corto desde *a* hacia el resto de los nodos.

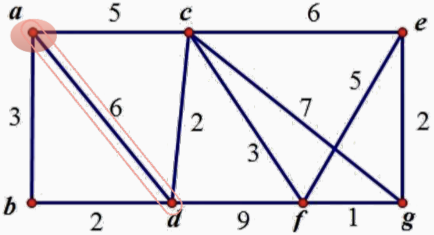


V	a	b	c	d	e	f	g
a	0_a	3_a	5_a				

Algoritmo de Dijkstra

El algoritmo de Dijkstra permite calcular el camino más corto desde un origen hacia los nodos restantes de un grafo.

Ejemplo: Calculemos el camino más corto desde *a* hacia el resto de los nodos.

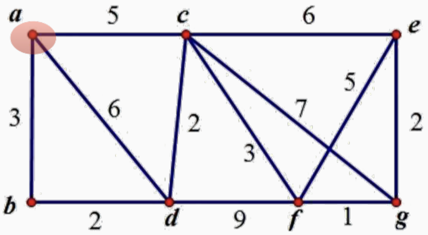


V	a	b	c	d	e	f	g
a	0_a	3_a	5_a	6_a			

Algoritmo de Dijkstra

El algoritmo de Dijkstra permite calcular el camino más corto desde un origen hacia los nodos restantes de un grafo.

Ejemplo: Calculemos el camino más corto desde a hacia el resto de los nodos.

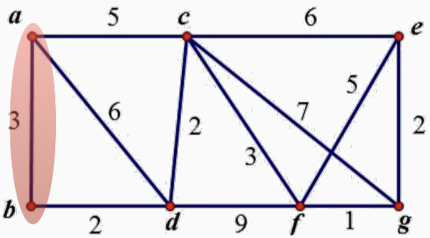


V	a	b	c	d	e	f	g
a	0_a	3_a	5_a	6_a	∞_a	∞_a	∞_a

Algoritmo de Dijkstra

El algoritmo de Dijkstra permite calcular el camino más corto desde un origen hacia los nodos restantes de un grafo.

Ejemplo: Calculemos el camino más corto desde *a* hacia el resto de los nodos.

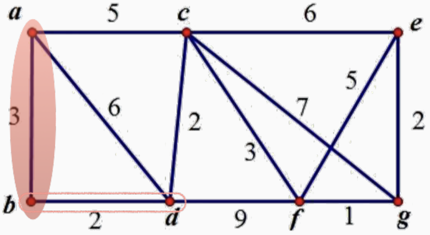


V	a	b	c	d	e	f	g
a	0_a	3_a	5_a	6_a	∞_a	∞_a	∞_a
b	0_a	3_a					

Algoritmo de Dijkstra

El algoritmo de Dijkstra permite calcular el camino más corto desde un origen hacia los nodos restantes de un grafo.

Ejemplo: Calculemos el camino más corto desde *a* hacia el resto de los nodos.

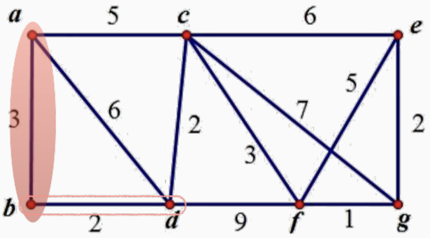


V	a	b	c	d	e	f	g
a	0_a	3_a	5_a	6_a	∞_a	∞_a	∞_a
b	0_a	3_a					

Algoritmo de Dijkstra

El algoritmo de Dijkstra permite calcular el camino más corto desde un origen hacia los nodos restantes de un grafo.

Ejemplo: Calculemos el camino más corto desde a hacia el resto de los nodos.

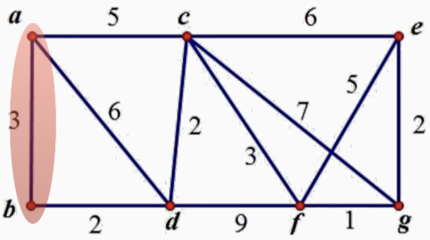


V	a	b	c	d	e	f	g
a	0_a	3_a	5_a	6_a	∞_a	∞_a	∞_a
b	0_a	3_a		5_b			

Algoritmo de Dijkstra

El algoritmo de Dijkstra permite calcular el camino más corto desde un origen hacia los nodos restantes de un grafo.

Ejemplo: Calculemos el camino más corto desde a hacia el resto de los nodos.

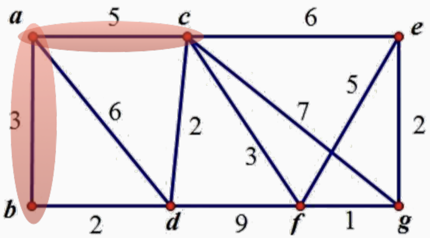


V	a	b	c	d	e	f	g
a	0_a	3_a	5_a	6_a	∞_a	∞_a	∞_a
b	0_a	3_a	5_a	5_b	∞_a	∞_a	∞_a

Algoritmo de Dijkstra

El algoritmo de Dijkstra permite calcular el camino más corto desde un origen hacia los nodos restantes de un grafo.

Ejemplo: Calculemos el camino más corto desde a hacia el resto de los nodos.

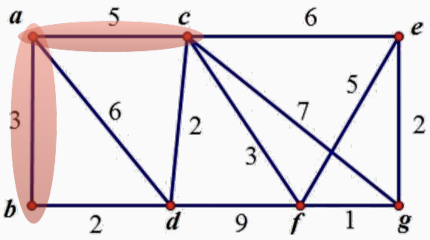


V	a	b	c	d	e	f	g
a	0_a	3_a	5_a	6_a	∞_a	∞_a	∞_a
b	0_a	3_a	5_a	5_b	∞_a	∞_a	∞_a

Algoritmo de Dijkstra

El algoritmo de Dijkstra permite calcular el camino más corto desde un origen hacia los nodos restantes de un grafo.

Ejemplo: Calculemos el camino más corto desde a hacia el resto de los nodos.

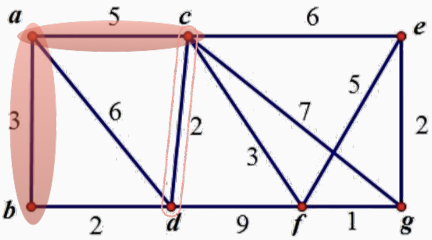


V	a	b	c	d	e	f	g
a	0_a	3_a	5_a	6_a	∞_a	∞_a	∞_a
b	0_a	3_a	5_a	5_b	∞_a	∞_a	∞_a
c	0_a	3_a	5_a				

Algoritmo de Dijkstra

El algoritmo de Dijkstra permite calcular el camino más corto desde un origen hacia los nodos restantes de un grafo.

Ejemplo: Calculemos el camino más corto desde a hacia el resto de los nodos.

