GrafosGreedy

Daniel Penazzi

19 de marzo de 2021

Tabla de Contenidos

1. Greedy (cont.)
   - Error de Greedy
   - Brooks
   - Reordenes
2. 2COLOR

Mas sobre Greedy

- En la primera clase dimos un ejemplo de que Greedy no necesariamente colorea un grafo G con $\chi(G)$ colores.

Mas sobre Greedy

- En la primera clase dimos un ejemplo de que Greedy no necesariamente colorea un grafo G con $\chi(G)$ colores.
- El ejemplo era C_6, que se puede colorear con 2 colores, pero en un cierto orden de los vértices Greedy usa 3 colores.

Mas sobre Greedy

- En la primera clase dimos un ejemplo de que Greedy no necesariamente colorea un grafo G con $\chi(G)$ colores.
- El ejemplo era C_6, que se puede colorear con 2 colores, pero en un cierto orden de los vértices Greedy usa 3 colores.
- En ese ejemplo Greedy le “erró” por uno.

Mas sobre Greedy

- En la primera clase dimos un ejemplo de que Greedy no necesariamente colorea un grafo G con $\chi(G)$ colores.
- El ejemplo era C_6, que se puede colorear con 2 colores, pero en un cierto orden de los vértices Greedy usa 3 colores.
- En ese ejemplo Greedy le “erró” por uno.
- Puede “errarle ” por mas? Que tanto mas?

Mas sobre Greedy

- En la primera clase dimos un ejemplo de que Greedy no necesariamente colorea un grafo G con $\chi(G)$ colores.
- El ejemplo era C_6, que se puede colorear con 2 colores, pero en un cierto orden de los vértices Greedy usa 3 colores.
- En ese ejemplo Greedy le “erró” por uno.
- Puede “errarle ” por mas? Que tanto mas?
- Veamos un ejemplo con n par, $n = 2r$.

Mas sobre Greedy

- En la primera clase dimos un ejemplo de que Greedy no necesariamente colorea un grafo G con $\chi(G)$ colores.
- El ejemplo era C_6, que se puede colorear con 2 colores, pero en un cierto orden de los vértices Greedy usa 3 colores.
- En ese ejemplo Greedy le “erró” por uno.
- Puede “errarle ” por mas? Que tanto mas?
- Veamos un ejemplo con n par, $n = 2r$.
- Tomaremos vertices $v_1, v_2, ..., v_n$.

Mas sobre Greedy

- Los lados son todos los $v_i v_j$ tales que:

Mas sobre Greedy

- Los lados son todos los $v_i v_j$ tales que:
  1. i es impar, j es par y:

Mas sobre Greedy

- Los lados son todos los $v_i v_j$ tales que:
  1. i es impar, j es par y:
  2. $j \neq i + 1$.

Mas sobre Greedy

- Los lados son todos los $v_i v_j$ tales que:
  1. i es impar, j es par y:
  2. $j \neq i + 1$.
- Corremos Greedy en el orden $v_1, v_2, ..., v_n$.

Mas sobre Greedy

- Los lados son todos los $v_i v_j$ tales que:
  1. i es impar, j es par y:
  2. $j \neq i + 1$.
- Corremos Greedy en el orden $v_1, v_2, ..., v_n$.
- Empezamos con $c(v_1) = 0$.

Mas sobre Greedy

- Los lados son todos los v_iv_j tales que:
  1. i es impar, j es par y:
  2. $j \neq i+1$.
- Corremos Greedy en el orden $v_1, v_2, ..., v_n$.
- Empezamos con $c(v_1) = 0$.
- v_2 no tiene vécinos en el conjunto de vértices anteriores a el, pues el único vértice anterior en el orden es v_1.

Mas sobre Greedy

- Los lados son todos los $v_i v_j$ tales que:
  1. i es impar, j es par y:
  2. $j \neq i + 1$.
- Corremos Greedy en el orden $v_1, v_2, ..., v_n$.
- Empezamos con $c(v_1) = 0$.
- v_2 no tiene vécinos en el conjunto de vértices anteriores a el, pues el único vértice anterior en el orden es v_1.
- Asi que Greedy le da el color $c(v_2) = 0$.

Continuando con el ejemplo

- Los vértices anteriores a v_3 son v_1 y v_2, y v_2 es vecino, asi que v_3 no puede tener el color 0. Greedy lo colorea $c(v_3) = 1$.

Continuando con el ejemplo

- Los vértices anteriores a v_3 son v_1 y v_2, y v_2 es vecino, asi que v_3 no puede tener el color 0. Greedy lo colorea $c(v_3) = 1$.
- Los vértices anteriores a v_3 son v_1, v_2 y v_3, y v_1 es el único vecino, asi que v_4 no puede tener el color 0. Greedy lo colorea $c(v_4) = 1$.

Continuando con el ejemplo

- Los vértices anteriores a v_3 son v_1 y v_2, y v_2 es vecino, asi que v_3 no puede tener el color 0. Greedy lo colorea $c(v_3) = 1$.
- Los vértices anteriores a v_3 son v_1, v_2 y v_3, y v_1 es el único vecino, asi que v_4 no puede tener el color 0. Greedy lo colorea $c(v_4) = 1$.
- Los vértices anteriores a v_5 son v_1, v_2, v_3 y v_4, y entre ellos los que son vecinos son v_2 y v_4, asi que v_5 no puede tener el color 0 ni el 1. Greedy lo colorea $c(v_5) = 2$.

Continuando con el ejemplo

- Los vértices anteriores a v_3 son v_1 y v_2, y v_2 es vecino, asi que v_3 no puede tener el color 0. Greedy lo colorea $c(v_3) = 1$.
- Los vértices anteriores a v_3 son v_1, v_2 y v_3, y v_1 es el único vecino, asi que v_4 no puede tener el color 0. Greedy lo colorea $c(v_4) = 1$.
- Los vértices anteriores a v_5 son v_1, v_2, v_3 y v_4, y entre ellos los que son vecinos son v_2 y v_4, asi que v_5 no puede tener el color 0 ni el 1. Greedy lo colorea $c(v_5) = 2$.
- Los vértices anteriores a v_6 son v_1, v_2, v_3, v_4 y v_5, y entre ellos los que son vecinos son v_1 y v_3, asi que v_6 no puede tener el color 0 ni el 1. Greedy lo colorea $c(v_6) = 2$.

Continuando con el ejemplo

- Vemos que colorea los 2 primeros con un color, los 4 primeros con 2 colores, los 6 primeros con 3.

Continuando con el ejemplo

- Vemos que colorea los 2 primeros con un color, los 4 primeros con 2 colores, los 6 primeros con 3.
- Podriamos poner como hipotesis inductiva que Greedy colorea los primeros $2i$ vértices con i colores,

Continuando con el ejemplo

- Vemos que colorea los 2 primeros con un color, los 4 primeros con 2 colores, los 6 primeros con 3.
- Podriamos poner como hipotesis inductiva que Greedy colorea los primeros $2i$ vértices con i colores,
- Ademas podemos poner dentro de la hipotesis inductiva que para todo $k \leq 2i$, el color de v_k es igual al color de v_{k-1} si k es par, y que ese color común es $\frac{k}{2} - 1$, pues eso es lo que ocurre en los primeros casos.

Continuando con el ejemplo

- Vemos que colorea los 2 primeros con un color, los 4 primeros con 2 colores, los 6 primeros con 3.
- Podriamos poner como hipotesis inductiva que Greedy colorea los primeros $2i$ vértices con i colores,
- Ademas podemos poner dentro de la hipotesis inductiva que para todo $k \leq 2i$, el color de v_k es igual al color de v_{k-1} si k es par, y que ese color común es $\frac{k}{2} - 1$, pues eso es lo que ocurre en los primeros casos.
- Probemos entonces esta hipotesis, suponiendola verdadera para i y probandola para $i + 1$.

Continuando con el ejemplo

- Los vértices anteriores a v_{2i+1} son $v_1, v_2, ..., v_{2i}$, y entre ellos los que son vecinos de v_{2i+1} son los v_t con t par, $t \leq 2i$.

Continuando con el ejemplo

- Los vértices anteriores a v_{2i+1} son $v_1, v_2, ..., v_{2i}$, y entre ellos los que son vecinos de v_{2i+1} son los v_t con t par, $t \leq 2i$.
- Por hipotesis inductiva, el color de v_t con t par$\leq 2i$ es $\frac{t}{2} - 1$, asi que v_{2i+1} no puede tener ninguno de los colores $\{\frac{t}{2} - 1 : t = 2, 4, ..., 2i\} = \{0, 1, ..., i-1\}$

Continuando con el ejemplo

- Los vértices anteriores a v_{2i+1} son $v_1, v_2, ..., v_{2i}$, y entre ellos los que son vecinos de v_{2i+1} son los v_t con t par, $t \leq 2i$.
- Por hipotesis inductiva, el color de v_t con t par$\leq 2i$ es $\frac{t}{2} - 1$, asi que v_{2i+1} no puede tener ninguno de los colores $\{\frac{t}{2} - 1 : t = 2, 4, ..., 2i\} = \{0, 1, ..., i - 1\}$
- Por lo tanto, Greedy le da el menor color posible: $c(v_{2i+1}) = i$.

Continuando con el ejemplo

- Los vértices anteriores a $v_{2i+2} = v_{2(i+1)}$ son $v_1, v_2, ..., v_{2i+1}$, y entre ellos los que son vecinos de v_{2i+2} son los v_t con t impar, $t < 2i$, pues v_{2i+1} no es vecino de v_{2i+2} y los pares tampoco.

Continuando con el ejemplo

- Los vértices anteriores a $v_{2i+2} = v_{2(i+1)}$ son $v_1, v_2, ..., v_{2i+1}$, y entre ellos los que son vecinos de v_{2i+2} son los v_t con t impar, $t < 2i$, pues v_{2i+1} no es vecino de v_{2i+2} y los pares tampoco.
- Por hipotesis inductiva, el color de v_t con t impar$\leq 2i$ es $\frac{t+1}{2} - 1 = \frac{t-1}{2}$, asi que v_{2i+2} no puede tener ninguno de los colores $\{\frac{t-1}{2} : t = 1, 3, ..., 2i-1\} = \{\frac{j}{2} : j = 0, 2, ..., 2i-2\} = \{0, 1, ..., i-1\}$

Continuando con el ejemplo

- Los vértices anteriores a $v_{2i+2} = v_{2(i+1)}$ son $v_1, v_2, ..., v_{2i+1}$, y entre ellos los que son vecinos de v_{2i+2} son los v_t con t impar, $t < 2i$, pues v_{2i+1} no es vecino de v_{2i+2} y los pares tampoco.
- Por hipotesis inductiva, el color de v_t con t impar$\leq 2i$ es $\frac{t+1}{2} - 1 = \frac{t-1}{2}$, asi que v_{2i+2} no puede tener ninguno de los colores $\{\frac{t-1}{2} : t = 1, 3, ..., 2i-1\} = \{\frac{j}{2} : j = 0, 2, ..., 2i-2\} = \{0, 1, ..., i-1\}$
- Por lo tanto, Greedy le da el menor color posible: $c(v_{2i+2}) = i$.

