Apareamientos

Miguel Raggi

Algoritmos en grafos
Escuela Nacional de Estudios Superiores
UNAM

1 de abril de 2018

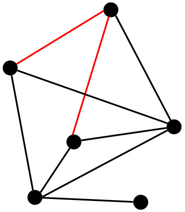
Índice:

Índice:

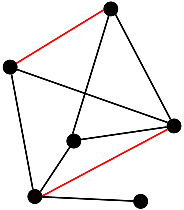
Apareamientos

Definición

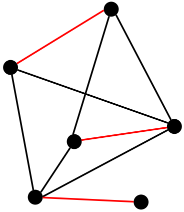
Un apareamiento *de una gráfica G es un subconjunto M de las aristas tal que cualesquiera dos aristas de M no comparten vértice. Un apareamiento es* perfecto *si las aristas de M utilizan todos los vértices de G. Un apareamiento es* máximo *si utiliza el máximo número de aristas de entre todos los apareamientos.*



nada



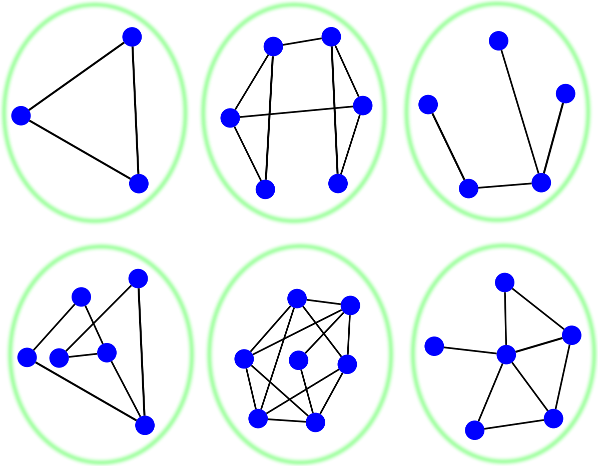
apareamiento



apareamiento perfecto

Ejercicios

Encuentra un apareamiento máximo en las siguientes gráficas. ¿cuándo son perfectos?



Preguntas

- ¿Cuándo tiene apareamiento perfecto K_n?

Preguntas

- ¿Cuándo tiene apareamiento perfecto K_n?
- ¿Cuándo tiene apareamiento perfecto C_n?

Preguntas

- ¿Cuándo tiene apareamiento perfecto K_n?
- ¿Cuándo tiene apareamiento perfecto C_n?
- ¿Cuándo tiene apareamiento perfecto P_n?

Preguntas

- ¿Cuándo tiene apareamiento perfecto K_n?
- ¿Cuándo tiene apareamiento perfecto C_n?
- ¿Cuándo tiene apareamiento perfecto P_n?
- ¿Puedes tener más de un apareamiento perfecto?

Preguntas

- ¿Cuándo tiene apareamiento perfecto K_n?
- ¿Cuándo tiene apareamiento perfecto C_n?
- ¿Cuándo tiene apareamiento perfecto P_n?
- ¿Puedes tener más de un apareamiento perfecto?
- Si G es subgráfica de H y H tiene apareamiento perfecto, entonces, ¿G también?

Preguntas

- ¿Cuándo tiene apareamiento perfecto K_n?
- ¿Cuándo tiene apareamiento perfecto C_n?
- ¿Cuándo tiene apareamiento perfecto P_n?
- ¿Puedes tener más de un apareamiento perfecto?
- Si G es subgráfica de H y H tiene apareamiento perfecto, entonces, ¿G también?
- Si agregas aristas a G, ¿qué pasa?

Aplicaciones

- Encontrar apareamientos máximos es un problema importante, por ejemplo en química.

Aplicaciones

- Encontrar apareamientos máximos es un problema importante, por ejemplo en química.
- Veremos algoritmos que resuelven este problema en diversas situaciones.

Aplicaciones

- Encontrar apareamientos máximos es un problema importante, por ejemplo en química.
- Veremos algoritmos que resuelven este problema en diversas situaciones.
- Empezaremos con apareamientos en gráficas bipartitas y luego los vemos en gráficas generales.

Índice:

Apareamientos en bipartitas: Imaginación

- Imaginemos que hay un grupito de niños y hay varios juguetes con los cuales quieren jugar.

Apareamientos en bipartitas: Imaginación

- Imaginemos que hay un grupito de niños y hay varios juguetes con los cuales quieren jugar.
- Cada juguete puede ser utilizado por a lo más un niño.

Apareamientos en bipartitas: Imaginación

- Imaginemos que hay un grupito de niños y hay varios juguetes con los cuales quieren jugar.
- Cada juguete puede ser utilizado por a lo más un niño.
- Además, cada niño tiene ciertas preferencias por los juguetes. Es decir, cada niño tiene un conjunto de juguetes con los cuales puede jugar.

Apareamientos en bipartitas: Imaginación

- Imaginemos que hay un grupito de niños y hay varios juguetes con los cuales quieren jugar.
- Cada juguete puede ser utilizado por a lo más un niño.
- Además, cada niño tiene ciertas preferencias por los juguetes. Es decir, cada niño tiene un conjunto de juguetes con los cuales puede jugar.
- ¿Bajo qué condiciones hay un acomodo de los juguetes en el que cada niño tiene un juguete?