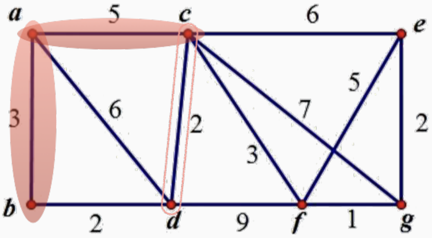


V	a	b	c	d	e	f	g
a	0_a	3_a	5_a	6_a	∞_a	∞_a	∞_a
b	0_a	3_a	5_a	5_b	∞_a	∞_a	∞_a
c	0_a	3_a	5_a				

Algoritmo de Dijkstra

El algoritmo de Dijkstra permite calcular el camino más corto desde un origen hacia los nodos restantes de un grafo.

Ejemplo: Calculemos el camino más corto desde a hacia el resto de los nodos.

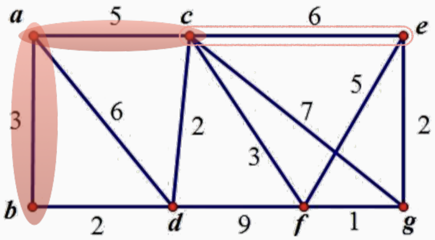


V	a	b	c	d	e	f	g
a	0_a	3_a	5_a	6_a	∞_a	∞_a	∞_a
b	0_a	3_a	5_a	5_b	∞_a	∞_a	∞_a
c	0_a	3_a	5_a	5_b			

Algoritmo de Dijkstra

El algoritmo de Dijkstra permite calcular el camino más corto desde un origen hacia los nodos restantes de un grafo.

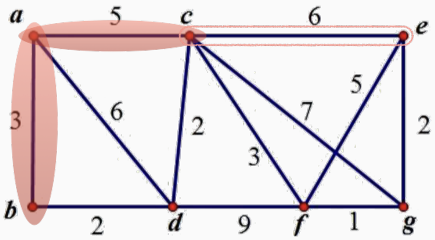
Ejemplo: Calculemos el camino más corto desde a hacia el resto de los nodos.



V	a	b	c	d	e	f	g
a	0_a	3_a	5_a	6_a	∞_a	∞_a	∞_a
b	0_a	3_a	5_a	5_b	∞_a	∞_a	∞_a
c	0_a	3_a	5_a	5_b			

El algoritmo de Dijkstra permite calcular el camino más corto desde un origen hacia los nodos restantes de un grafo.

Ejemplo: Calculemos el camino más corto desde a hacia el resto de los nodos.

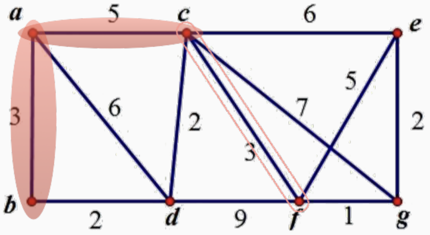


V	a	b	c	d	e	f	g
a	0_a	3_a	5_a	6_a	∞_a	∞_a	∞_a
b	0_a	3_a	5_a	5_b	∞_a	∞_a	∞_a
c	0_a	3_a	5_a	5_b	11_c		

Algoritmo de Dijkstra

El algoritmo de Dijkstra permite calcular el camino más corto desde un origen hacia los nodos restantes de un grafo.

Ejemplo: Calculemos el camino más corto desde a hacia el resto de los nodos.

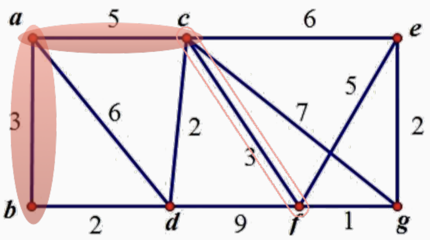


V	a	b	c	d	e	f	g
a	0_a	3_a	5_a	6_a	∞_a	∞_a	∞_a
b	0_a	3_a	5_a	5_b	∞_a	∞_a	∞_a
c	0_a	3_a	5_a	5_b	11_c		

Algoritmo de Dijkstra

El algoritmo de Dijkstra permite calcular el camino más corto desde un origen hacia los nodos restantes de un grafo.

Ejemplo: Calculemos el camino más corto desde a hacia el resto de los nodos.

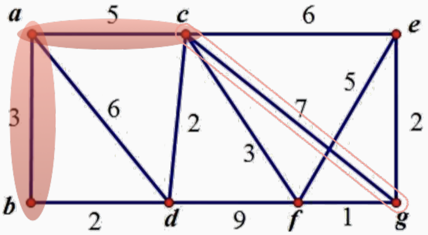


V	a	b	c	d	e	f	g
a	0_a	3_a	5_a	6_a	∞_a	∞_a	∞_a
b	0_a	3_a	5_a	5_b	∞_a	∞_a	∞_a
c	0_a	3_a	5_a	5_b	11_c	8_c	

Algoritmo de Dijkstra

El algoritmo de Dijkstra permite calcular el camino más corto desde un origen hacia los nodos restantes de un grafo.

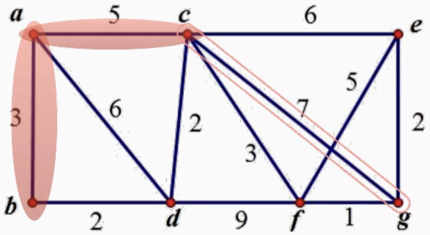
Ejemplo: Calculemos el camino más corto desde a hacia el resto de los nodos.



V	a	b	c	d	e	f	g
a	0_a	3_a	5_a	6_a	∞_a	∞_a	∞_a
b	0_a	3_a	5_a	5_b	∞_a	∞_a	∞_a
c	0_a	3_a	5_a	5_b	11_c	8_c	

El algoritmo de Dijkstra permite calcular el camino más corto desde un origen hacia los nodos restantes de un grafo.

Ejemplo: Calculemos el camino más corto desde a hacia el resto de los nodos.

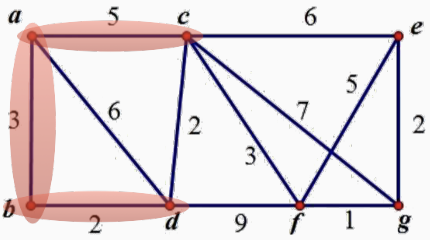


V	a	b	c	d	e	f	g
a	0_a	3_a	5_a	6_a	∞_a	∞_a	∞_a
b	0_a	3_a	5_a	5_b	∞_a	∞_a	∞_a
c	0_a	3_a	5_a	5_b	11_c	8_c	12_c

Algoritmo de Dijkstra

El algoritmo de Dijkstra permite calcular el camino más corto desde un origen hacia los nodos restantes de un grafo.

Ejemplo: Calculemos el camino más corto desde a hacia el resto de los nodos.