Continuando con el ejemplo

- Los vértices anteriores a $v_{2i+2} = v_{2(i+1)}$ son $v_1, v_2, ..., v_{2i+1}$, y entre ellos los que son vecinos de v_{2i+2} son los v_t con t impar, $t < 2i$, pues v_{2i+1} no es vecino de v_{2i+2} y los pares tampoco.
- Por hipotesis inductiva, el color de v_t con t impar$\leq 2i$ es $\frac{t+1}{2} - 1 = \frac{t-1}{2}$, asi que v_{2i+2} no puede tener ninguno de los colores $\{\frac{t-1}{2} : t = 1, 3, ..., 2i-1\} = \{\frac{j}{2} : j = 0, 2, ..., 2i-2\} = \{0, 1, ..., i-1\}$
- Por lo tanto, Greedy le da el menor color posible: $c(v_{2i+2}) = i$.
- Y hemos probado la hipotesis inductiva.

Cont. ejemplo

- Por lo tanto, Greedy colorea el grafo con $\frac{n}{2}$ colores.

Cont. ejemplo

- Por lo tanto, Greedy colorea el grafo con $\frac{n}{2}$ colores.
- Por otro lado, podriamos colorear $c(v_i) = (i \mod 2)$.

Cont. ejemplo

- Por lo tanto, Greedy colorea el grafo con $\frac{n}{2}$ colores.
- Por otro lado, podriamos colorear $c(v_i) = (i \mod 2)$.
- Puesto que no hay lados entre vértices v_i con i impar, ni hay lados entre vértices v_i con i par, ese coloreo es propio.

Cont. ejemplo

- Por lo tanto, Greedy colorea el grafo con $\frac{n}{2}$ colores.
- Por otro lado, podriamos colorear $c(v_i) = (i \mod 2)$.
- Puesto que no hay lados entre vértices v_i con i impar, ni hay lados entre vértices v_i con i par, ese coloreo es propio.
- Asi que $\chi(G) = 2$ pero Greedy usa $\frac{n}{2}$ colores, asi que la diferencia entre $\chi(G)$ puede ser tan grande como se quiera.

Cont. ejemplo

- Por lo tanto, Greedy colorea el grafo con $\frac{n}{2}$ colores.
- Por otro lado, podriamos colorear $c(v_i) = (i \mod 2)$.
- Puesto que no hay lados entre vértices v_i con i impar, ni hay lados entre vértices v_i con i par, ese coloreo es propio.
- Asi que $\chi(G) = 2$ pero Greedy usa $\frac{n}{2}$ colores, asi que la diferencia entre $\chi(G)$ puede ser tan grande como se quiera.
- En el ejercicio 5 del práctico se les pide generalizar este ejemplo para n impar, e incluso ver que se puede dar un ejemplo en el caso n par para el cual $\chi(G) = 2$ pero Greedy usa $\frac{n}{2} + 1$ colores.

Cotas para Greedy

- Pero hay alguna cota útiles que podemos obtener, usando Greedy.

Cotas para Greedy

- Pero hay alguna cota útiles que podemos obtener, usando Greedy.

Teorema

$$\chi(G) \leq \Delta + 1$$

Cotas para Greedy

- Pero hay alguna cota útiles que podemos obtener, usando Greedy.

Teorema

$$\chi(G) \leq \Delta + 1$$

- Prueba: Cuando Greedy colorea un vértice x, debe eliminar de consideración todos los colores de los vecinos de x que esten antes que x en la lista de vértices.

Cotas para Greedy

- Pero hay alguna cota útiles que podemos obtener, usando Greedy.

Teorema

$$\chi(G) \leq \Delta + 1$$

- Prueba: Cuando Greedy colorea un vértice x, debe eliminar de consideración todos los colores de los vecinos de x que esten antes que x en la lista de vértices.
- En el peor de los casos, todos los vecinos de x estan antes.

Cotas para Greedy

- Pero hay alguna cota útiles que podemos obtener, usando Greedy.

Teorema

$$\chi(G) \leq \Delta + 1$$

- Prueba: Cuando Greedy colorea un vértice x, debe eliminar de consideración todos los colores de los vecinos de x que esten antes que x en la lista de vértices.
- En el peor de los casos, todos los vecinos de x estan antes.
- Asi que en el peor de los casos, Greedy elimina $d(x)$ colores.

Cotas para Greedy

- Pero hay alguna cota útiles que podemos obtener, usando Greedy.

Teorema

$$\chi(G) \leq \Delta + 1$$

- Prueba: Cuando Greedy colorea un vértice x, debe eliminar de consideración todos los colores de los vecinos de x que esten antes que x en la lista de vértices.
- En el peor de los casos, todos los vecinos de x estan antes.
- Asi que en el peor de los casos, Greedy elimina $d(x)$ colores.
- Como $d(x) \leq \Delta$, si tenemos $\Delta + 1$ colores disponibles siempre habrá un color extra para colorear a x.QED

Cotas para Greedy

- ¿Que tan realista es esa cota? ¿Hay grafos para los que $\chi(G) = \Delta + 1$?

Cotas para Greedy

- ¿Que tan realista es esa cota? ¿Hay grafos para los que $\chi(G) = \Delta + 1$?
- Si, una cantidad infinita.

Cotas para Greedy

- ¿Que tan realista es esa cota? ¿Hay grafos para los que $\chi(G) = \Delta + 1$?
- Si, una cantidad infinita.
- Por ejemplo, todos los grafos completos:

Cotas para Greedy

- ¿Que tan realista es esa cota? ¿Hay grafos para los que $\chi(G) = \Delta + 1$?
- Si, una cantidad infinita.
- Por ejemplo, todos los grafos completos:
  - $\chi(K_n) = n$

Cotas para Greedy

- ¿Que tan realista es esa cota? ¿Hay grafos para los que $\chi(G) = \Delta + 1$?
- Si, una cantidad infinita.
- Por ejemplo, todos los grafos completos:
  - $\chi(K_n) = n$
  - $\Delta(K_n) = n - 1$

Cotas para Greedy

- ¿Que tan realista es esa cota? ¿Hay grafos para los que $\chi(G) = \Delta + 1$?
- Si, una cantidad infinita.
- Por ejemplo, todos los grafos completos:
  - $\chi(K_n) = n$
  - $\Delta(K_n) = n - 1$
- O los ciclos impares:

Cotas para Greedy

- ¿Que tan realista es esa cota? ¿Hay grafos para los que $\chi(G) = \Delta + 1$?
- Si, una cantidad infinita.
- Por ejemplo, todos los grafos completos:
  - $\chi(K_n) = n$
  - $\Delta(K_n) = n - 1$
- O los ciclos impares:
  - $\chi(C_{2r+1}) = 3$

Cotas para Greedy

- ¿Que tan realista es esa cota? ¿Hay grafos para los que $\chi(G) = \Delta + 1$?
- Si, una cantidad infinita.
- Por ejemplo, todos los grafos completos:
  - $\chi(K_n) = n$
  - $\Delta(K_n) = n - 1$
- O los ciclos impares:
  - $\chi(C_{2r+1}) = 3$
  - $\Delta(C_{2r+1}) = 2$

Cotas para Greedy

- ¿Que tan realista es esa cota? ¿Hay grafos para los que $\chi(G) = \Delta + 1$?
- Si, una cantidad infinita.
- Por ejemplo, todos los grafos completos:
  - $\chi(K_n) = n$
  - $\Delta(K_n) = n - 1$
- O los ciclos impares:
  - $\chi(C_{2r+1}) = 3$
  - $\Delta(C_{2r+1}) = 2$
- O bien un grafo con muchas componentes conexas, algunas de ellas que sean ciclos pares y al menos una que sea un ciclo impar.

Cotas para Greedy

- ¿Que tan realista es esa cota? ¿Hay grafos para los que $\chi(G) = \Delta + 1$?
- Si, una cantidad infinita.
- Por ejemplo, todos los grafos completos:
  - $\chi(K_n) = n$
  - $\Delta(K_n) = n - 1$
- O los ciclos impares:
  - $\chi(C_{2r+1}) = 3$
  - $\Delta(C_{2r+1}) = 2$
- O bien un grafo con muchas componentes conexas, algunas de ellas que sean ciclos pares y al menos una que sea un ciclo impar.
- O un grafo con muchas componentes conexas todas con menos de r vertices y una componente conexa que sea un K_r

Brooks

- Se pueden construir mas ejemplos usando grafos disconexos, obviamente.

Brooks

- Se pueden construir mas ejemplos usando grafos disconexos, obviamente.
- Pero ¿hay algún otro ejemplo conexo?

Brooks

- Se pueden construir mas ejemplos usando grafos disconexos, obviamente.
- Pero ¿hay algún otro ejemplo conexo?
- No:

Brooks

- Se pueden construir mas ejemplos usando grafos disconexos, obviamente.
- Pero ¿hay algún otro ejemplo conexo?
- No:

Teorema de Brooks (1941)

Si G es conexo, entonces $\chi(G) \leq \Delta$, a menos que G sea un ciclo impar o un grafo completo.

Demostración de Brooks

- Sea x tal que $d(x) = \delta$.

Demostración de Brooks

- Sea x tal que $d(x) = \delta$.
- Corramos BFS(x). (o DFS(x), el que quieran)

Demostración de Brooks

- Sea x tal que $d(x) = \delta$.
- Corramos BFS(x). (o DFS(x), el que quieran)
- Cuando BFS(x) termina, obtiene la componente conexa donde esta x, pero como G es conexo, hay una sola componente conexa, asi que se obtienen todos los vértices de G.