Apareamientos en bipartitas: Imaginación

- Imaginemos que hay un grupito de niños y hay varios juguetes con los cuales quieren jugar.
- Cada juguete puede ser utilizado por a lo más un niño.
- Además, cada niño tiene ciertas preferencias por los juguetes. Es decir, cada niño tiene un conjunto de juguetes con los cuales puede jugar.
- ¿Bajo qué condiciones hay un acomodo de los juguetes en el que cada niño tiene un juguete?
- O con hombres y mujeres, etc.

Apareamientos en gráficas bipartitas

- Cuando tenemos una gráfica bipartita con partes X y Y, un apareamiento consta de aparear vértices de X con vértices de Y.

Apareamientos en gráficas bipartitas

- Cuando tenemos una gráfica bipartita con partes X y Y, un apareamiento consta de aparear vértices de X con vértices de Y.
- Entonces un apareamiento máximo tendrá que tener, cuando mucho, $\min(|X|, |Y|)$.

Apareamientos en gráficas bipartitas

- Cuando tenemos una gráfica bipartita con partes X y Y, un apareamiento consta de aparear vértices de X con vértices de Y.
- Entonces un apareamiento máximo tendrá que tener, cuando mucho, $\min(|X|, |Y|)$.
- En lo que resta de esta sección vamos a suponer SPG que $|X| \leq |Y|$.

Apareamientos en gráficas bipartitas

- Cuando tenemos una gráfica bipartita con partes X y Y, un apareamiento consta de aparear vértices de X con vértices de Y.
- Entonces un apareamiento máximo tendrá que tener, cuando mucho, $\min(|X|, |Y|)$.
- En lo que resta de esta sección vamos a suponer SPG que $|X| \leq |Y|$.
- Voy a llamarle X-perfecto a un apareamiento en el cuál todos los vértices de X estén apareados, aunque quizás no todos los de Y.

Apareamientos en gráficas bipartitas

- Cuando tenemos una gráfica bipartita con partes X y Y, un apareamiento consta de aparear vértices de X con vértices de Y.
- Entonces un apareamiento máximo tendrá que tener, cuando mucho, $\text{mín}(|X|, |Y|)$.
- En lo que resta de esta sección vamos a suponer SPG que $|X| \leq |Y|$.
- Voy a llamarle X-perfecto a un apareamiento en el cuál todos los vértices de X estén apareados, aunque quizás no todos los de Y.
- ¿Cuándo hay un apareamiento X-perfecto en una gráfica bipartita?

Apareamientos en gráficas bipartitas

- Cuando tenemos una gráfica bipartita con partes X y Y, un apareamiento consta de aparear vértices de X con vértices de Y.
- Entonces un apareamiento máximo tendrá que tener, cuando mucho, $\text{mín}(|X|, |Y|)$.
- En lo que resta de esta sección vamos a suponer SPG que $|X| \leq |Y|$.
- Voy a llamarle X-perfecto a un apareamiento en el cuál todos los vértices de X estén apareados, aunque quizás no todos los de Y.
- ¿Cuándo hay un apareamiento X-perfecto en una gráfica bipartita?
- El Teorema de Hall nos da condiciones necesarias y suficientes.

Apareamientos en gráficas bipartitas

- Cuando tenemos una gráfica bipartita con partes X y Y, un apareamiento consta de aparear vértices de X con vértices de Y.
- Entonces un apareamiento máximo tendrá que tener, cuando mucho, $\text{mín}(|X|, |Y|)$.
- En lo que resta de esta sección vamos a suponer SPG que $|X| \leq |Y|$.
- Voy a llamarle X-perfecto a un apareamiento en el cuál todos los vértices de X estén apareados, aunque quizás no todos los de Y.
- ¿Cuándo hay un apareamiento X-perfecto en una gráfica bipartita?
- El Teorema de Hall nos da condiciones necesarias y suficientes.
- Para cada $S \subset X$, defino $N(S)$ como el conjunto de vecinos de los vértices de S. Es decir, el conjunto de los $y \in Y$ tales que por lo menos tienen una arista con algún vértice en S.

Teorema de Hall

Teorema

Sea G una gráfica bipartita con partes X y Y. Entonces existe un apareamiento X-perfecto si y sólo si para todo $S \subset X$ se tiene que $|S| \leq |N(S)|$

Teorema de Hall

Teorema

Sea G una gráfica bipartita con partes X y Y. Entonces existe un apareamiento X-perfecto si y sólo si para todo $S \subset X$ se tiene que $|S| \leq |N(S)|$

- Es obvio que para poder tener un apareamiento X perfecto se debe satisfacer la condición de que siempre $|S| \leq |N(S)|$.