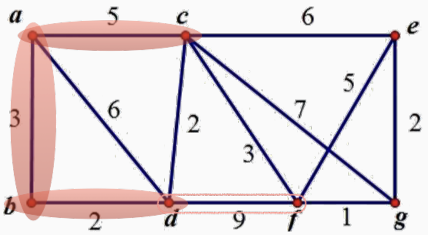


V	a	b	c	d	e	f	g
a	0_a	3_a	5_a	6_a	∞_a	∞_a	∞_a
b	0_a	3_a	5_a	5_b	∞_a	∞_a	∞_a
c	0_a	3_a	5_a	5_b	11_c	8_c	12_c
d	0_a	3_a	5_a	5_b			

Algoritmo de Dijkstra

El algoritmo de Dijkstra permite calcular el camino más corto desde un origen hacia los nodos restantes de un grafo.

Ejemplo: Calculemos el camino más corto desde a hacia el resto de los nodos.

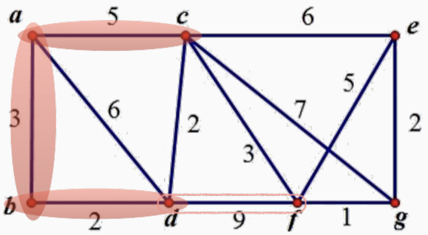


V	a	b	c	d	e	f	g
a	0_a	3_a	5_a	6_a	∞_a	∞_a	∞_a
b	0_a	3_a	5_a	5_b	∞_a	∞_a	∞_a
c	0_a	3_a	5_a	5_b	11_c	8_c	12_c
d	0_a	3_a	5_a	5_b			

Algoritmo de Dijkstra

El algoritmo de Dijkstra permite calcular el camino más corto desde un origen hacia los nodos restantes de un grafo.

Ejemplo: Calculemos el camino más corto desde a hacia el resto de los nodos.

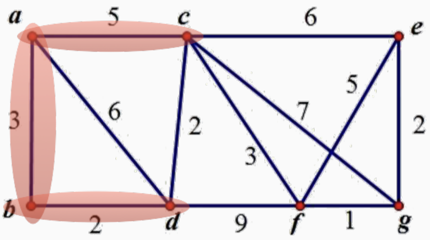


V	a	b	c	d	e	f	g
a	0_a	3_a	5_a	6_a	∞_a	∞_a	∞_a
b	0_a	3_a	5_a	5_b	∞_a	∞_a	∞_a
c	0_a	3_a	5_a	5_b	11_c	8_c	12_c
d	0_a	3_a	5_a	5_b		8_c	

Algoritmo de Dijkstra

El algoritmo de Dijkstra permite calcular el camino más corto desde un origen hacia los nodos restantes de un grafo.

Ejemplo: Calculemos el camino más corto desde a hacia el resto de los nodos.

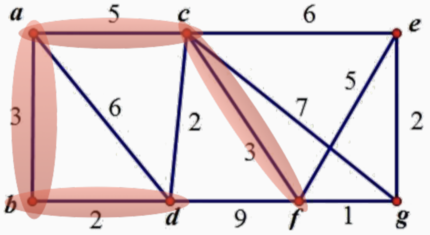


V	a	b	c	d	e	f	g
a	0_a	3_a	5_a	6_a	∞_a	∞_a	∞_a
b	0_a	3_a	5_a	5_b	∞_a	∞_a	∞_a
c	0_a	3_a	5_a	5_b	11_c	8_c	12_c
d	0_a	3_a	5_a	5_b	11_c	8_c	12_c

Algoritmo de Dijkstra

El algoritmo de Dijkstra permite calcular el camino más corto desde un origen hacia los nodos restantes de un grafo.

Ejemplo: Calculemos el camino más corto desde a hacia el resto de los nodos.

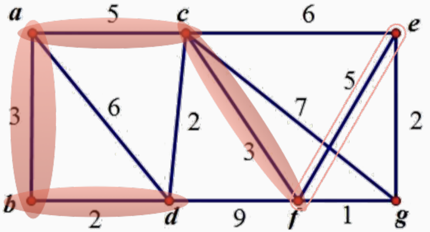


V	a	b	c	d	e	f	g
a	0_a	3_a	5_a	6_a	∞_a	∞_a	∞_a
b	0_a	3_a	5_a	5_b	∞_a	∞_a	∞_a
c	0_a	3_a	5_a	5_b	11_c	8_c	12_c
d	0_a	3_a	5_a	5_b	11_c	8_c	12_c
f	0_a	3_a	5_a	5_b		8_c	

Algoritmo de Dijkstra

El algoritmo de Dijkstra permite calcular el camino más corto desde un origen hacia los nodos restantes de un grafo.

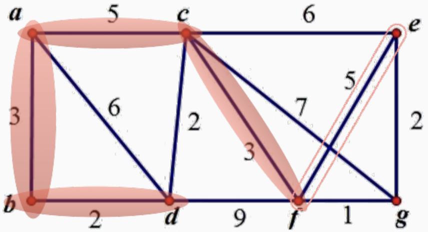
Ejemplo: Calculemos el camino más corto desde *a* hacia el resto de los nodos.



V	a	b	c	d	e	f	g
a	0_a	3_a	5_a	6_a	∞_a	∞_a	∞_a
b	0_a	3_a	5_a	5_b	∞_a	∞_a	∞_a
c	0_a	3_a	5_a	5_b	11_c	8_c	12_c
d	0_a	3_a	5_a	5_b	11_c	8_c	12_c
f	0_a	3_a	5_a	5_b		8_c	

El algoritmo de Dijkstra permite calcular el camino más corto desde un origen hacia los nodos restantes de un grafo.

Ejemplo: Calculemos el camino más corto desde a hacia el resto de los nodos.

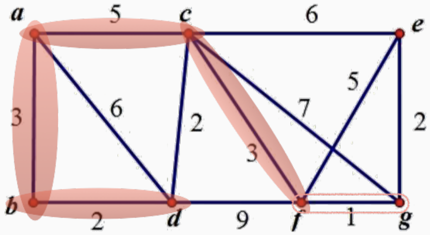


V	a	b	c	d	e	f	g
a	0_a	3_a	5_a	6_a	∞_a	∞_a	∞_a
b	0_a	3_a	5_a	5_b	∞_a	∞_a	∞_a
c	0_a	3_a	5_a	5_b	11_c	8_c	12_c
d	0_a	3_a	5_a	5_b	11_c	8_c	12_c
f	0_a	3_a	5_a	5_b	11_c	8_c	

Algoritmo de Dijkstra

El algoritmo de Dijkstra permite calcular el camino más corto desde un origen hacia los nodos restantes de un grafo.

Ejemplo: Calculemos el camino más corto desde a hacia el resto de los nodos.

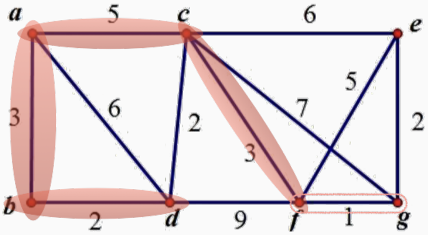


V	a	b	c	d	e	f	g
a	0_a	3_a	5_a	6_a	∞_a	∞_a	∞_a
b	0_a	3_a	5_a	5_b	∞_a	∞_a	∞_a
c	0_a	3_a	5_a	5_b	11_c	8_c	12_c
d	0_a	3_a	5_a	5_b	11_c	8_c	12_c
f	0_a	3_a	5_a	5_b	11_c	8_c	

Algoritmo de Dijkstra

El algoritmo de Dijkstra permite calcular el camino más corto desde un origen hacia los nodos restantes de un grafo.