Demostración de Brooks

- Sea x tal que $d(x) = \delta$.
- Corramos BFS(x). (o DFS(x), el que quieran)
- Cuando BFS(x) termina, obtiene la componente conexa donde esta x, pero como G es conexo, hay una sola componente conexa, asi que se obtienen todos los vértices de G.
- Al correr BFS(x), se iran incorporando ciertos vértices en cierto orden.

Demostración de Brooks

- Sea x tal que $d(x) = \delta$.
- Corramos BFS(x). (o DFS(x), el que quieran)
- Cuando BFS(x) termina, obtiene la componente conexa donde esta x, pero como G es conexo, hay una sola componente conexa, asi que se obtienen todos los vértices de G.
- Al correr BFS(x), se iran incorporando ciertos vértices en cierto orden.
- Ordenemos los vértices de G en el orden inverso al cual fueron incorporados por BFS(x).

Demostración de Brooks

- Sea x tal que $d(x) = \delta$.
- Corramos BFS(x). (o DFS(x), el que quieran)
- Cuando BFS(x) termina, obtiene la componente conexa donde esta x, pero como G es conexo, hay una sola componente conexa, asi que se obtienen todos los vértices de G.
- Al correr BFS(x), se iran incorporando ciertos vértices en cierto orden.
- Ordenemos los vértices de G en el orden inverso al cual fueron incorporados por BFS(x).
- El orden será $x_1, x_2, ..., x_n$ con $x_n = x$ porque x es el primero al correr BFS(x) asi que es el último en el orden inverso.

Demostración de Brooks (cont.)

- Salvo x, todos los demas vértices son incorporados a la componente conexa que esta construyendo BFS(x) por un vértice que es vecino y que ya está. (lo mismo vale para DFS si prefieren DFS).

Demostración de Brooks (cont.)

- Salvo x, todos los demas vértices son incorporados a la componente conexa que esta construyendo BFS(x) por un vértice que es vecino y que ya está. (lo mismo vale para DFS si prefieren DFS).
- Asi que en el orden en que se incorporan los vértices, todo vértice, salvo x, tiene un vécino anterior.

Demostración de Brooks (cont.)

- Salvo x, todos los demas vértices son incorporados a la componente conexa que esta construyendo BFS(x) por un vértice que es vecino y que ya está. (lo mismo vale para DFS si prefieren DFS).
- Asi que en el orden en que se incorporan los vértices, todo vértice, salvo x, tiene un vécino anterior.
- Entonces, en el orden inverso, tenemos que todo vertice (salvo x) tiene un vécino posterior

Demostración de Brooks (cont.)

- Salvo x, todos los demas vértices son incorporados a la componente conexa que esta construyendo BFS(x) por un vértice que es vecino y que ya está. (lo mismo vale para DFS si prefieren DFS).
- Asi que en el orden en que se incorporan los vértices, todo vértice, salvo x, tiene un vécino anterior.
- Entonces, en el orden inverso, tenemos que todo vertice (salvo x) tiene un vécino posterior
- Greedy le da el color 0 a x_1.

Demostración de Brooks (cont.)

- Salvo x, todos los demas vértices son incorporados a la componente conexa que esta construyendo BFS(x) por un vértice que es vecino y que ya está. (lo mismo vale para DFS si prefieren DFS).
- Asi que en el orden en que se incorporan los vértices, todo vértice, salvo x, tiene un vécino anterior.
- Entonces, en el orden inverso, tenemos que todo vertice (salvo x) tiene un vécino posterior
- Greedy le da el color 0 a x_1.
- A los demas vértices, Greedy les revisa los colores de los vecinos anteriores.

Demostración de Brooks (cont.)

- Pero si tomamos un vértice x_i con $1 < i < n$, entonces x_i, por lo que dijimos antes, tiene al menos a un vértice posterior que es vecino de x.

Demostración de Brooks (cont.)

- Pero si tomamos un vértice x_i con $1 < i < n$, entonces x_i, por lo que dijimos antes, tiene al menos a un vértice posterior que es vecino de x.
- Lo cual quiere decir que no todos sus vécinos estan antes que el en el orden.

Demostración de Brooks (cont.)

- Pero si tomamos un vértice x_i con $1 < i < n$, entonces x_i, por lo que dijimos antes, tiene al menos a un vértice posterior que es vecino de x.
- Lo cual quiere decir que no todos sus vécinos estan antes que el en el orden.
- Entonces x_i tiene a lo sumo $d(x_i) - 1$ vecinos anteriores, con lo cual Greedy elimina a lo suma esa cantidad de colores.

Demostración de Brooks (cont.)

- Pero si tomamos un vértice x_i con $1 < i < n$, entonces x_i, por lo que dijimos antes, tiene al menos a un vértice posterior que es vecino de x.
- Lo cual quiere decir que no todos sus vécinos estan antes que el en el orden.
- Entonces x_i tiene a lo sumo $d(x_i) - 1$ vecinos anteriores, con lo cual Greedy elimina a lo suma esa cantidad de colores.
- Como $d(x_i) - 1 \leq \Delta - 1$, entonces si tenemos Δ colores disponibles, siempre habrá uno sobrante para colorear x_i. (siempre que $i < n$)

Demostración de Brooks (cont.)

- Pero si tomamos un vértice x_i con $1 < i < n$, entonces x_i, por lo que dijimos antes, tiene al menos a un vértice posterior que es vecino de x.
- Lo cual quiere decir que no todos sus vécinos estan antes que el en el orden.
- Entonces x_i tiene a lo sumo $d(x_i) - 1$ vecinos anteriores, con lo cual Greedy elimina a lo suma esa cantidad de colores.
- Como $d(x_i) - 1 \leq \Delta - 1$, entonces si tenemos Δ colores disponibles, siempre habrá uno sobrante para colorear x_i. (siempre que $i < n$)
- Resumamos esto, que es importante mas allá del resto de la prueba.

Demostración de Brooks (cont.)

Propiedad

Si G es conexo, entonces existe un ordenamiento de los vértices tal que Greeedy colorea todos los vértices, salvo uno, con Δ colores o menos.

Demostración de Brooks (cont.)

Propiedad

Si G es conexo, entonces existe un ordenamiento de los vértices tal que Greeedy colorea todos los vértices, salvo uno, con Δ colores o menos.

- Bueno, nos queda por colorear x.

Demostración de Brooks (cont.)

Propiedad

Si G es conexo, entonces existe un ordenamiento de los vértices tal que Greeedy colorea todos los vértices, salvo uno, con Δ colores o menos.

- Bueno, nos queda por colorear x.
- Si G no es regular, es trivial.

Demostración de Brooks (cont.)

Propiedad

Si G es conexo, entonces existe un ordenamiento de los vértices tal que Greeedy colorea todos los vértices, salvo uno, con Δ colores o menos.

- Bueno, nos queda por colorear x.
- Si G no es regular, es trivial.
- Pues $d(x) = \delta$ y $\delta < \Delta$ si G no es regular.

Demostración de Brooks (cont.)

Propiedad

Si G es conexo, entonces existe un ordenamiento de los vértices tal que Greeedy colorea todos los vértices, salvo uno, con Δ colores o menos.

- Bueno, nos queda por colorear x.
- Si G no es regular, es trivial.
- Pues $d(x) = \delta$ y $\delta < \Delta$ si G no es regular.
- Entonces para colorear x Greedy elimina a lo sumo $d(x) = \delta$ colores, y como $\delta < \Delta$, le queda al menos un color para colorear x.

Demostración de Brooks (cont.)

Propiedad

Si G es conexo, entonces existe un ordenamiento de los vértices tal que Greeedy colorea todos los vértices, salvo uno, con Δ colores o menos.

- Bueno, nos queda por colorear x.
- Si G no es regular, es trivial.
- Pues $d(x) = \delta$ y $\delta < \Delta$ si G no es regular.
- Entonces para colorear x Greedy elimina a lo sumo $d(x) = \delta$ colores, y como $\delta < \Delta$, le queda al menos un color para colorear x.
- Queda el caso regular, y sólo un vértice. (observemos que tanto los ciclos impares como los completos son regulares, y es en esta parte donde usaremos que G no es uno de ellos).

Demostración de Brooks (cont.)

Propiedad

Si G es conexo, entonces existe un ordenamiento de los vértices tal que Greeedy colorea todos los vértices, salvo uno, con Δ colores o menos.

- Bueno, nos queda por colorear x.
- Si G no es regular, es trivial.
- Pues $d(x) = \delta$ y $\delta < \Delta$ si G no es regular.
- Entonces para colorear x Greedy elimina a lo sumo $d(x) = \delta$ colores, y como $\delta < \Delta$, le queda al menos un color para colorear x.
- Queda el caso regular, y sólo un vértice. (observemos que tanto los ciclos impares como los completos son regulares, y es en esta parte donde usaremos que G no es uno de ellos).
- Pan comido, no?

Demostración de Brooks (cont.)

Propiedad

Si G es conexo, entonces existe un ordenamiento de los vértices tal que Greeedy colorea todos los vértices, salvo uno, con Δ colores o menos.

- Bueno, nos queda por colorear x.
- Si G no es regular, es trivial.
- Pues $d(x) = \delta$ y $\delta < \Delta$ si G no es regular.
- Entonces para colorear x Greedy elimina a lo sumo $d(x) = \delta$ colores, y como $\delta < \Delta$, le queda al menos un color para colorear x.
- Queda el caso regular, y sólo un vértice. (observemos que tanto los ciclos impares como los completos son regulares, y es en esta parte donde usaremos que G no es uno de ellos).
- Pan comido, no?
- No. Lo dejamos para la semana que viene.

- Greedy depende del orden en que se corra los vértices.

- Greedy depende del orden en que se corra los vértices.
- Una estrategia consistiría en correr Greedy en algún orden, luego en otro, luego en otro, etc, y quedarse con el mejor coloreo posible.

- Greedy depende del orden en que se corra los vértices.
- Una estrategia consistiría en correr Greedy en algún orden, luego en otro, luego en otro, etc, y quedarse con el mejor coloreo posible.
- Quizas no sea $\chi(G)$ pero habria alguna esperanza de tener un coloreo aunque sea aproximado.

- Greedy depende del orden en que se corra los vértices.
- Una estrategia consistiría en correr Greedy en algún orden, luego en otro, luego en otro, etc, y quedarse con el mejor coloreo posible.
- Quizas no sea $\chi(G)$ pero habria alguna esperanza de tener un coloreo aunque sea aproximado.
- En general si uno simplemente busca reordenes al azar, es muy dificil encontrar uno que funcione bien.