Teorema de Hall

Teorema

Sea G una gráfica bipartita con partes X y Y. Entonces existe un apareamiento X-perfecto si y sólo si para todo $S \subset X$ se tiene que $|S| \leq |N(S)|$

- Es obvio que para poder tener un apareamiento X perfecto se debe satisfacer la condición de que siempre $|S| \leq |N(S)|$.
- Para probar la otra parte, necesitamos un poco más. Como la prueba ayuda al algoritmo, la pondré aquí.

Caminos de Aumento

- Sea G una gráfica simple y sea M un apareamiento.

Caminos de Aumento

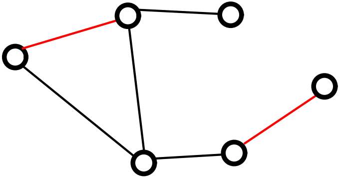
- Sea G una gráfica simple y sea M un apareamiento.
- Un camino alternante es una trayectoria en G en donde la primera arista no está en M, la segunda sí, la tercera no, etc. (o al revés)

Caminos de Aumento

- Sea G una gráfica simple y sea M un apareamiento.
- Un camino alternante es una trayectoria en G en donde la primera arista no está en M, la segunda sí, la tercera no, etc. (o al revés)
- Un camino de aumento es un camino alternante máximo en donde la primera y última aristas no están en M.

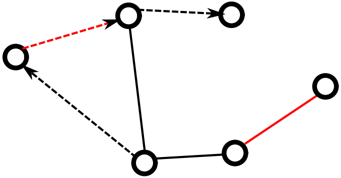
Caminos de Aumento

- Sea G una gráfica simple y sea M un apareamiento.
- Un camino alternante es una trayectoria en G en donde la primera arista no está en M, la segunda sí, la tercera no, etc. (o al revés)
- Un camino de aumento es un camino alternante máximo en donde la primera y última aristas no están en M.



Caminos de Aumento

- Sea G una gráfica simple y sea M un apareamiento.
- Un camino alternante es una trayectoria en G en donde la primera arista no está en M, la segunda sí, la tercera no, etc. (o al revés)
- Un camino de aumento es un camino alternante máximo en donde la primera y última aristas no están en M.



Caminos de Aumento

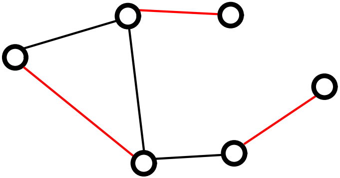
- Supongamos que M es un apareamiento y P es un camino de aumento.

Caminos de Aumento

- Supongamos que M es un apareamiento y P es un camino de aumento.
- Podemos intercambiar las aristas en P que no están en M con las que sí lo están y encontrar un apareamiento con más aristas:

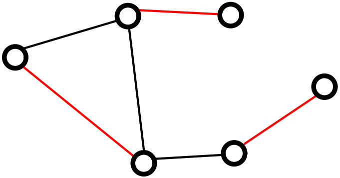
Caminos de Aumento

- Supongamos que M es un apareamiento y P es un camino de aumento.
- Podemos intercambiar las aristas en P que no están en M con las que sí lo están y encontrar un apareamiento con más aristas:



Caminos de Aumento

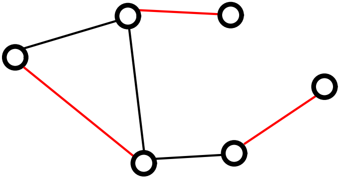
- Supongamos que M es un apareamiento y P es un camino de aumento.
- Podemos intercambiar las aristas en P que no están en M con las que sí lo están y encontrar un apareamiento con más aristas:



- Es decir, si M es un apareamiento máximo, entonces no puede existir camino de aumento.

Caminos de Aumento

- Supongamos que M es un apareamiento y P es un camino de aumento.
- Podemos intercambiar las aristas en P que no están en M con las que sí lo están y encontrar un apareamiento con más aristas:



- Es decir, si M es un apareamiento máximo, entonces no puede existir camino de aumento.
- Probaremos el regreso: Si M no es máximo, entonces sí existe camino de aumento.

Caminos de Aumento

Teorema

Sea M un apareamiento en G. Entonces M es máximo si y sólo si no existen caminos de aumento.

Caminos de Aumento

Teorema

Sea M un apareamiento en G. Entonces M es máximo si y sólo si no existen caminos de aumento.

Demostración: Sea M apareamiento donde no hay camino de aumento.

Caminos de Aumento

Teorema

Sea M un apareamiento en G. Entonces M es máximo si y sólo si no existen caminos de aumento.

Demostración: Sea M apareamiento donde no hay camino de aumento.

- Supongamos que existe M' apareamiento con $|M'| > |M|$.

Caminos de Aumento

Teorema

Sea M un apareamiento en G. Entonces M es máximo si y sólo si no existen caminos de aumento.

Demostración: Sea M apareamiento donde no hay camino de aumento.

- Supongamos que existe M' apareamiento con $|M'| > |M|$.
- En G, coloreamos de rojo las aristas que están en M, de azúl las aristas que están en M' y de morado las de ambas.

Caminos de Aumento

Teorema

Sea M un apareamiento en G. Entonces M es máximo si y sólo si no existen caminos de aumento.

Demostración: Sea M apareamiento donde no hay camino de aumento.