Ejemplo: Calculemos el camino más corto desde *a* hacia el resto de los nodos.

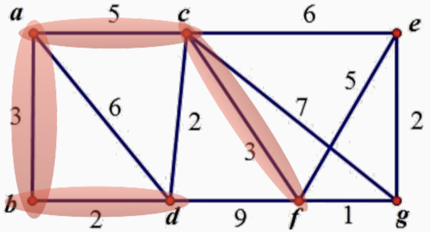


V	a	b	c	d	e	f	g
a	0_a	3_a	5_a	6_a	∞_a	∞_a	∞_a
b	0_a	3_a	5_a	5_b	∞_a	∞_a	∞_a
c	0_a	3_a	5_a	5_b	11_c	8_c	12_c
d	0_a	3_a	5_a	5_b	11_c	8_c	12_c
f	0_a	3_a	5_a	5_b	11_c	8_c	9_f

Algoritmo de Dijkstra

El algoritmo de Dijkstra permite calcular el camino más corto desde un origen hacia los nodos restantes de un grafo.

Ejemplo: Calculemos el camino más corto desde a hacia el resto de los nodos.

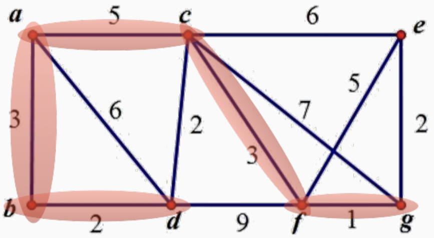


V	a	b	c	d	e	f	g
a	0_a	3_a	5_a	6_a	∞_a	∞_a	∞_a
b	0_a	3_a	5_a	5_b	∞_a	∞_a	∞_a
c	0_a	3_a	5_a	5_b	11_c	8_c	12_c
d	0_a	3_a	5_a	5_b	11_c	8_c	12_c
f	0_a	3_a	5_a	5_b	11_c	8_c	9_f

Algoritmo de Dijkstra

El algoritmo de Dijkstra permite calcular el camino más corto desde un origen hacia los nodos restantes de un grafo.

Ejemplo: Calculemos el camino más corto desde a hacia el resto de los nodos.

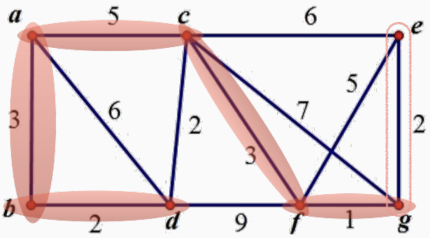


V	a	b	c	d	e	f	g
a	0_a	3_a	5_a	6_a	∞_a	∞_a	∞_a
b	0_a	3_a	5_a	5_b	∞_a	∞_a	∞_a
c	0_a	3_a	5_a	5_b	11_c	8_c	12_c
d	0_a	3_a	5_a	5_b	11_c	8_c	12_c
f	0_a	3_a	5_a	5_b	11_c	8_c	9_f
g	0_a	3_a	5_a	5_b		8_c	9_f

Algoritmo de Dijkstra

El algoritmo de Dijkstra permite calcular el camino más corto desde un origen hacia los nodos restantes de un grafo.

Ejemplo: Calculemos el camino más corto desde a hacia el resto de los nodos.

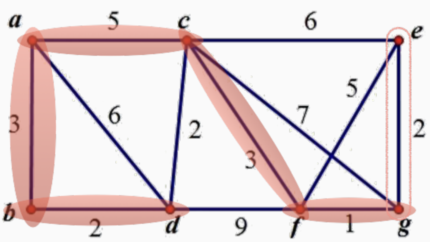


V	a	b	c	d	e	f	g
a	0_a	3_a	5_a	6_a	∞_a	∞_a	∞_a
b	0_a	3_a	5_a	5_b	∞_a	∞_a	∞_a
c	0_a	3_a	5_a	5_b	11_c	8_c	12_c
d	0_a	3_a	5_a	5_b	11_c	8_c	12_c
f	0_a	3_a	5_a	5_b	11_c	8_c	9_f
g	0_a	3_a	5_a	5_b		8_c	9_f

Algoritmo de Dijkstra

El algoritmo de Dijkstra permite calcular el camino más corto desde un origen hacia los nodos restantes de un grafo.

Ejemplo: Calculemos el camino más corto desde a hacia el resto de los nodos.

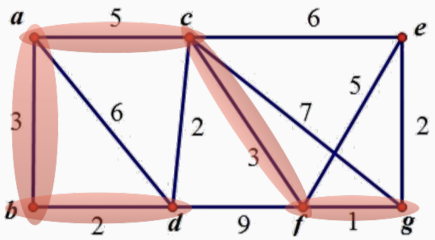


V	a	b	c	d	e	f	g
a	0_a	3_a	5_a	6_a	∞_a	∞_a	∞_a
b	0_a	3_a	5_a	5_b	∞_a	∞_a	∞_a
c	0_a	3_a	5_a	5_b	11_c	8_c	12_c
d	0_a	3_a	5_a	5_b	11_c	8_c	12_c
f	0_a	3_a	5_a	5_b	11_c	8_c	9_f
g	0_a	3_a	5_a	5_b	11_c	8_c	9_f

Algoritmo de Dijkstra

El algoritmo de Dijkstra permite calcular el camino más corto desde un origen hacia los nodos restantes de un grafo.

Ejemplo: Calculemos el camino más corto desde *a* hacia el resto de los nodos.

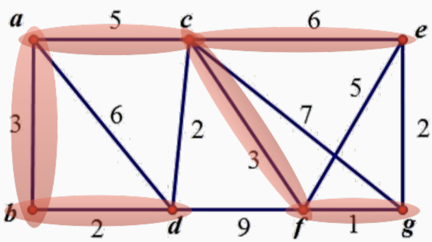


V	a	b	c	d	e	f	g
a	0_a	3_a	5_a	6_a	∞_a	∞_a	∞_a
b	0_a	3_a	5_a	5_b	∞_a	∞_a	∞_a
c	0_a	3_a	5_a	5_b	11_c	8_c	12_c
d	0_a	3_a	5_a	5_b	11_c	8_c	12_c
f	0_a	3_a	5_a	5_b	11_c	8_c	9_f
g	0_a	3_a	5_a	5_b	11_c	8_c	9_f

Algoritmo de Dijkstra

El algoritmo de Dijkstra permite calcular el camino más corto desde un origen hacia los nodos restantes de un grafo.

Ejemplo: Calculemos el camino más corto desde a hacia el resto de los nodos.