- Greedy depende del orden en que se corra los vértices.
- Una estrategia consistiría en correr Greedy en algún orden, luego en otro, luego en otro, etc, y quedarse con el mejor coloreo posible.
- Quizas no sea $\chi(G)$ pero habria alguna esperanza de tener un coloreo aunque sea aproximado.
- En general si uno simplemente busca reordenes al azar, es muy dificil encontrar uno que funcione bien.
- Otra posible estrategia seria ir reordenando los vértices de forma tal de asegurarnos de nunca empeorar el coloreo.

- Greedy depende del orden en que se corra los vértices.
- Una estrategia consistiría en correr Greedy en algún orden, luego en otro, luego en otro, etc, y quedarse con el mejor coloreo posible.
- Quizas no sea $\chi(G)$ pero habria alguna esperanza de tener un coloreo aunque sea aproximado.
- En general si uno simplemente busca reordenes al azar, es muy dificil encontrar uno que funcione bien.
- Otra posible estrategia seria ir reordenando los vértices de forma tal de asegurarnos de nunca empeorar el coloreo.
- ¿esto es posible?

- Greedy depende del orden en que se corra los vértices.
- Una estrategia consistiría en correr Greedy en algún orden, luego en otro, luego en otro, etc, y quedarse con el mejor coloreo posible.
- Quizas no sea $\chi(G)$ pero habria alguna esperanza de tener un coloreo aunque sea aproximado.
- En general si uno simplemente busca reordenes al azar, es muy dificil encontrar uno que funcione bien.
- Otra posible estrategia seria ir reordenando los vértices de forma tal de asegurarnos de nunca empeorar el coloreo.
- ¿esto es posible?
- Si, gracias al siguiente teorema.

VIT

Very Important Theorem (VIT)

Sea $G = (V, E)$ un grafo cuyos vértices estan coloreados con un coloreo propio c con r colores $\{0, 1, ..., r - 1\}$.

VIT

Very Important Theorem (VIT)

Sea $G = (V, E)$ un grafo cuyos vértices estan coloreados con un coloreo propio c con r colores $\{0, 1, ..., r-1\}$.
Sea π una permutación de los números $0, 1, ..., r-1$

VIT

Very Important Theorem (VIT)

Sea $G = (V, E)$ un grafo cuyos vértices estan coloreados con un coloreo propio c con r colores $\{0, 1, ..., r-1\}$.
Sea π una permutación de los números $0, 1, ..., r-1$, es decir, $\pi : \{0, 1, ..., r-1\} \mapsto \{0, 1, ..., r-1\}$ es una biyección.

VIT

Very Important Theorem (VIT)

Sea $G = (V, E)$ un grafo cuyos vértices estan coloreados con un coloreo propio c con r colores $\{0, 1, ..., r - 1\}$.
Sea π una permutación de los números $0, 1, ..., r - 1$, es decir, $\pi : \{0, 1, ..., r - 1\} \mapsto \{0, 1, ..., r - 1\}$ es una biyección.
Sea $V_i = \{x \in V : c(x) = i\}, i = 0, 1, ..., r - 1$.

VIT

Very Important Theorem (VIT)

Sea $G = (V, E)$ un grafo cuyos vértices estan coloreados con un coloreo propio c con r colores $\{0, 1, ..., r-1\}$.
Sea π una permutación de los números $0, 1, ..., r-1$, es decir, $\pi : \{0, 1, ..., r-1\} \mapsto \{0, 1, ..., r-1\}$ es una biyección.
Sea $V_i = \{x \in V : c(x) = i\}, i = 0, 1, ..., r-1$.
Ordenemos los vértices poniendo primero los vértices de $V_{\pi(0)}$, luego los de $V_{\pi(1)}$,etc, hasta $V_{\pi(r-1)}$.

VIT

Very Important Theorem (VIT)

Sea $G = (V, E)$ un grafo cuyos vértices estan coloreados con un coloreo propio c con r colores $\{0, 1, ..., r-1\}$.
Sea π una permutación de los números $0, 1, ..., r-1$, es decir, $\pi : \{0, 1, ..., r-1\} \mapsto \{0, 1, ..., r-1\}$ es una biyección.
Sea $V_i = \{x \in V : c(x) = i\}, i = 0, 1, ..., r-1$.
Ordenemos los vértices poniendo primero los vértices de $V_{\pi(0)}$, luego los de $V_{\pi(1)}$,etc, hasta $V_{\pi(r-1)}$. (el orden interno de los vértices dentro de cada $V_{\pi(i)}$ es irrelevante)

VIT

Very Important Theorem (VIT)

Sea $G = (V, E)$ un grafo cuyos vértices estan coloreados con un coloreo propio c con r colores $\{0, 1, ..., r-1\}$.
Sea π una permutación de los números $0, 1, ..., r-1$, es decir, $\pi : \{0, 1, ..., r-1\} \mapsto \{0, 1, ..., r-1\}$ es una biyección.
Sea $V_i = \{x \in V : c(x) = i\}, i = 0, 1, ..., r-1$.
Ordenemos los vértices poniendo primero los vértices de $V_{\pi(0)}$, luego los de $V_{\pi(1)}$,etc, hasta $V_{\pi(r-1)}$. (el orden interno de los vértices dentro de cada $V_{\pi(i)}$ es irrelevante)

Entonces Greedy en ese orden coloreará G con r colores o menos.

Prueba del VIT

- Por inducción. Sea $W_i = V_{\pi(0)} \cup V_{\pi(1)} \cup \cdots \cup V_{\pi(i-1)}$.

Prueba del VIT

- Por inducción. Sea $W_i = V_{\pi(0)} \cup V_{\pi(1)} \cup \cdots \cup V_{\pi(i-1)}$.
- La hipotesis inductiva será que Greedy no usa mas de i colores para colorear W_i.

Prueba del VIT

- Por inducción. Sea $W_i = V_{\pi(0)} \cup V_{\pi(1)} \cup \cdots \cup V_{\pi(i-1)}$.
- La hipotesis inductiva será que Greedy no usa mas de i colores para colorear W_i.
- Como $W_r = V$, si probamos la hipotesis inductiva probamos el teorema.

Prueba del VIT

- Por inducción. Sea $W_i = V_{\pi(0)} \cup V_{\pi(1)} \cup \cdots \cup V_{\pi(i-1)}$.
- La hipotesis inductiva será que Greedy no usa mas de i colores para colorear W_i.
- Como $W_r = V$, si probamos la hipotesis inductiva probamos el teorema.
- Caso base: $i = 1$. (por lo tanto $V_{\pi(i-1)} = V_{\pi(0)}$)

Prueba del VIT

- Por inducción. Sea $W_i = V_{\pi(0)} \cup V_{\pi(1)} \cup \dots \cup V_{\pi(i-1)}$.
- La hipotesis inductiva será que Greedy no usa mas de i colores para colorear W_i.
- Como $W_r = V$, si probamos la hipotesis inductiva probamos el teorema.
- Caso base: $i = 1$. (por lo tanto $V_{\pi(i-1)} = V_{\pi(0)}$)
  - Los vértices de $V_{\pi(0)}$ tienen todos el mismo color ($\pi(0)$) asi que no puede haber ningún lado entre esos vértices.

Prueba del VIT

- Por inducción. Sea $W_i = V_{\pi(0)} \cup V_{\pi(1)} \cup \cdots \cup V_{\pi(i-1)}$.
- La hipotesis inductiva será que Greedy no usa mas de i colores para colorear W_i.
- Como $W_r = V$, si probamos la hipotesis inductiva probamos el teorema.
- Caso base: $i = 1$. (por lo tanto $V_{\pi(i-1)} = V_{\pi(0)}$)
  - Los vértices de $V_{\pi(0)}$ tienen todos el mismo color ($\pi(0)$) asi que no puede haber ningún lado entre esos vértices.
  - Por lo tanto, Greedy los coloreará a todos con el color 0.

Prueba del VIT

- Por inducción. Sea $W_i = V_{\pi(0)} \cup V_{\pi(1)} \cup \cdots \cup V_{\pi(i-1)}$.
- La hipotesis inductiva será que Greedy no usa mas de i colores para colorear W_i.
- Como $W_r = V$, si probamos la hipotesis inductiva probamos el teorema.
- Caso base: $i = 1$. (por lo tanto $V_{\pi(i-1)} = V_{\pi(0)}$)
  - Los vértices de $V_{\pi(0)}$ tienen todos el mismo color ($\pi(0)$) asi que no puede haber ningún lado entre esos vértices.
  - Por lo tanto, Greedy los coloreará a todos con el color 0.
- Listo el caso base.

Prueba del VIT (Cont.)

- Supongamos la hipotesis cierta para i y probemosla para $i+1$.

Prueba del VIT (Cont.)

- Supongamos la hipotesis cierta para i y probemosla para $i+1$.
- es decir, probemos que Greedy colorea W_{i+1} con a lo sumo $i+1$ colores.

Prueba del VIT (Cont.)

- Supongamos la hipotesis cierta para i y probemosla para $i+1$.
- es decir, probemos que Greedy colorea W_{i+1} con a lo sumo $i+1$ colores.
- Supongamos que no es cierto.

Prueba del VIT (Cont.)

- Supongamos la hipotesis cierta para i y probemosla para $i+1$.
- es decir, probemos que Greedy colorea W_{i+1} con a lo sumo $i+1$ colores.
- Supongamos que no es cierto.
- Entonces, Greedy debe usar al menos $i+2$ colores, lo que significa que existe al menos un vértice x que esta coloreado con el color $i+1$ (porque empezamos a colorear desde 0).

Prueba del VIT (Cont.)

- Supongamos la hipotesis cierta para i y probemosla para $i+1$.
- es decir, probemos que Greedy colorea W_{i+1} con a lo sumo $i+1$ colores.
- Supongamos que no es cierto.
- Entonces, Greedy debe usar al menos $i+2$ colores, lo que significa que existe al menos un vértice x que esta coloreado con el color $i+1$ (porque empezamos a colorear desde 0).
- Para distinguir el coloreo que da Greedy del coloreo original que llamamos c, llamaremos al coloreo dado por Greedy en este orden por g.

Prueba del VIT (Cont.)