- Supongamos que existe M' apareamiento con $|M'| > |M|$.
- En G, coloreamos de rojo las aristas que están en M, de azúl las aristas que están en M' y de morado las de ambas.
- Borramos el resto de las aristas de G.

Caminos de Aumento

Teorema

Sea M un apareamiento en G. Entonces M es máximo si y sólo si no existen caminos de aumento.

Demostración: Sea M apareamiento donde no hay camino de aumento.

- Supongamos que existe M' apareamiento con $|M'| > |M|$.
- En G, coloreamos de rojo las aristas que están en M, de azúl las aristas que están en M' y de morado las de ambas.
- Borramos el resto de las aristas de G.
- Las aristas moradas no nos interesan: Simplemente son aristas sueltas no conectadas a nada.

Caminos de Aumento

Teorema

Sea M un apareamiento en G. Entonces M es máximo si y sólo si no existen caminos de aumento.

Demostración: Sea M apareamiento donde no hay camino de aumento.

- Supongamos que existe M' apareamiento con $|M'| > |M|$.
- En G, coloreamos de rojo las aristas que están en M, de azúl las aristas que están en M' y de morado las de ambas.
- Borramos el resto de las aristas de G.
- Las aristas moradas no nos interesan: Simplemente son aristas sueltas no conectadas a nada.
- Tenemos una gráfica donde todos los vértices tienen grado 1 o 2.

Caminos de Aumento

Teorema

Sea M un apareamiento en G. Entonces M es máximo si y sólo si no existen caminos de aumento.

Demostración: Sea M apareamiento donde no hay camino de aumento.

- Supongamos que existe M' apareamiento con $|M'| > |M|$.
- En G, coloreamos de rojo las aristas que están en M, de azúl las aristas que están en M' y de morado las de ambas.
- Borramos el resto de las aristas de G.
- Las aristas moradas no nos interesan: Simplemente son aristas sueltas no conectadas a nada.
- Tenemos una gráfica donde todos los vértices tienen grado 1 o 2.
- Hay más aristas azules que rojas. Formamos así un camino de aumento.

Demostración del Teorema de Hall

Demostración:

- Debemos probar que, bajo la hipótesis de que $|S| \leq |N(S)|$ para todo S, existe un camino de aumento.

Demostración del Teorema de Hall

Demostración:

- Debemos probar que, bajo la hipótesis de que $|S| \leq |N(S)|$ para todo S, existe un camino de aumento.
- Consideremos un apareamiento maximal y supongamos que $x \in X$ no está apareado.

Demostración del Teorema de Hall

Demostración:

- Debemos probar que, bajo la hipótesis de que $|S| \leq |N(S)|$ para todo S, existe un camino de aumento.
- Consideremos un apareamiento maximal y supongamos que $x \in X$ no está apareado.
- Dividimos los vértices de X y de Y en 2: los apareados y los no apareados.

Demostración del Teorema de Hall

Demostración:

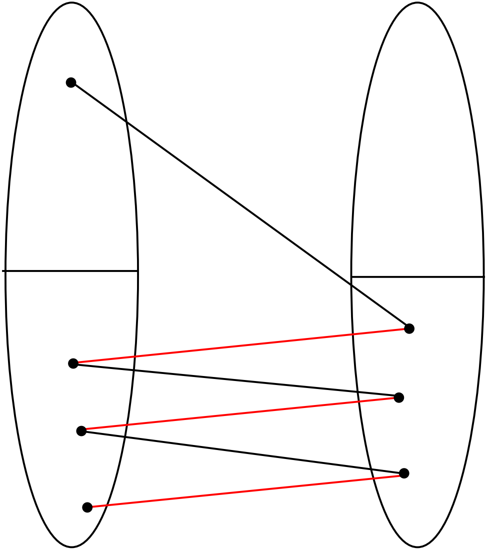
- Debemos probar que, bajo la hipótesis de que $|S| \leq |N(S)|$ para todo S, existe un camino de aumento.
- Consideremos un apareamiento maximal y supongamos que $x \in X$ no está apareado.
- Dividimos los vértices de X y de Y en 2: los apareados y los no apareados.
- Hay por lo menos una arista que sale de x. La consideramos. Si el nuevo vértice no está apareado, ya. Si sí, tomamos su pareja.

Demostración del Teorema de Hall

Demostración:

- Debemos probar que, bajo la hipótesis de que $|S| \leq |N(S)|$ para todo S, existe un camino de aumento.
- Consideremos un apareamiento maximal y supongamos que $x \in X$ no está apareado.
- Dividimos los vértices de X y de Y en 2: los apareados y los no apareados.
- Hay por lo menos una arista que sale de x. La consideramos. Si el nuevo vértice no está apareado, ya. Si sí, tomamos su pareja.
- Si en algún momento del camino podemos “ir” a la zona de Y que no tiene pareja, ya terminamos.

Demostración



Teorema de König-Egerváry

Definición

Una cubierta de vértices de una gráfica G es un subconjunto de los vértices tal que toda arista de G tiene al menos un vértice de la cubierta.