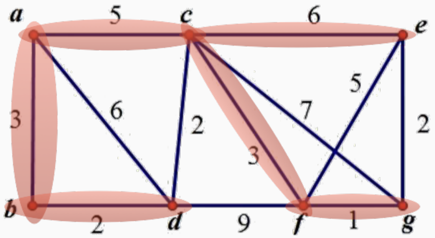


V	a	b	c	d	e	f	g
a	0_a	3_a	5_a	6_a	∞_a	∞_a	∞_a
b	0_a	3_a	5_a	5_b	∞_a	∞_a	∞_a
c	0_a	3_a	5_a	5_b	11_c	8_c	12_c
d	0_a	3_a	5_a	5_b	11_c	8_c	12_c
f	0_a	3_a	5_a	5_b	11_c	8_c	9_f
g	0_a	3_a	5_a	5_b	11_c	8_c	9_f
e	0_a	3_a	5_a	5_b	11_c	8_c	9_f

Algoritmo de Dijkstra

El algoritmo de Dijkstra permite calcular el camino más corto desde un origen hacia los nodos restantes de un grafo.

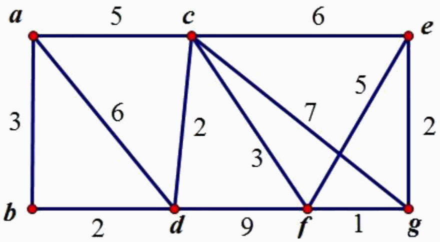
Ejemplo: Calculemos el camino más corto desde a hacia el resto de los nodos.



V	a	b	c	d	e	f	g
a	0_a	3_a	5_a	6_a	∞_a	∞_a	∞_a
b	0_a	3_a	5_a	5_b	∞_a	∞_a	∞_a
c	0_a	3_a	5_a	5_b	11_c	8_c	12_c
d	0_a	3_a	5_a	5_b	11_c	8_c	12_c
f	0_a	3_a	5_a	5_b	11_c	8_c	9_f
g	0_a	3_a	5_a	5_b	11_c	8_c	9_f
e	0_a	3_a	5_a	5_b	11_c	8_c	9_f

Ver demo del algoritmo en `https://youtu.be/0nVYi3o161A`

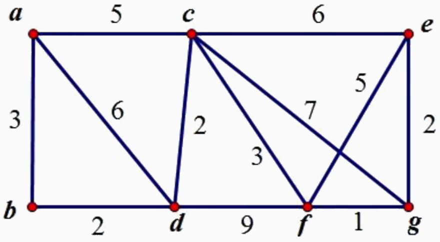
Algoritmo de Dijkstra



V	a	b	c	d	e	f	g
a	0_a	3_a	5_a	6_a	∞_a	∞_a	∞_a
b	0_a	3_a	5_a	5_b	∞_a	∞_a	∞_a
c	0_a	3_a	5_a	5_b	11_c	8_c	12_c
d	0_a	3_a	5_a	5_b	11_c	8_c	12_c
f	0_a	3_a	5_a	5_b	11_c	8_c	9_f
g	0_a	3_a	5_a	5_b	11_c	8_c	9_f
e	0_a	3_a	5_a	5_b	11_c	8_c	9_f

- La última fila lista la longitud de los caminos más cortos desde *a* al resto de los nodos.

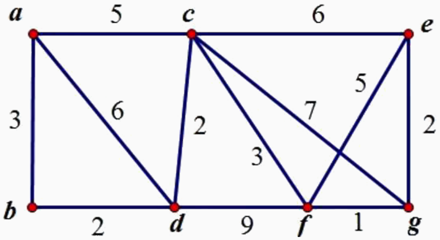
Algoritmo de Dijkstra



V	a	b	c	d	e	f	g
a	0_a	3_a	5_a	6_a	∞_a	∞_a	∞_a
b	0_a	3_a	5_a	5_b	∞_a	∞_a	∞_a
c	0_a	3_a	5_a	5_b	11_c	8_c	12_c
d	0_a	3_a	5_a	5_b	11_c	8_c	12_c
f	0_a	3_a	5_a	5_b	11_c	8_c	9_f
g	0_a	3_a	5_a	5_b	11_c	8_c	9_f
e	0_a	3_a	5_a	5_b	11_c	8_c	9_f

- Si quiero encontrar el camino más corto desde a hacia un nodo destino específico, digamos f, puedo detenerme apenas f esté en mi conjunto coloreado. En este caso, al comenzar la 5º iteración.

Algoritmo de Dijkstra



V	a	b	c	d	e	f	g
a	0_a	3_a	5_a	6_a	∞_a	∞_a	∞_a
b	0_a	3_a	5_a	5_b	∞_a	∞_a	∞_a
c	0_a	3_a	5_a	5_b	11_c	8_c	12_c
d	0_a	3_a	5_a	5_b	11_c	8_c	12_c
f	0_a	3_a	5_a	5_b	11_c	8_c	9_f
g	0_a	3_a	5_a	5_b	11_c	8_c	9_f
e	0_a	3_a	5_a	5_b	11_c	8_c	9_f

- Puedo reconstruir dicho camino más corto, de manera recursiva, hacia atrás, mirando los subíndices:

$$f \leftarrow c \leftarrow a$$

Algoritmo de Dijkstra

```
function Dijkstra(Grafo, fuente):

    Q ← ∅

    for each vertex v in Grafo:              // Inicialización
        dist[v] ← ∞                          // Distancia desconocida desde la fuente a v
        prev[v] ← undef                      // Nodo previo en el camino optimal desde la fuente
        add v to Q                           // Todos los nodos inicialmente están en Q  (no han sido visitados)

    dist[source] ← 0                         // Distancia desde la fuente a la fuente

    while Q is not empty:
        u ← vertex in Q with min dist[u]     // El nodo con la menor distancia será seleccionado primero

        remove u from Q

        for each neighbor v of u in Q:
            alt ← dist[u] + length(u, v)
            if alt < dist[v]:                // Se encontró un camino
                dist[v] ← alt                //  más corto a v
                prev[v] ← u

    return dist[], prev[]
```

Q representa el *complemento* del conjunto de nodos coloreados

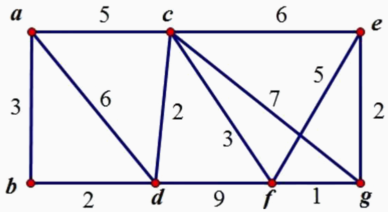
Observaciones

- El algoritmo funciona de la misma manera para grafos dirigidos

Observaciones

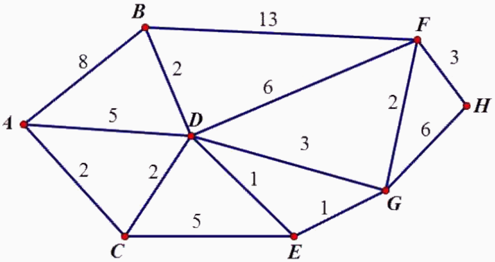
- El algoritmo funciona de la misma manera para grafos dirigidos
- Si un grafo tiene lazos, puedo eliminarlos antes de aplicar el algoritmo. De la misma manera, si un grafo tiene aristas paralelas, puedo quedarme sólo con la de menor peso.