- Supongamos la hipotesis cierta para i y probemosla para $i+1$.
- es decir, probemos que Greedy colorea W_{i+1} con a lo sumo $i+1$ colores.
- Supongamos que no es cierto.
- Entonces, Greedy debe usar al menos $i+2$ colores, lo que significa que existe al menos un vértice x que esta coloreado con el color $i+1$ (porque empezamos a colorear desde 0).
- Para distinguir el coloreo que da Greedy del coloreo original que llamamos c, llamaremos al coloreo dado por Greedy en este orden por g.
- Es decir, estamos diciendo que existe x tal que $g(x)=i+1$.

cont. prueba VIT

- Cuando Greedy colorea un vértice, le da el menor color posible que no cause conflicto con los vértices anterioremente coloreados.

cont. prueba VIT

- Cuando Greedy colorea un vértice, le da el menor color posible que no cause conflicto con los vértices anterioremente coloreados.
- Por lo tanto si $g(x) = i + 1$, entonces debe haber vecinos v_k de x, $k = 0, 1, ..., i$, anteriores a x en el orden, tales que $g(v_k) = k$, .

cont. prueba VIT

- Cuando Greedy colorea un vértice, le da el menor color posible que no cause conflicto con los vértices anterioremente coloreados.
- Por lo tanto si $g(x) = i + 1$, entonces debe haber vecinos v_k de x, $k = 0, 1, ..., i$, anteriores a x en el orden, tales que $g(v_k) = k$, .
- Observemos que $W_{i+1} = W_i \cup V_{\pi(i)}$ y que los vértices de W_i estan antes en el orden que los de $V_{\pi(i)}$, por lo tanto Greedy los coloreará primero.

cont. prueba VIT

- Cuando Greedy colorea un vértice, le da el menor color posible que no cause conflicto con los vértices anterioremente coloreados.
- Por lo tanto si $g(x) = i + 1$, entonces debe haber vecinos v_k de x, $k = 0, 1, ..., i$, anteriores a x en el orden, tales que $g(v_k) = k$, .
- Observemos que $W_{i+1} = W_i \cup V_{\pi(i)}$ y que los vértices de W_i estan antes en el orden que los de $V_{\pi(i)}$, por lo tanto Greedy los coloreará primero.
- Por hipotesis inductiva, Greedy coloreará esos vértices de W_i con a lo sumo i colores.

cont. prueba VIT

- Cuando Greedy colorea un vértice, le da el menor color posible que no cause conflicto con los vértices anterioremente coloreados.
- Por lo tanto si $g(x) = i + 1$, entonces debe haber vecinos v_k de x, $k = 0, 1, ..., i$, anteriores a x en el orden, tales que $g(v_k) = k$, .
- Observemos que $W_{i+1} = W_i \cup V_{\pi(i)}$ y que los vértices de W_i estan antes en el orden que los de $V_{\pi(i)}$, por lo tanto Greedy los coloreará primero.
- Por hipotesis inductiva, Greedy coloreará esos vértices de W_i con a lo sumo i colores.
- Por la forma que colorea Greedy, esos colores son $0, 1, ..., i - 1$.

cont. prueba VIT

- Cuando Greedy colorea un vértice, le da el menor color posible que no cause conflicto con los vértices anterioremente coloreados.
- Por lo tanto si $g(x) = i + 1$, entonces debe haber vecinos v_k de x, $k = 0, 1, ..., i$, anteriores a x en el orden, tales que $g(v_k) = k$, .
- Observemos que $W_{i+1} = W_i \cup V_{\pi(i)}$ y que los vértices de W_i estan antes en el orden que los de $V_{\pi(i)}$, por lo tanto Greedy los coloreará primero.
- Por hipotesis inductiva, Greedy coloreará esos vértices de W_i con a lo sumo i colores.
- Por la forma que colorea Greedy, esos colores son $0, 1, ..., i - 1$.
- Como $g(x) = i + 1$, lo anterior implica que $x \notin W_i$.

cont. prueba VIT

- Cuando Greedy colorea un vértice, le da el menor color posible que no cause conflicto con los vértices anterioremente coloreados.
- Por lo tanto si $g(x) = i + 1$, entonces debe haber vecinos v_k de x, $k = 0, 1, ..., i$, anteriores a x en el orden, tales que $g(v_k) = k$, .
- Observemos que $W_{i+1} = W_i \cup V_{\pi(i)}$ y que los vértices de W_i estan antes en el orden que los de $V_{\pi(i)}$, por lo tanto Greedy los coloreará primero.
- Por hipotesis inductiva, Greedy coloreará esos vértices de W_i con a lo sumo i colores.
- Por la forma que colorea Greedy, esos colores son $0, 1, ..., i - 1$.
- Como $g(x) = i + 1$, lo anterior implica que $x \notin W_i$.
- x debe por lo tanto estar en $V_{\pi(i)}$.

cont. prueba VIT

- Pero $g(v_i) = i > i - 1$, asi que v_i tampoco puede estar en W_i y debe estar en $V_{\pi(i)}$.

cont. prueba VIT

- Pero $g(v_i) = i > i - 1$, asi que v_i tampoco puede estar en W_i y debe estar en $V_{\pi(i)}$.
- Ahora bien, los v_k eran todos vecinos de x, asi que tenemos que v_i es vecino de x.

cont. prueba VIT

- Pero $g(v_i) = i > i - 1$, asi que v_i tampoco puede estar en W_i y debe estar en $V_{\pi(i)}$.
- Ahora bien, los v_k eran todos vecinos de x, asi que tenemos que v_i es vecino de x.
- Pero ambos estan en $V_{\pi(i)}$.

cont. prueba VIT

- Pero $g(v_i) = i > i - 1$, asi que v_i tampoco puede estar en W_i y debe estar en $V_{\pi(i)}$.
- Ahora bien, los v_k eran todos vecinos de x, asi que tenemos que v_i es vecino de x.
- Pero ambos estan en $V_{\pi(i)}$.
- Es decir, son vecinos entre si y estan en $V_{\pi(i)}$, pero por la definición de $V_{\pi(i)}$, debe ser $c(x) = c(v_i) = \pi(i)$, absurdo pues c era un coloreo propio.

cont. prueba VIT

- Pero $g(v_i) = i > i - 1$, asi que v_i tampoco puede estar en W_i y debe estar en $V_{\pi(i)}$.
- Ahora bien, los v_k eran todos vecinos de x, asi que tenemos que v_i es vecino de x.
- Pero ambos estan en $V_{\pi(i)}$.
- Es decir, son vecinos entre si y estan en $V_{\pi(i)}$, pero por la definición de $V_{\pi(i)}$, debe ser $c(x) = c(v_i) = \pi(i)$, absurdo pues c era un coloreo propio.
- Fin prueba VIT.

Corolario

Existe un ordenamiento de los vértices de G tal que Greedy colorea G con $\chi(G)$ colores.

Corolario

Existe un ordenamiento de los vértices de G tal que Greedy colorea G con $\chi(G)$ colores.

- Prueba: Por definición $\chi(G)$ es el menor k tal que existe un coloreo propio con k colores de G.

Corolario

Existe un ordenamiento de los vértices de G tal que Greedy colorea G con $\chi(G)$ colores.

- Prueba: Por definición $\chi(G)$ es el menor k tal que existe un coloreo propio con k colores de G.
- Sea c un coloreo propio de G con $k = \chi(G)$ colores.

Corolario

Existe un ordenamiento de los vértices de G tal que Greedy colorea G con $\chi(G)$ colores.

- Prueba: Por definición $\chi(G)$ es el menor k tal que existe un coloreo propio con k colores de G.
- Sea c un coloreo propio de G con $k = \chi(G)$ colores.
- Sea $V_i = \{x \in V : c(x) = i\}, i = 0, 1, ..., k - 1$.

Corolario

Existe un ordenamiento de los vértices de G tal que Greedy colorea G con $\chi(G)$ colores.

- Prueba: Por definición $\chi(G)$ es el menor k tal que existe un coloreo propio con k colores de G.
- Sea c un coloreo propio de G con $k = \chi(G)$ colores.
- Sea $V_i = \{x \in V : c(x) = i\}, i = 0, 1, ..., k - 1$.
- Ordenemos los vértices poniendo primero los vértices de V_0, luego los de V_1, etc.

Corolario

Existe un ordenamiento de los vértices de G tal que Greedy colorea G con $\chi(G)$ colores.

- Prueba: Por definición $\chi(G)$ es el menor k tal que existe un coloreo propio con k colores de G.
- Sea c un coloreo propio de G con $k = \chi(G)$ colores.
- Sea $V_i = \{x \in V : c(x) = i\}, i = 0, 1, ..., k-1$.
- Ordenemos los vértices poniendo primero los vértices de V_0, luego los de V_1, etc.
- Por el VIT, Greedy en ese orden no usa mas de k colores.

Corolario

Existe un ordenamiento de los vértices de G tal que Greedy colorea G con $\chi(G)$ colores.

- Prueba: Por definición $\chi(G)$ es el menor k tal que existe un coloreo propio con k colores de G.
- Sea c un coloreo propio de G con $k = \chi(G)$ colores.
- Sea $V_i = \{x \in V : c(x) = i\}, i = 0, 1, ..., k-1$.
- Ordenemos los vértices poniendo primero los vértices de V_0, luego los de V_1, etc.
- Por el VIT, Greedy en ese orden no usa mas de k colores.
- Dado que no hay coloreo propio con menos de k colores, Greedy usa exactamente k colores. Fin.

Consecuencia

- Podriamos entonces simplemente ordenar con todos los ordenes posibles y correr Greedy para obtener $\chi(G)$.

Consecuencia

- Podriamos entonces simplemente ordenar con todos los ordenes posibles y correr Greedy para obtener $\chi(G)$.
- Pero dado que hay $n!$ ordenes posibles, este algoritmo no es polinomial.

Consecuencia

- Podriamos entonces simplemente ordenar con todos los ordenes posibles y correr Greedy para obtener $\chi(G)$.
- Pero dado que hay $n!$ ordenes posibles, este algoritmo no es polinomial.
- Pero al menos, si no podemos obtener $\chi(G)$ polinomialmente, usaremos el VIT para tratar de obtener una aproximación a $\chi(G)$.

Consecuencia

- Podriamos entonces simplemente ordenar con todos los ordenes posibles y correr Greedy para obtener $\chi(G)$.
- Pero dado que hay n! ordenes posibles, este algoritmo no es polinomial.
- Pero al menos, si no podemos obtener $\chi(G)$ polinomialmente, usaremos el VIT para tratar de obtener una aproximación a $\chi(G)$.
- No siempre se puede, y hay grafos construidos especialmente para hacer fracasar a VIT, pero en la practica suele funcionar bastante bien, dependiendo de cuales permutaciones π se usen.