Teorema de König-Egerváry

Definición

Una cubierta de vértices de una gráfica G es un subconjunto de los vértices tal que toda arista de G tiene al menos un vértice de la cubierta.

- Observación: El complemento de una cubierta es un conjunto independiente y viceversa.

Teorema de König-Egerváry

Definición

Una cubierta de vértices de una gráfica G es un subconjunto de los vértices tal que toda arista de G tiene al menos un vértice de la cubierta.

- Observación: El complemento de una cubierta es un conjunto independiente y viceversa.
- Tenemos el siguiente teorema para gráficas bipartitas:

Teorema de König-Egerváry

Definición

Una cubierta de vértices de una gráfica G es un subconjunto de los vértices tal que toda arista de G tiene al menos un vértice de la cubierta.

- Observación: El complemento de una cubierta es un conjunto independiente y viceversa.
- Tenemos el siguiente teorema para gráficas bipartitas:

Teorema (König-Egerváry)

Dado una gráfica bipartita G, el número de aristas en un apareamiento máximo es igual al número de vértices de una cubierta mínima.

Demostración de König-Egerváry

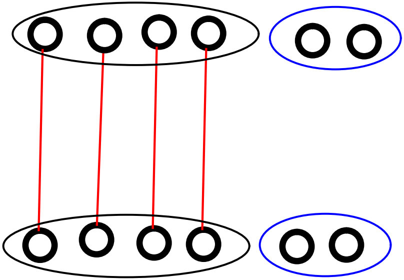
- Hay que ver qué es un apareamiento máximo y ver qué es una cubierta mínima.

Demostración de König-Egerváry

- Hay que ver qué es un apareamiento máximo y ver qué es una cubierta mínima.
- Consideremos un apareamiento máximo en G:

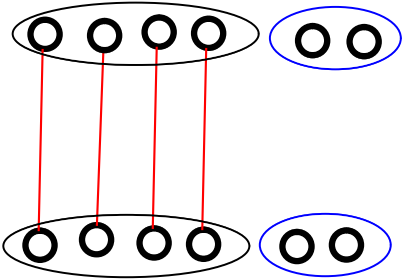
Demostración de König-Egerváry

- Hay que ver qué es un apareamiento máximo y ver qué es una cubierta mínima.
- Consideremos un apareamiento máximo en G:



Demostración de König-Egerváry

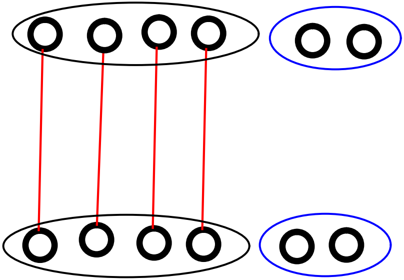
- Hay que ver qué es un apareamiento máximo y ver qué es una cubierta mínima.
- Consideremos un apareamiento máximo en G:



- Notemos que los "no apareados" no tienen aristas entre ellos.

Demostración de König-Egerváry

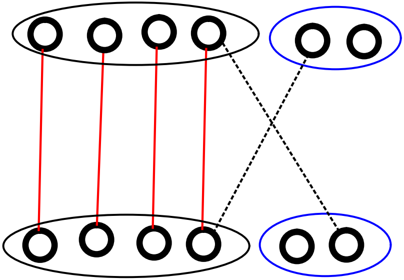
- Hay que ver qué es un apareamiento máximo y ver qué es una cubierta mínima.
- Consideremos un apareamiento máximo en G:



- Notemos que los "no apareados" no tienen aristas entre ellos.
- Para una tener una cubierta, debemos escoger al menos uno de los vértices de cada arista roja (y entonces $\#$apareamiento $\leq$ $\#$cubierta)

Demostración de König-Egerváry

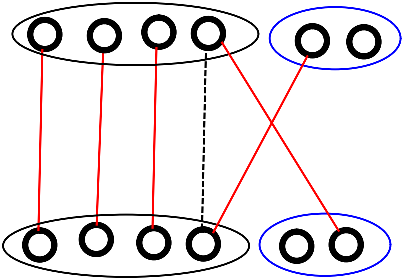
- Falta ver que podemos escoger de manera que obtenemos uno de cada arista roja de manera que obtenemos un apareamiento. Si tuviéramos una situación así:



- Si cada uno de los vértices de la arista tiene una conexión con nodos "no cubiertos", podemos intercambiar y obtener un apareamiento con más aristas.

Demostración de König-Egerváry

- Falta ver que podemos escoger de manera que obtenemos uno de cada arista roja de manera que obtenemos un apareamiento. Si tuviéramos una situación así:



- Si cada uno de los vértices de la arista tiene una conexión con nodos "no cubiertos", podemos intercambiar y obtener un apareamiento con más aristas.

Demostración de König-Egerváry

- Entonces podemos escoger de cada arista un vértice: aquél que tenga aristas hacia los "no apareados".