Algoritmo de Dijkstra

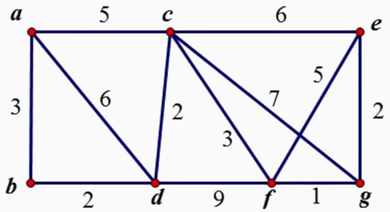


V	a	b	c	d	e	f	g
a	0_a	3_a	5_a	6_a	∞_a	∞_a	∞_a
b	0_a	3_a	5_a	5_b	∞_a	∞_a	∞_a
c	0_a	3_a	5_a	5_b	11_c	8_c	12_c
d	0_a	3_a	5_a	5_b	11_c	8_c	12_c
f	0_a	3_a	5_a	5_b	11_c	8_c	9_f
g	0_a	3_a	5_a	5_b	11_c	8_c	9_f
e	0_a	3_a	5_a	5_b	11_c	8_c	9_f

Ejercicio: Aplicar el algoritmo de Dijkstra para calcular el camino más corto desde A hacia E.

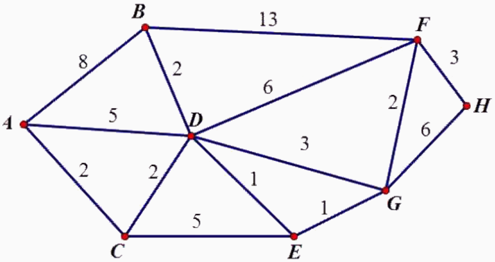


Algoritmo de Dijkstra



V	a	b	c	d	e	f	g
a	0_a	3_a	5_a	6_a	∞_a	∞_a	∞_a
b	0_a	3_a	5_a	5_b	∞_a	∞_a	∞_a
c	0_a	3_a	5_a	5_b	11_c	8_c	12_c
d	0_a	3_a	5_a	5_b	11_c	8_c	12_c
f	0_a	3_a	5_a	5_b	11_c	8_c	9_f
g	0_a	3_a	5_a	5_b	11_c	8_c	9_f
e	0_a	3_a	5_a	5_b	11_c	8_c	9_f

Ejercicio: Aplicar el algoritmo de Dijkstra para calcular el camino más corto desde A hacia E.



Solución:

V	A	B	C	D	E	F	G	H
A	$\boxed{0_A}$	8_A	2_A	5_A	∞	∞	∞	∞
C		8_A	$\boxed{2_A}$	4_C	7_C	∞	∞	∞
D		6_D		$\boxed{4_C}$	5_D	10_D	7_D	∞
					$\boxed{5_D}$			

Demo en vivo del algoritmo

Para ver el algoritmo en funcionamiento, sobre un grafo ingresado por el usuario, ver

`https://www-m9.ma.tum.de/graph-algorithms/spp-dijkstra/index_en.html`

Colorabilidad de grafos

Colorabilidad

Definición

Una coloración de un grafo simple G es una asignación de colores a los vértices de G de tal forma que vértices adyacentes reciban colores distintos.

Colorabilidad

Definición

Una coloración de un grafo simple G es una asignación de colores a los vértices de G de tal forma que vértices adyacentes reciban colores distintos.

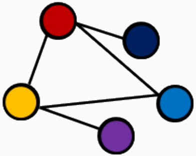
Por supuesto, todo grafo simple de n vértices puede ser coloreado con n colores, pero "la mayoría" de los grafos pueden ser coloreados con menos colores.

Colorabilidad

Definición

Una coloración de un grafo simple G es una asignación de colores a los vértices de G de tal forma que vértices adyacentes reciban colores distintos.

Por supuesto, todo grafo simple de n vértices puede ser coloreado con n colores, pero "la mayoría" de los grafos pueden ser coloreados con menos colores.

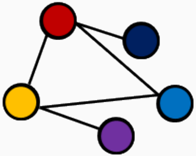


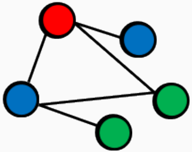
Colorabilidad

Definición

Una coloración de un grafo simple G es una asignación de colores a los vértices de G de tal forma que vértices adyacentes reciban colores distintos.

Por supuesto, todo grafo simple de n vértices puede ser coloreado con n colores, pero "la mayoría" de los grafos pueden ser coloreados con menos colores.





Número cromático

Definición

El número cromático de un grafo simple G, notado $\chi(G)$, es el menor número de colores que se necesitan para colorear a G.

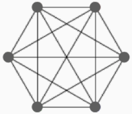
Número cromático

Definición

El número cromático de un grafo simple G, notado $\chi(G)$, es el menor número de colores que se necesitan para colorear a G.

Ejemplos:

$\chi(K_6) =$



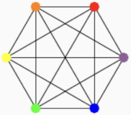
Número cromático

Definición

El número cromático de un grafo simple G, notado $\chi(G)$, es el menor número de colores que se necesitan para colorear a G.

Ejemplos:

$\chi(K_6) = 6$



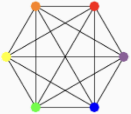
Número cromático

Definición

El número cromático de un grafo simple G, notado $\chi(G)$, es el menor número de colores que se necesitan para colorear a G.

Ejemplos:

$\chi(K_6) = 6$



$\chi(S_6) =$

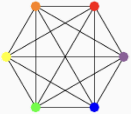
Número cromático

Definición

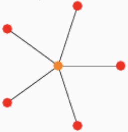
El número cromático de un grafo simple G, notado $\chi(G)$, es el menor número de colores que se necesitan para colorear a G.

Ejemplos:

$\chi(K_6) = 6$



$\chi(S_6) = 6$



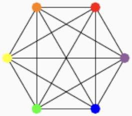
Número cromático

Definición

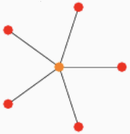
El número cromático de un grafo simple G, notado $\chi(G)$, es el menor número de colores que se necesitan para colorear a G.

Ejemplos:

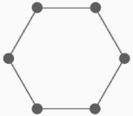
$\chi(K_6) = 6$



$\chi(S_6) = 6$



$\chi(C_6) =$



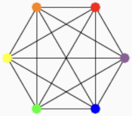
Número cromático

Definición

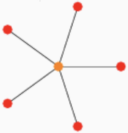
El número cromático de un grafo simple G, notado $\chi(G)$, es el menor número de colores que se necesitan para colorear a G.

Ejemplos:

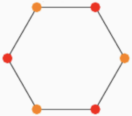
$\chi(K_6) = 6$



$\chi(S_6) = 6$



$\chi(C_6) = 2$



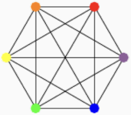
Número cromático

Definición

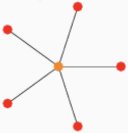
El número cromático de un grafo simple G, notado $\chi(G)$, es el menor número de colores que se necesitan para colorear a G.

Ejemplos:

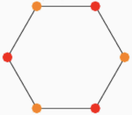
$\chi(K_6) = 6$



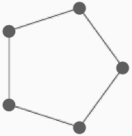
$\chi(S_6) = 6$



$\chi(C_6) = 2$



$\chi(C_5) =$



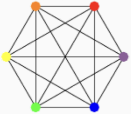
Número cromático

Definición

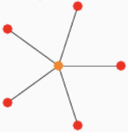
El número cromático de un grafo simple G, notado $\chi(G)$, es el menor número de colores que se necesitan para colorear a G.