Consecuencia

- Podriamos entonces simplemente ordenar con todos los ordenes posibles y correr Greedy para obtener $\chi(G)$.
- Pero dado que hay $n!$ ordenes posibles, este algoritmo no es polinomial.
- Pero al menos, si no podemos obtener $\chi(G)$ polinomialmente, usaremos el VIT para tratar de obtener una aproximación a $\chi(G)$.
- No siempre se puede, y hay grafos construidos especialmente para hacer fracasar a VIT, pero en la practica suele funcionar bastante bien, dependiendo de cuales permutaciones π se usen.
- Parte del proyecto involucrará el VIT.

Grafos bipartitos

- Un grafo se dice bipartito si $\chi(G) = 2$.

Grafos bipartitos

- Un grafo se dice bipartito si $\chi(G) = 2$.
- El nombre viene de que si $\chi(G) = 2$ entonces tenemos un coloreo propio con 2 colores, y los vértices se "parten"en dos partes, (una de un color, otra de otro) de forma tal que los lados son únicamente entre vértices de partes distintas.

Grafos bipartitos

- Un grafo se dice bipartito si $\chi(G) = 2$.
- El nombre viene de que si $\chi(G) = 2$ entonces tenemos un coloreo propio con 2 colores, y los vértices se “parten”en dos partes, (una de un color, otra de otro) de forma tal que los lados son únicamente entre vértices de partes distintas.
- Es decir, si $G = (V, E)$ entonces existen $X, Y \subseteq V$ tales que:

Grafos bipartitos

- Un grafo se dice bipartito si $\chi(G) = 2$.
- El nombre viene de que si $\chi(G) = 2$ entonces tenemos un coloreo propio con 2 colores, y los vértices se "parten"en dos partes, (una de un color, otra de otro) de forma tal que los lados son únicamente entre vértices de partes distintas.
- Es decir, si $G = (V, E)$ entonces existen $X, Y \subseteq V$ tales que:
  1. $V = X \cup Y$.

Grafos bipartitos

- Un grafo se dice bipartito si $\chi(G) = 2$.
- El nombre viene de que si $\chi(G) = 2$ entonces tenemos un coloreo propio con 2 colores, y los vértices se “parten”en dos partes, (una de un color, otra de otro) de forma tal que los lados son únicamente entre vértices de partes distintas.
- Es decir, si $G = (V, E)$ entonces existen $X, Y \subseteq V$ tales que:
  1. $V = X \cup Y$.
  2. $X \cap Y = \emptyset$

Grafos bipartitos

- Un grafo se dice bipartito si $\chi(G) = 2$.
- El nombre viene de que si $\chi(G) = 2$ entonces tenemos un coloreo propio con 2 colores, y los vértices se “parten”en dos partes, (una de un color, otra de otro) de forma tal que los lados son únicamente entre vértices de partes distintas.
- Es decir, si $G = (V, E)$ entonces existen $X, Y \subseteq V$ tales que:
  1. $V = X \cup Y$.
  2. $X \cap Y = \emptyset$
  3. $wv \in E \Rightarrow (w \in X, v \in Y) \vee (w \in Y, v \in X)$

Grafos bipartitos

- Un grafo se dice bipartito si $\chi(G) = 2$.
- El nombre viene de que si $\chi(G) = 2$ entonces tenemos un coloreo propio con 2 colores, y los vértices se "parten"en dos partes, (una de un color, otra de otro) de forma tal que los lados son únicamente entre vértices de partes distintas.
- Es decir, si $G = (V, E)$ entonces existen $X, Y \subseteq V$ tales que:
  1. $V = X \cup Y$.
  2. $X \cap Y = \emptyset$
  3. $wv \in E \Rightarrow (w \in X, v \in Y) \vee (w \in Y, v \in X)$
- De hecho esta es otra definición posible, excepto que no es equivalente a la primera en el caso $E = \emptyset$.

Grafos bipartitos

- Un grafo se dice bipartito si $\chi(G) = 2$.
- El nombre viene de que si $\chi(G) = 2$ entonces tenemos un coloreo propio con 2 colores, y los vértices se “parten”en dos partes, (una de un color, otra de otro) de forma tal que los lados son únicamente entre vértices de partes distintas.
- Es decir, si $G = (V, E)$ entonces existen $X, Y \subseteq V$ tales que:
  1. $V = X \cup Y$.
  2. $X \cap Y = \emptyset$
  3. $wv \in E \Rightarrow (w \in X, v \in Y) \vee (w \in Y, v \in X)$
- De hecho esta es otra definición posible, excepto que no es equivalente a la primera en el caso $E = \emptyset$.
- A veces se toma como definición de bipartito esa definición, que equivale a decir $\chi(G) \leq 2$.

El problema 2COLOR

- El problema 2COLOR es:

El problema 2COLOR

- El problema 2COLOR es:
  - Dado un grafo G, ¿es $\chi(G) \leq 2$?

El problema 2COLOR

- El problema 2COLOR es:
  - Dado un grafo *G*, ¿es $\chi(G) \leq 2$?
- En realidad para todo *k* existen problemas *k*COLOR: “Dado un grafo *G*, ¿es $\chi(G) \leq k$?”pero por ahora nos concentraremos en el caso $k = 2$.

El problema 2COLOR

- El problema 2COLOR es:
  - Dado un grafo G, ¿es $\chi(G) \leq 2$?
- En realidad para todo k existen problemas kCOLOR: “Dado un grafo G, ¿es $\chi(G) \leq k$?”pero por ahora nos concentraremos en el caso $k = 2$.
- Queremos saber si se puede resolver el problema 2COLOR con un algoritmo polinomial.

El problema 2COLOR

- El problema 2COLOR es:
  - Dado un grafo G, ¿es $\chi(G) \leq 2$?
- En realidad para todo k existen problemas kCOLOR: “Dado un grafo G, ¿es $\chi(G) \leq k$?”pero por ahora nos concentraremos en el caso $k = 2$.
- Queremos saber si se puede resolver el problema 2COLOR con un algoritmo polinomial.
- Esto suele abreviarse diciendo “2COLOR es polinomial”

El problema 2COLOR

- El problema 2COLOR es:
  - Dado un grafo G, ¿es $\chi(G) \leq 2$?
- En realidad para todo k existen problemas kCOLOR: "Dado un grafo G, ¿es $\chi(G) \leq k$?"pero por ahora nos concentraremos en el caso $k = 2$.
- Queremos saber si se puede resolver el problema 2COLOR con un algoritmo polinomial.
- Esto suele abreviarse diciendo "2COLOR es polinomial"
- El algoritmo de buscar todos los coloreos posibles con 2 colores y ver si alguno es propio tiene complejidad $O(2^n m)$ asi que no nos sirve.

El problema 2COLOR

- El problema 2COLOR es:
  - Dado un grafo G, ¿es $\chi(G) \leq 2$?
- En realidad para todo k existen problemas kCOLOR: "Dado un grafo G, ¿es $\chi(G) \leq k$?"pero por ahora nos concentraremos en el caso $k = 2$.
- Queremos saber si se puede resolver el problema 2COLOR con un algoritmo polinomial.
- Esto suele abreviarse diciendo "2COLOR es polinomial"
- El algoritmo de buscar todos los coloreos posibles con 2 colores y ver si alguno es propio tiene complejidad $O(2^n m)$ asi que no nos sirve.
- Greedy es polinomial pero vimos un ejemplo donde G es bipartito pero Greedy lo colorea con $\frac{n}{2}$ colores, asi que no nos sirve.

Teorema

2COLOR es polinomial

Teorema

2COLOR es polinomial

- Prueba: Recordemos el concepto de nivel en un arbol con raiz:

Teorema

2COLOR es polinomial

- Prueba: Recordemos el concepto de nivel en un arbol con raiz:
- En un arbol con raiz x definimos el nivel de un vértice z de acuerdo con su distancia a x, es decir:

Teorema

2COLOR es polinomial

- Prueba: Recordemos el concepto de nivel en un arbol con raiz:
- En un arbol con raiz x definimos el nivel de un vértice z de acuerdo con su distancia a x, es decir:
  - Dado que es un arbol, es conexo, asi que entre dos vértices cualesquiera hay al menos un camino.

Teorema

2COLOR es polinomial

- Prueba: Recordemos el concepto de nivel en un arbol con raiz:
- En un arbol con raiz x definimos el nivel de un vértice z de acuerdo con su distancia a x, es decir:
  - Dado que es un arbol, es conexo, asi que entre dos vértices cualesquiera hay al menos un camino.
  - Puesto que es un arbol, no tiene ciclos, por lo tanto entre dos vértices cualesquiera hay UN SOLO camino. (si hubiera 2, crearian un ciclo).

Teorema

2COLOR es polinomial

- Prueba: Recordemos el concepto de nivel en un arbol con raiz:
- En un arbol con raiz x definimos el nivel de un vértice z de acuerdo con su distancia a x, es decir:
  - Dado que es un arbol, es conexo, asi que entre dos vértices cualesquiera hay al menos un camino.
  - Puesto que es un arbol, no tiene ciclos, por lo tanto entre dos vértices cualesquiera hay UN SOLO camino. (si hubiera 2, crearian un ciclo).
- Por lo tanto, tomamos el único camino entre z y x, contamos cuantos lados hay en ese camino, y ese es el nivel de z en el arbol.

- Pej, x tiene nivel 0. Sus vecinos en el arbol tienen nivel 1. Los vecinos de los vecinos, nivel 2, etc.

- Pej, x tiene nivel 0. Sus vecinos en el arbol tienen nivel 1. Los vecinos de los vecinos, nivel 2, etc.
- Observemos que $\chi(G) \leq 2$ si y solo si $\chi(C) \leq 2$ para toda componente conexa C de G, pues al no haber caminos entre componentes conexas, cada componente conexa se puede colorear por separado.

- Pej, x tiene nivel 0. Sus vecinos en el arbol tienen nivel 1. Los vecinos de los vecinos, nivel 2, etc.
- Observemos que $\chi(G) \leq 2$ si y solo si $\chi(C) \leq 2$ para toda componente conexa C de G, pues al no haber caminos entre componentes conexas, cada componente conexa se puede colorear por separado.
- Por lo tanto para ver que 2COLOR es polinomial basta dar un algoritmo polinomial que determine si $\chi(G) \leq 2$ para el caso G conexo.