Demostración de König-Egerváry

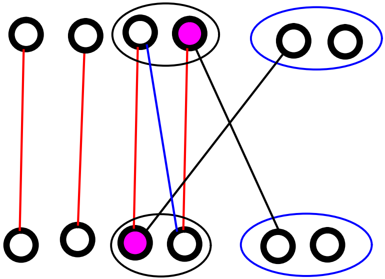
- Entonces podemos escoger de cada arista un vértice: aquél que tenga aristas hacia los "no apareados".
- Sólo escogemos, en el primer paso, los que SÍ tengan arista hacia los no apareados.

Demostración de König-Egerváry

- Entonces podemos escoger de cada arista un vértice: aquél que tenga aristas hacia los "no apareados".
- Sólo escogemos, en el primer paso, los que SÍ tengan arista hacia los no apareados.
- Repetimos: Si tuviéramos esta situación:

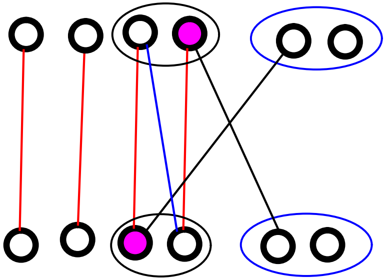
Demostración de König-Egerváry

- Entonces podemos escoger de cada arista un vértice: aquél que tenga aristas hacia los "no apareados".
- Sólo escogemos, en el primer paso, los que SÍ tengan arista hacia los no apareados.
- Repetimos: Si tuviéramos esta situación:



Demostración de König-Egerváry

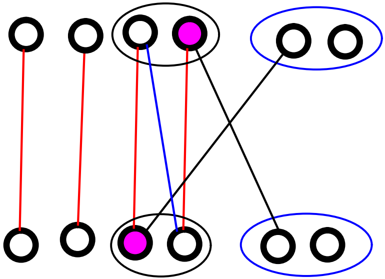
- Entonces podemos escoger de cada arista un vértice: aquél que tenga aristas hacia los "no apareados".
- Sólo escogemos, en el primer paso, los que SÍ tengan arista hacia los no apareados.
- Repetimos: Si tuviéramos esta situación:



- Tendríamos un camino de aumento.

Demostración de König-Egerváry

- Entonces podemos escoger de cada arista un vértice: aquél que tenga aristas hacia los “no apareados”.
- Sólo escogemos, en el primer paso, los que SÍ tengan arista hacia los no apareados.
- Repetimos: Si tuviéramos esta situación:



- Tendríamos un camino de aumento.
- Etc. Si no queda claro, hazlo en el pizarrón

Algoritmo en bipartitas

- ¿Cómo entonces encontramos un apareamiento máximo en una bipartita?

Algoritmo en bipartitas

- ¿Cómo entonces encontramos un apareamiento máximo en una bipartita?
- No es dificíl:

Algoritmo en bipartitas

- ¿Cómo entonces encontramos un apareamiento máximo en una bipartita?
- No es dificíl:
- Comienza con cualquier apareamiento.

Algoritmo en bipartitas

- ¿Cómo entonces encontramos un apareamiento máximo en una bipartita?
- No es dificíl:
- Comienza con cualquier apareamiento.
- Encuentra caminos de aumento.

Algoritmo en bipartitas

- ¿Cómo entonces encontramos un apareamiento máximo en una bipartita?
- No es dificíl:
- Comienza con cualquier apareamiento.
- Encuentra caminos de aumento.
- Úsalos par extender el apareamiento.

Algoritmo en bipartitas

- ¿Cómo entonces encontramos un apareamiento máximo en una bipartita?
- No es dificíl:
- Comienza con cualquier apareamiento.
- Encuentra caminos de aumento.
- Úsalos par extender el apareamiento.
- Posible tema de exposición: Hopcroft-Karp tienen un algoritmo que encuentra varios caminos de aumento a la vez. En la teoría es más rápido, pero en la práctica parece que no.

Algoritmo en bipartitas

- ¿Cómo encuentro caminos de aumento?

Algoritmo en bipartitas

- ¿Cómo encuentro caminos de aumento?
- DFS o BFS. En la práctica parece que BFS es mejor.

Algoritmo en bipartitas

- ¿Cómo encuentro caminos de aumento?
- DFS o BFS. En la práctica parece que BFS es mejor.
- Vamos a escribir pseudo-código.

Índice:

Algoritmo

- Suponiendo que tenemos una función que encuentra caminos de aumento, aquí está el algoritmo para encontrar un apareamiento máximo:

Algoritmo

- Suponiendo que tenemos una función que encuentra caminos de aumento, aquí está el algoritmo para encontrar un apareamiento máximo:
  - Escoge al azar cualquier apareamiento.

Algoritmo

- Suponiendo que tenemos una función que encuentra caminos de aumento, aquí está el algoritmo para encontrar un apareamiento máximo:
  - Escoge al azar cualquier apareamiento.
  - Encuentra un camino de aumento, si es que existe.

Algoritmo

- Suponiendo que tenemos una función que encuentra caminos de aumento, aquí está el algoritmo para encontrar un apareamiento máximo:
  - Escoge al azar cualquier apareamiento.
  - Encuentra un camino de aumento, si es que existe.
  - Si no existe, terminamos.