Ejemplos:

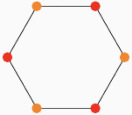
$\chi(K_6) = 6$



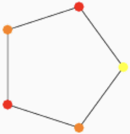
$\chi(S_6) = 6$



$\chi(C_6) = 2$



$\chi(C_5) = 3$



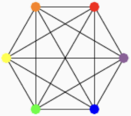
Número cromático

Definición

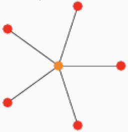
El número cromático de un grafo simple G, notado $\chi(G)$, es el menor número de colores que se necesitan para colorear a G.

Ejemplos:

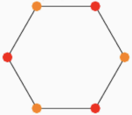
$\chi(K_6) = 6$



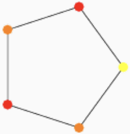
$\chi(S_6) = 6$



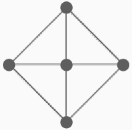
$\chi(C_6) = 2$



$\chi(C_5) = 3$



$\chi(W_5) =$



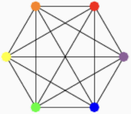
Número cromático

Definición

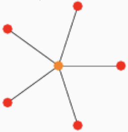
El número cromático de un grafo simple G, notado $\chi(G)$, es el menor número de colores que se necesitan para colorear a G.

Ejemplos:

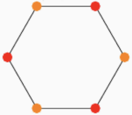
$\chi(K_6) = 6$



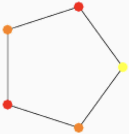
$\chi(S_6) = 6$



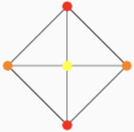
$\chi(C_6) = 2$



$\chi(C_5) = 3$



$\chi(W_5) = 3$



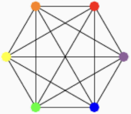
Número cromático

Definición

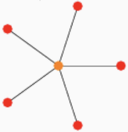
El número cromático de un grafo simple G, notado $\chi(G)$, es el menor número de colores que se necesitan para colorear a G.

Ejemplos:

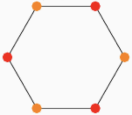
$\chi(K_6) = 6$



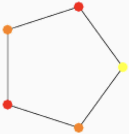
$\chi(S_6) = 6$



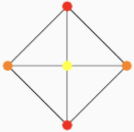
$\chi(C_6) = 2$



$\chi(C_5) = 3$



$\chi(W_5) = 3$



$\chi(W_6) =$

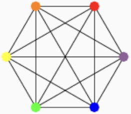
Número cromático

Definición

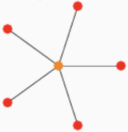
El número cromático de un grafo simple G, notado $\chi(G)$, es el menor número de colores que se necesitan para colorear a G.

Ejemplos:

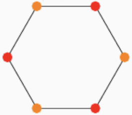
$\chi(K_6) = 6$



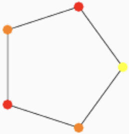
$\chi(S_6) = 6$



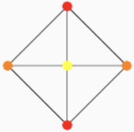
$\chi(C_6) = 2$



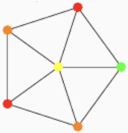
$\chi(C_5) = 3$



$\chi(W_5) = 3$



$\chi(W_6) = 4$



Ejercicios: Número cromático

Ejercicio: Demuestre que $\chi(K_n) = n$ para todo $n \geq 1$.

Ejercicios: Número cromático

Ejercicio: Demuestre que $\chi(K_n) = n$ para todo $n \geq 1$.

Ejercicio: Demuestre que $\chi(C_n) = 2$ si n es (entero positivo) par, e igual a 3 si n es impar.

Ejercicios: Número cromático

Ejercicio: Demuestre que $\chi(K_n) = n$ para todo $n \geq 1$.

Ejercicio: Demuestre que $\chi(C_n) = 2$ si n es (entero positivo) par, e igual a 3 si n es impar.

Ejercicio: Enuncie y pruebe un resultado similar al ejercicio anterior sobre $\chi(W_n)$.

Ejercicios: Número cromático

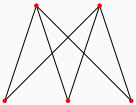
Ejercicio: Demuestre que $\chi(K_n) = n$ para todo $n \geq 1$.

Ejercicio: Demuestre que $\chi(C_n) = 2$ si n es (entero positivo) par, e igual a 3 si n es impar.

Ejercicio: Enuncie y pruebe un resultado similar al ejercicio anterior sobre $\chi(W_n)$.

Ejercicio: Calcule $\chi(K_{n,m})$ para $n, m \geq 1$ arbitrarios, donde $K_{n,m} = (V, E)$ es el grafo bipartito completo, esto es, sus nodos $V = V_1 \bigcup V_2$ están particionados en dos conjuntos V_1 y V_2, con $|V_1| = n$, $|V_2| = m$, tales que existe un arco $(u, v) \in E$ si y sólo si $u \in V_1$ y $v \in V_2$.

Ejemplo: $K_{2,3}$:

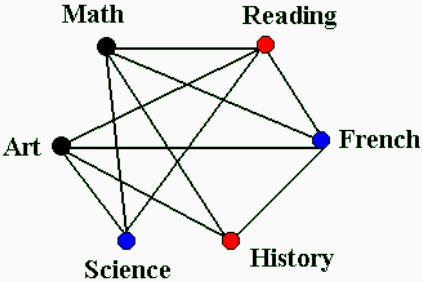


Aplicación: Programación de exámenes

La Escuela asigna un bloque de tres horas para el examen final de cada ramo. ¿Cuál es la menor cantidad de bloques que puede asignar, de tal manera que los alumnos no tengan tope entre los examenes de los distintos ramos en los que están inscriptos?

Aplicación: Programación de exámenes

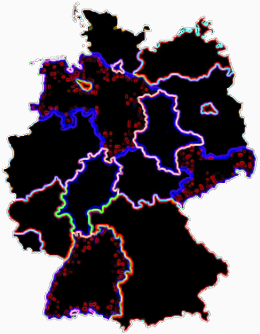
La Escuela asigna un bloque de tres horas para el examen final de cada ramo. ¿Cuál es la menor cantidad de bloques que puede asignar, de tal manera que los alumnos no tengan tope entre los examenes de los distintos ramos en los que están inscriptos?



La respuesta está dada por el número cromático del grafo de incompatibilidad, donde los nodos representan los ramos, y dos nodos son adyacentes ssi tienen algún alumno en común.

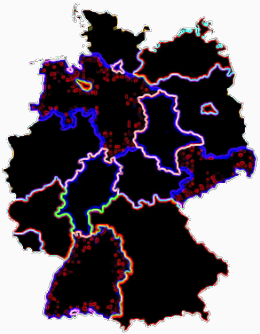
Aplicación: Pintando mapas

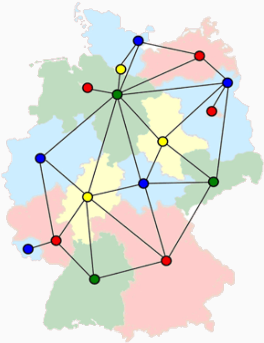
¿Cuál es la mínima cantidad de colores que son necesarios para pintar el siguiente mapa si las regiones lindantes deben tener distinto color?