- Pej, x tiene nivel 0. Sus vecinos en el arbol tienen nivel 1. Los vecinos de los vecinos, nivel 2, etc.
- Observemos que $\chi(G) \leq 2$ si y solo si $\chi(C) \leq 2$ para toda componente conexa C de G, pues al no haber caminos entre componentes conexas, cada componente conexa se puede colorear por separado.
- Por lo tanto para ver que 2COLOR es polinomial basta dar un algoritmo polinomial que determine si $\chi(G) \leq 2$ para el caso G conexo.
- Si G es conexo, tanto $BFS(x)$ como $DFS(x)$ encuentran todos los vertices de G, y como explicamos la clase pasada, si se agregan los lados entre un vértice que ya estaba agregado y los vértices que ese vértice agregó se obtiene un arbol.

- Pej, x tiene nivel 0. Sus vecinos en el arbol tienen nivel 1. Los vecinos de los vecinos, nivel 2, etc.
- Observemos que $\chi(G) \leq 2$ si y solo si $\chi(C) \leq 2$ para toda componente conexa C de G, pues al no haber caminos entre componentes conexas, cada componente conexa se puede colorear por separado.
- Por lo tanto para ver que 2COLOR es polinomial basta dar un algoritmo polinomial que determine si $\chi(G) \leq 2$ para el caso G conexo.
- Si G es conexo, tanto $BFS(x)$ como $DFS(x)$ encuentran todos los vertices de G, y como explicamos la clase pasada, si se agregan los lados entre un vértice que ya estaba agregado y los vértices que ese vértice agregó se obtiene un arbol.
- Usaremos concretamente BFS.

Algoritmo 2COLOR para G conexo.

1. Elegir un vértice x cualquiera.

Algoritmo 2COLOR para G conexo.

1. Elegir un vértice x cualquiera.
2. Correr BFS(x), creando un arbol.

Algoritmo 2COLOR para G conexo.

1. Elegir un vértice x cualquiera.
2. Correr BFS(x), creando un arbol.
3. Para cada vértice z, sea $N(z)$ el nivel de z en el arbol BFS(x).

Algoritmo 2COLOR para G conexo.

1. Elegir un vértice x cualquiera.
2. Correr BFS(x), creando un arbol.
3. Para cada vértice z, sea $N(z)$ el nivel de z en el arbol BFS(x).
4. Colorear $c(z) = (N(z) \mod 2)$.

Algoritmo 2COLOR para G conexo.

1. Elegir un vértice x cualquiera.
2. Correr BFS(x), creando un arbol.
3. Para cada vértice z, sea $N(z)$ el nivel de z en el arbol BFS(x).
4. Colorear $c(z) = (N(z) \mod 2)$.
5. Chequear si el colorario dado en [4] es propio.

Algoritmo 2COLOR para G conexo.

1. Elegir un vértice x cualquiera.
2. Correr BFS(x), creando un arbol.
3. Para cada vértice z, sea $N(z)$ el nivel de z en el arbol BFS(x).
4. Colorear $c(z) = (N(z) \mod 2)$.
5. Chequear si el colorario dado en [4] es propio.
6. Si lo es, retornar "$\chi(G) \leq 2$"

Algoritmo 2COLOR para G conexo.

1. Elegir un vértice x cualquiera.
2. Correr BFS(x), creando un arbol.
3. Para cada vértice z, sea $N(z)$ el nivel de z en el arbol BFS(x).
4. Colorear $c(z) = (N(z) \mod 2)$.
5. Chequear si el colorario dado en [4] es propio.
6. Si lo es, retornar "$\chi(G) \leq 2$"
7. Si no lo es, retornar "$\chi(G) > 2$"

Complejidad

- Tenemos que probar que ese algoritmo es correcto (es decir, siempre da la respuesta correcta) y que es polinomial.

Complejidad

- Tenemos que probar que ese algoritmo es correcto (es decir, siempre da la respuesta correcta) y que es polinomial.
- Empezemos por lo segundo:

Complejidad

- Tenemos que probar que ese algoritmo es correcto (es decir, siempre da la respuesta correcta) y que es polinomial.
- Empezemos por lo segundo:
  - Crear BFS(x) es $O(m)$.

Complejidad

- Tenemos que probar que ese algoritmo es correcto (es decir, siempre da la respuesta correcta) y que es polinomial.
- Empezemos por lo segundo:
  - Crear BFS(x) es $O(m)$.
  - Los niveles y el coloreo asociado se pueden ir calculando a medida que hacemos BFS(x), basicamente, si un vértice u que está en el arbol agrega a un vértice v que no está, entonces coloreamos a v con 1 si el color de u era 0 y con 0 si el color de u era 1.

Complejidad

- Tenemos que probar que ese algoritmo es correcto (es decir, siempre da la respuesta correcta) y que es polinomial.
- Empezemos por lo segundo:
  - Crear BFS(x) es $O(m)$.
  - Los niveles y el coloreo asociado se pueden ir calculando a medida que hacemos BFS(x), basicamente, si un vértice u que está en el arbol agrega a un vértice v que no está, entonces coloreamos a v con 1 si el color de u era 0 y con 0 si el color de u era 1.
  - Asi que el coloreo es "gratis", su complejidad esté metida dentro de la complejidad $O(m)$ de construir BFS.

Complejidad

- Tenemos que probar que ese algoritmo es correcto (es decir, siempre da la respuesta correcta) y que es polinomial.
- Empezemos por lo segundo:
  - Crear BFS(x) es $O(m)$.
  - Los niveles y el coloreo asociado se pueden ir calculando a medida que hacemos BFS(x), basicamente, si un vértice u que está en el arbol agrega a un vértice v que no está, entonces coloreamos a v con 1 si el color de u era 0 y con 0 si el color de u era 1.
  - Asi que el coloreo es "gratis", su complejidad esté metida dentro de la complejidad $O(m)$ de construir BFS.
  - Chequear que un coloreo sea propio es $O(m)$.

Complejidad

- Tenemos que probar que ese algoritmo es correcto (es decir, siempre da la respuesta correcta) y que es polinomial.
- Empezemos por lo segundo:
  - Crear BFS(x) es $O(m)$.
  - Los niveles y el coloreo asociado se pueden ir calculando a medida que hacemos BFS(x), basicamente, si un vértice u que está en el arbol agrega a un vértice v que no está, entonces coloreamos a v con 1 si el color de u era 0 y con 0 si el color de u era 1.
  - Asi que el coloreo es “gratis”, su complejidad esté metida dentro de la complejidad $O(m)$ de construir BFS.
  - Chequear que un coloreo sea propio es $O(m)$.
  - Asi que la complejidad total es $O(m) + O(m) = O(m)$.

Complejidad

- Tenemos que probar que ese algoritmo es correcto (es decir, siempre da la respuesta correcta) y que es polinomial.
- Empezemos por lo segundo:
  - Crear BFS(x) es $O(m)$.
  - Los niveles y el coloreo asociado se pueden ir calculando a medida que hacemos BFS(x), basicamente, si un vértice u que está en el arbol agrega a un vértice v que no está, entonces coloreamos a v con 1 si el color de u era 0 y con 0 si el color de u era 1.
  - Asi que el coloreo es "gratis", su complejidad esté metida dentro de la complejidad $O(m)$ de construir BFS.
  - Chequear que un coloreo sea propio es $O(m)$.
  - Asi que la complejidad total es $O(m) + O(m) = O(m)$.
- Ahora la correctitud:

Correctitud

- Obviamente si la respuesta que devuelve el algoritmo es "$\chi(G) \leq 2$", entonces esa respuesta es correcta, pues el algoritmo sólo devuelve esa respuesta si efectivamente el coloreo con 2 colores propuestos es un coloreo propio.

Correctitud

- Obviamente si la respuesta que devuelve el algoritmo es "$\chi(G) \leq 2$", entonces esa respuesta es correcta, pues el algoritmo sólo devuelve esa respuesta si efectivamente el coloreo con 2 colores propuestos es un coloreo propio.
- Asi que nos queda ver que cuando devuelve la respuesta "$\chi(G) > 2$", esa respuesta es correcta.

Correctitud

- Obviamente si la respuesta que devuelve el algoritmo es "$\chi(G) \leq 2$", entonces esa respuesta es correcta, pues el algoritmo sólo devuelve esa respuesta si efectivamente el coloreo con 2 colores propuestos es un coloreo propio.
- Asi que nos queda ver que cuando devuelve la respuesta "$\chi(G) > 2$", esa respuesta es correcta.
- Lo que tenemos que probar es que si devuelve "$\chi(G) > 2$" entonces no hay ningún coloreo con 2 colores.

Correctitud

- Obviamente si la respuesta que devuelve el algoritmo es "$\chi(G) \leq 2$", entonces esa respuesta es correcta, pues el algoritmo sólo devuelve esa respuesta si efectivamente el coloreo con 2 colores propuestos es un coloreo propio.
- Asi que nos queda ver que cuando devuelve la respuesta "$\chi(G) > 2$", esa respuesta es correcta.
- Lo que tenemos que probar es que si devuelve "$\chi(G) > 2$" entonces no hay ningún coloreo con 2 colores.
- Obvervemos que esa respuesta es devuelta sólo si ese coloreo particular del algoritmo no es propio.

Correctitud

- Obviamente si la respuesta que devuelve el algoritmo es "$\chi(G) \leq 2$", entonces esa respuesta es correcta, pues el algoritmo sólo devuelve esa respuesta si efectivamente el coloreo con 2 colores propuestos es un coloreo propio.
- Asi que nos queda ver que cuando devuelve la respuesta "$\chi(G) > 2$", esa respuesta es correcta.
- Lo que tenemos que probar es que si devuelve "$\chi(G) > 2$" entonces no hay ningún coloreo con 2 colores.
- Obvervemos que esa respuesta es devuelta sólo si ese coloreo particular del algoritmo no es propio.
- Entonces tenemos que probar que si ese coloreo no es propio, ningún otro coloreo con 2 colores es propio.

Correctitud (cont.)

- Supongamos entonces que el coloreo no es propio.

Correctitud (cont.)

- Supongamos entonces que el coloreo no es propio.
- Esto implica que existen z, y $(z \neq y)$ tales que $c(z) = c(y)$ pero zy es un lado en G.

Correctitud (cont.)