Algoritmo

- Suponiendo que tenemos una función que encuentra caminos de aumento, aquí está el algoritmo para encontrar un apareamiento máximo:
  - Escoge al azar cualquier apareamiento.
  - Encuentra un camino de aumento, si es que existe.
  - Si no existe, terminamos.
  - Si existe, intercambia las aristas impares del camino por las pares y encuentra apareamiento con más aristas.

Algoritmo

- Suponiendo que tenemos una función que encuentra caminos de aumento, aquí está el algoritmo para encontrar un apareamiento máximo:
  - Escoge al azar cualquier apareamiento.
  - Encuentra un camino de aumento, si es que existe.
  - Si no existe, terminamos.
  - Si existe, intercambia las aristas impares del camino por las pares y encuentra apareamiento con más aristas.
- Entonces sólo queda encontrar un algoritmo que encuentre caminos de aumento: El algoritmo de las flores.

Algoritmo

- Suponiendo que tenemos una función que encuentra caminos de aumento, aquí está el algoritmo para encontrar un apareamiento máximo:
  - Escoge al azar cualquier apareamiento.
  - Encuentra un camino de aumento, si es que existe.
  - Si no existe, terminamos.
  - Si existe, intercambia las aristas impares del camino por las pares y encuentra apareamiento con más aristas.
- Entonces sólo queda encontrar un algoritmo que encuentre caminos de aumento: El algoritmo de las flores.
- Podemos suponer que ningún vértice tiene grado 1 o 0 (así nunca se nos terminará un camino alternante en una arista roja).

Qué va a pasar

- Intentemos lo obvio:

Qué va a pasar

- Intentemos lo obvio:
- Consideremos un vértice que no tiene pareja y una arista (negra) que sale de él. Si el núevo vértice no tiene pareja tampoco, terminaste.

Qué va a pasar

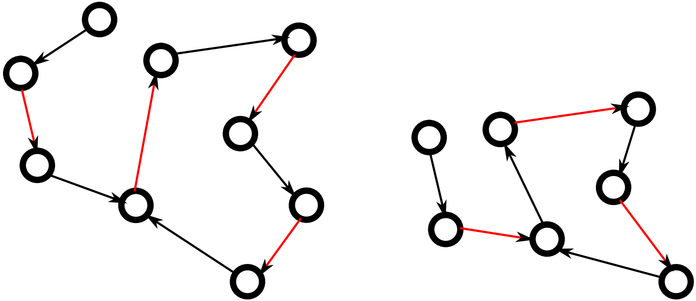
- Intentemos lo obvio:
- Consideremos un vértice que no tiene pareja y una arista (negra) que sale de él. Si el núevo vértice no tiene pareja tampoco, terminaste.
- Si sí tiene pareja, camina por esa arista roja, luego por la arista negra, etc.

Qué va a pasar

- Intentemos lo obvio:
- Consideremos un vértice que no tiene pareja y una arista (negra) que sale de él. Si el núevo vértice no tiene pareja tampoco, terminaste.
- Si sí tiene pareja, camina por esa arista roja, luego por la arista negra, etc.
- El problema es que puedo encontrar ciclos así:

Qué va a pasar

- Intentemos lo obvio:
- Consideremos un vértice que no tiene pareja y una arista (negra) que sale de él. Si el núevo vértice no tiene pareja tampoco, terminaste.
- Si sí tiene pareja, camina por esa arista roja, luego por la arista negra, etc.
- El problema es que puedo encontrar ciclos así:

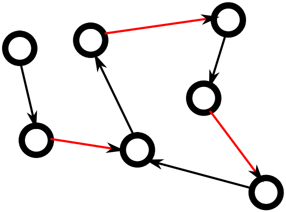


Ciclos impares

- Veamos el segundo caso, cuando el ciclo es impar:

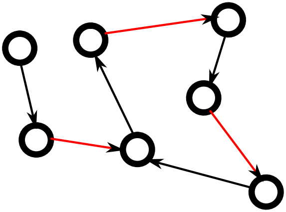
Ciclos impares

- Veamos el segundo caso, cuando el ciclo es impar:



Ciclos impares

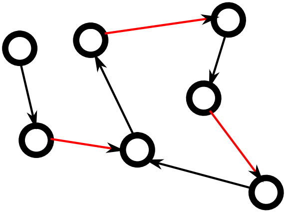
- Veamos el segundo caso, cuando el ciclo es impar:



- Notemos que podríamos haber recorrido el ciclo en dirección opuesta.

Ciclos impares

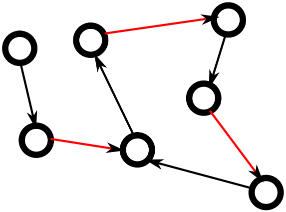
- Veamos el segundo caso, cuando el ciclo es impar:



- Notemos que podríamos haber recorrido el ciclo en dirección opuesta.
- A cada vértice del ciclo, podemos llegarle con un camino alternante cuyo última arista sea del color que queramos.