Aplicación: Pintando mapas

¿Cuál es la mínima cantidad de colores que son necesarios para pintar el siguiente mapa si las regiones lindantes deben tener distinto color?

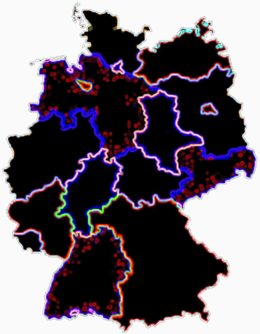


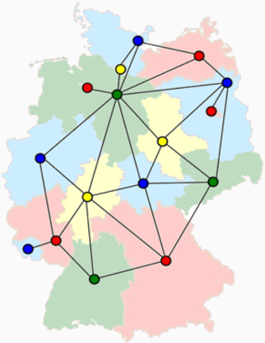


La respuesta está dada por el número cromático del grafo de la derecha, donde cada vértice representa una región y dos vertices están conectados si las regiones que representan son limítrofes.

Aplicación: Pintando mapas

¿Cuál es la mínima cantidad de colores que son necesarios para pintar el siguiente mapa si las regiones lindantes deben tener distinto color?





La respuesta está dada por el número cromático del grafo de la derecha, donde cada vértice representa una región y dos vertices están conectados si las regiones que representan son limítrofes. **Nota:** Todos los grafos asociados a mapas son planares, esto es, puede dibujarse sin cruzar las aristas.

Teorema de los 4 colores

El mapa anterior se puede pintar usando sólo 4 colores. Se puede demostrar que eso vale, en general, para cualquier mapa.

Teorema de los 4 colores

El mapa anterior se puede pintar usando sólo 4 colores. Se puede demostrar que eso vale, en general, para cualquier mapa.

La clave de la prueba está en observar que el grafo que construimos es siempre conexo y planar; luego el resultado se desprende del siguiente teorema:

Teorema de los 4 colores (Appel y Haken, 1976)

Sea G un grafo simple, conexo y planar. Entonces $\chi(G) \leq 4$.

Teorema de los 4 colores y demostraciones automatizadas

Teorema de los 4 colores (Appel y Haken, 1976)

Sea G un grafo simple, conexo y planar. Entonces $\chi(G) \leq 4$.

Datos interesantes:

Teorema de los 4 colores (Appel y Haken, 1976)

Sea G un grafo simple, conexo y planar. Entonces $\chi(G) \leq 4$.

Datos interesantes:

- Teorema conjeturado inicialmente en 1850.

Teorema de los 4 colores y demostraciones automatizadas

Teorema de los 4 colores (Appel y Haken, 1976)

Sea G un grafo simple, conexo y planar. Entonces $\chi(G) \leq 4$.

Datos interesantes:

- Teorema conjeturado inicialmente en 1850.
- Demostración original (Appel y Haken, 1976) requirió verificar múltiples casos por medio de un computador: requería realizar un número de verificaciones demasiado grande como para ser ejecutado “a mano”. Fue la primera demostración “matemática” que requería inevitablemente un computador. Fue muy controversial, incluso no fue aceptada inicialmente.

Teorema de los 4 colores y demostraciones automatizadas

Teorema de los 4 colores (Appel y Haken, 1976)

Sea G un grafo simple, conexo y planar. Entonces $\chi(G) \leq 4$.

Datos interesantes:

- Teorema conjeturado inicialmente en 1850.
- Demostración original (Appel y Haken, 1976) requirió verificar múltiples casos por medio de un computador: requería realizar un número de verificaciones demasiado grande como para ser ejecutado "a mano". Fue la primera demostración "matemática" que requería inevitablemente un computador. Fue muy controversial, incluso no fue aceptada inicialmente.
- Mejorada en 1995 por Saunders, Seymour, y Thomas. Pero aún requiere un computador.

Teorema de los 4 colores y demostraciones automatizadas

Teorema de los 4 colores (Appel y Haken, 1976)

Sea G un grafo simple, conexo y planar. Entonces $\chi(G) \leq 4$.

Datos interesantes:

- Teorema conjeturado inicialmente en 1850.
- Demostración original (Appel y Haken, 1976) requirió verificar múltiples casos por medio de un computador: requería realizar un número de verificaciones demasiado grande como para ser ejecutado "a mano". Fue la primera demostración "matemática" que requería inevitablemente un computador. Fue muy controversial, incluso no fue aceptada inicialmente.
- Mejorada en 1995 por Saunders, Seymour, y Thomas. Pero aún requiere un computador.
- El 2005 (Gonthier) toda la demostración fue unificada en un argumento formal, demostrado con el formalismo de demostración computacional denominado Coq (similar a Lean).

Referencias

Artículos recomendados:

- "The Four Color Theorem", Yuriy Brun, Undergraduate Journal of Mathematics, May 2002, pp. 21–28.
  `https://people.cs.umass.edu/~brun/pubs/pubs/Brun02four-color.pdf`
  Explica el problema de los 4 colores y demuestra un par de variantes simples.

Referencias

Artículos recomendados:

- "The Four Color Theorem", Yuriy Brun, Undergraduate Journal of Mathematics, May 2002, pp. 21–28.
  https://people.cs.umass.edu/~brun/pubs/pubs/Brun02four-color.pdf
  Explica el problema de los 4 colores y demuestra un par de variantes simples.
- "Four Colors Suffice: How the Map Problem Was Solved", Robin Wilson. Princeton University Press, 2002. Disponible comercialmente (amazon kindle).
  Libro de divulgación que cuenta la historia de cómo fue demostrado el teorema de los 4 colores.
  Presentación sobre el libro disponible: https://math.illinois.edu/system/files/inline-files/wilson-slides-11-2-17.pdf.

Referencias

Artículos recomendados:

- "The Four Color Theorem", Yuriy Brun, Undergraduate Journal of Mathematics, May 2002, pp. 21–28.
  `https://people.cs.umass.edu/~brun/pubs/pubs/Brun02four-color.pdf`
  Explica el problema de los 4 colores y demuestra un par de variantes simples.
- "Four Colors Suffice: How the Map Problem Was Solved", Robin Wilson. Princeton University Press, 2002. Disponible comercialmente (amazon kindle).
  Libro de divulgación que cuenta la historia de cómo fue demostrado el teorema de los 4 colores.
  Presentación sobre el libro disponible: `https://math.illinois.edu/system/files/inline-files/wilson-slides-11-2-17.pdf`.
- "Formal Proof, The Four-Color Theorem", Georges Gonthier, Notices of the AMS, Volume 55, Number 11. Dic 2008.
  `https://www.ams.org//notices/200811/tx081101382p.pdf`
  Explica más detalles del proceso de formalizar el teorema y su demostración en Coq.