- Supongamos entonces que el coloreo no es propio.
- Esto implica que existen z, y $(z \neq y)$ tales que $c(z) = c(y)$ pero zy es un lado en G.
- Sea $z_0 z_1 .... z_k$ el único camino entre x y z en el arbol BFS $(z_0 = x, z_k = z)$.

Correctitud (cont.)

- Supongamos entonces que el coloreo no es propio.
- Esto implica que existen z, y $(z \neq y)$ tales que $c(z) = c(y)$ pero zy es un lado en G.
- Sea $z_0 z_1 \ldots. z_k$ el único camino entre x y z en el arbol BFS $(z_0 = x, z_k = z)$.
- Sea $y_0 y_1 \ldots. y_j$ el único camino entre x e y en el arbol BFS $(y_0 = x, y_j = y)$.

Correctitud (cont.)

- Supongamos entonces que el coloreo no es propio.
- Esto implica que existen z, y ($z \neq y$) tales que $c(z) = c(y)$ pero zy es un lado en G.
- Sea $z_0 z_1 .... z_k$ el único camino entre x y z en el arbol BFS ($z_0 = x, z_k = z$).
- Sea $y_0 y_1 .... y_j$ el único camino entre x e y en el arbol BFS ($y_0 = x, y_j = y$).
- Entonces k es igual al nivel de z y j igual al nivel de y.

Correctitud (cont.)

- Supongamos entonces que el coloreo no es propio.
- Esto implica que existen z, y ($z \neq y$) tales que $c(z) = c(y)$ pero zy es un lado en G.
- Sea $z_0 z_1 ....z_k$ el único camino entre x y z en el arbol BFS ($z_0 = x, z_k = z$).
- Sea $y_0 y_1 ....y_j$ el único camino entre x e y en el arbol BFS ($y_0 = x, y_j = y$).
- Entonces k es igual al nivel de z y j igual al nivel de y.
- Por lo tanto $c(z) = (k \mod 2)$ y $c(y) = (j \mod 2)$

Correctitud (cont.)

- Supongamos entonces que el coloreo no es propio.
- Esto implica que existen z, y $(z \neq y)$ tales que $c(z) = c(y)$ pero zy es un lado en G.
- Sea $z_0 z_1 \dots . z_k$ el único camino entre x y z en el arbol BFS $(z_0 = x, z_k = z)$.
- Sea $y_0 y_1 \dots . y_j$ el único camino entre x e y en el arbol BFS $(y_0 = x, y_j = y)$.
- Entonces k es igual al nivel de z y j igual al nivel de y.
- Por lo tanto $c(z) = (k \mod 2)$ y $c(y) = (j \mod 2)$
- Como $c(z) = c(y)$ concluimos que $(k \mod 2) = (j \mod 2)$, es decir son ambos pares o ambos impares.

Correctitud (cont.)

- Supongamos entonces que el coloreo no es propio.
- Esto implica que existen z, y ($z \neq y$) tales que $c(z) = c(y)$ pero zy es un lado en G.
- Sea $z_0 z_1 \dots . z_k$ el único camino entre x y z en el arbol BFS ($z_0 = x, z_k = z$).
- Sea $y_0 y_1 \dots . y_j$ el único camino entre x e y en el arbol BFS ($y_0 = x, y_j = y$).
- Entonces k es igual al nivel de z y j igual al nivel de y.
- Por lo tanto $c(z) = (k \mod 2)$ y $c(y) = (j \mod 2)$
- Como $c(z) = c(y)$ concluimos que $(k \mod 2) = (j \mod 2)$, es decir son ambos pares o ambos impares.
- Por lo tanto la suma $k + j$ es PAR.

Correctitud (cont.)

- Esos dos caminos comienzan igual (con x) pero terminan distinto (con $z \neq y$).

Correctitud (cont.)

- Esos dos caminos comienzan igual (con x) pero terminan distinto (con $z \neq y$).
- En algun lado hay un último vértice en común entre esos dos caminos.

Correctitud (cont.)

- Esos dos caminos comienzan igual (con x) pero terminan distinto (con $z \neq y$).
- En algun lado hay un último vértice en común entre esos dos caminos.
- Sea w ese último vértice en común.

Correctitud (cont.)

- Esos dos caminos comienzan igual (con x) pero terminan distinto (con $z \neq y$).
- En algun lado hay un último vértice en común entre esos dos caminos.
- Sea w ese último vértice en común.
- Los caminos son iguales hasta ahi, asi que existe p con $z_0 = y_0, z_1 = y_1, ..., z_p = y_p = w$

Correctitud (cont.)

- Esos dos caminos comienzan igual (con x) pero terminan distinto (con $z \neq y$).
- En algun lado hay un último vértice en común entre esos dos caminos.
- Sea w ese último vértice en común.
- Los caminos son iguales hasta ahi, asi que existe p con $z_0 = y_0, z_1 = y_1, \ldots, z_p = y_p = w$
- Teniendo en cuenta que xy es un lado en G y que $z_k = z$, $y_j = y$, entonces en G tenemos el ciclo $C: \; wz_{p+1}z_{p+2} \ldots z_{k-1}zyy_{j-1} \ldots y_{p+1}w$.

Correctitud(cont.)

- Veamos cuantos vértices hay en ese ciclo

Correctitud(cont.)

- Veamos cuantos vértices hay en ese ciclo
  1. w, suma 1.

Correctitud(cont.)

- Veamos cuantos vértices hay en ese ciclo
  1. w, suma 1.
  2. los $z_{p+1}, z_{p+2}, ..., z_{k-1} z$ son $k - p$.

Correctitud(cont.)

- Veamos cuantos vértices hay en ese ciclo
  1. w, suma 1.
  2. los $z_{p+1}, z_{p+2}, ..., z_{k-1}z$ son $k-p$.
  3. Los $y, y_{j-1}, ..., y_{p+1}$ son $j-p$.

Correctitud(cont.)

- Veamos cuantos vértices hay en ese ciclo
  1. w, suma 1.
  2. los $z_{p+1}, z_{p+2}, ..., z_{k-1} z$ son $k-p$.
  3. Los $y, y_{j-1}, ..., y_{p+1}$ son $j-p$.
- El total son $1+k-p+j-p = k+j-2p+1$.

Correctitud(cont.)

- Veamos cuantos vértices hay en ese ciclo
  1. w, suma 1.
  2. los $z_{p+1}, z_{p+2}, ..., z_{k-1} z$ son $k-p$.
  3. Los $y, y_{j-1}, ..., y_{p+1}$ son $j-p$.
- El total son $1+k-p+j-p = k+j-2p+1$.
- Como vimos antes que $k+j$ es par y $2p$ es par, entonces $k+j-2p$ es par.

Correctitud(cont.)

- Veamos cuantos vértices hay en ese ciclo
  1. w, suma 1.
  2. los $z_{p+1}, z_{p+2}, ..., z_{k-1} z$ son $k - p$.
  3. Los $y, y_{j-1}, ..., y_{p+1}$ son $j - p$.
- El total son $1 + k - p + j - p = k + j - 2p + 1$.
- Como vimos antes que $k + j$ es par y $2p$ es par, entonces $k + j - 2p$ es par.
- Por lo tanto $k + j - 2p + 1$ es IMPAR.

Correctitud(cont.)

- Veamos cuantos vértices hay en ese ciclo
  1. w, suma 1.
  2. los $z_{p+1}, z_{p+2}, ..., z_{k-1} z$ son $k - p$.
  3. Los $y, y_{j-1}, ..., y_{p+1}$ son $j - p$.
- El total son $1 + k - p + j - p = k + j - 2p + 1$.
- Como vimos antes que $k + j$ es par y $2p$ es par, entonces $k + j - 2p$ es par.
- Por lo tanto $k + j - 2p + 1$ es IMPAR.
- Entonces G tiene adentro un ciclo impar, por lo tanto $\chi(G) \geq 3 > 2$ y la respuesta es correcta.

Correctitud(cont.)

- Veamos cuantos vértices hay en ese ciclo
  1. w, suma 1.
  2. los $z_{p+1}, z_{p+2}, ..., z_{k-1}z$ son $k-p$.
  3. Los $y, y_{j-1}, ..., y_{p+1}$ son $j-p$.
- El total son $1+k-p+j-p = k+j-2p+1$.
- Como vimos antes que $k+j$ es par y $2p$ es par, entonces $k+j-2p$ es par.
- Por lo tanto $k+j-2p+1$ es IMPAR.
- Entonces G tiene adentro un ciclo impar, por lo tanto $\chi(G) \geq 3 > 2$ y la respuesta es correcta.
- Fin prueba.

Corolario (de la prueba)

- Sea G un grafo con $\chi(G) \geq 3$.

Corolario (de la prueba)

- Sea G un grafo con $\chi(G) \geq 3$.
- Corramos el algoritmo sobre G.

Corolario (de la prueba)

- Sea G un grafo con $\chi(G) \geq 3$.
- Corramos el algoritmo sobre G.
- Como $\chi(G) \geq 3$, el coloreo de 2 colores dado en el algoritmo no puede ser propio.

Corolario (de la prueba)

- Sea G un grafo con $\chi(G) \geq 3$.
- Corramos el algoritmo sobre G.
- Como $\chi(G) \geq 3$, el coloreo de 2 colores dado en el algoritmo no puede ser propio.
- En la demostración de la correctitud del algoritmo vimos que en este caso, podemos construir un ciclo impar en G.

Corolario (de la prueba)

- Sea G un grafo con $\chi(G) \geq 3$.
- Corramos el algoritmo sobre G.
- Como $\chi(G) \geq 3$, el coloreo de 2 colores dado en el algoritmo no puede ser propio.
- En la demostración de la correctitud del algoritmo vimos que en este caso, podemos construir un ciclo impar en G.
- Conclusión: $\chi(G) \geq 3 \Rightarrow$ existe un ciclo impar en G.

Corolario (de la prueba)

- Sea G un grafo con $\chi(G) \geq 3$.
- Corramos el algoritmo sobre G.
- Como $\chi(G) \geq 3$, el coloreo de 2 colores dado en el algoritmo no puede ser propio.
- En la demostración de la correctitud del algoritmo vimos que en este caso, podemos construir un ciclo impar en G.
- Conclusión: $\chi(G) \geq 3 \Rightarrow$ existe un ciclo impar en G.
- Como ya sabiamos la implicación para el otro lado, podemos decir que $\chi(G) \geq 3$ si y solo si existe un ciclo impar en G.