Ciclos impares

- Veamos el segundo caso, cuando el ciclo es impar:



- Notemos que podríamos haber recorrido el ciclo en dirección opuesta.
- A cada vértice del ciclo, podemos llegarle con un camino alternante cuyo última arista sea del color que queramos.
- Es decir, podemos contraer ese ciclo impar y no cambiar nada.

Flores

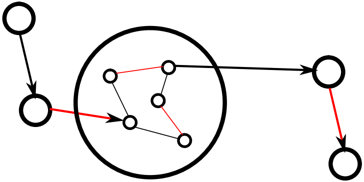
- Decimos que un ciclo impar de tamaño $2k+1$ en la que exactamente k aristas son rojas es una flor.

Flores

- Decimos que un ciclo impar de tamaño $2k+1$ en la que exactamente k aristas son rojas es una flor.
- Al contraer una flor a un punto, si tuviéramos un camino de aumento que use la flor como vértice, podemos completarlo a un camino de aumento de la gráfica original:

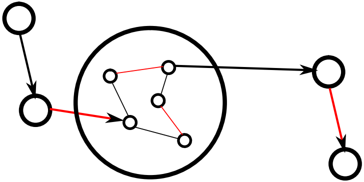
Flores

- Decimos que un ciclo impar de tamaño $2k+1$ en la que exactamente k aristas son rojas es una flor.
- Al contraer una flor a un punto, si tuviéramos un camino de aumento que use la flor como vértice, podemos completarlo a un camino de aumento de la gráfica original:



Flores

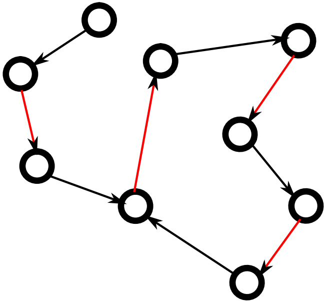
- Decimos que un ciclo impar de tamaño $2k+1$ en la que exactamente k aristas son rojas es una flor.
- Al contraer una flor a un punto, si tuviéramos un camino de aumento que use la flor como vértice, podemos completarlo a un camino de aumento de la gráfica original:



- Notemos que siempre la arista roja del camino de aumento debe llegar al vértice con dos aristas negras de la flor.

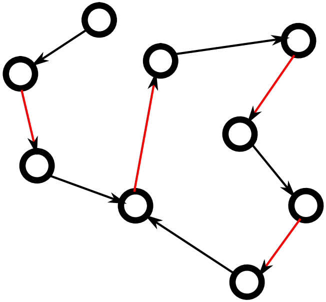
Ciclos Pares

- Ahora consideremos el primer caso, cuando el ciclo es par:



Ciclos Pares

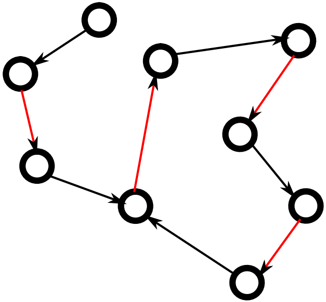
- Ahora consideremos el primer caso, cuando el ciclo es par:



- Aquí no podemos hacer nada y tendremos que seguir buscando por otro lado.

Ciclos Pares

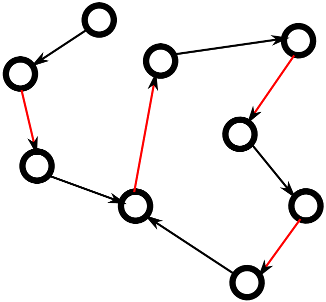
- Ahora consideremos el primer caso, cuando el ciclo es par:



- Aquí no podemos hacer nada y tendremos que seguir buscando por otro lado.
- Vamos a formar un bosque de búsqueda.

Ciclos Pares

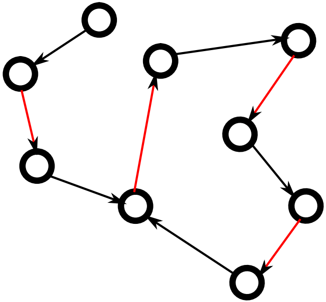
- Ahora consideremos el primer caso, cuando el ciclo es par:



- Aquí no podemos hacer nada y tendremos que seguir buscando por otro lado.
- Vamos a formar un bosque de búsqueda.
- Consideremos el conjunto de todos los nodos no apareados. A estos les llamamos nodos expuestos.

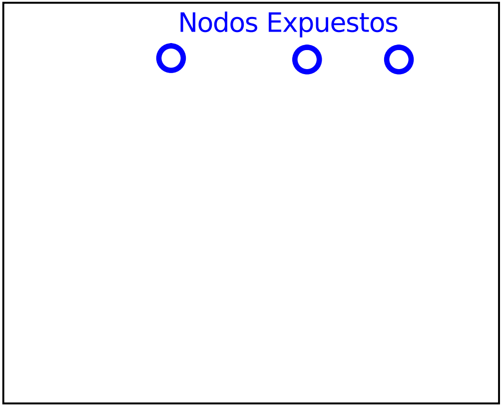
Ciclos Pares

- Ahora consideremos el primer caso, cuando el ciclo es par:



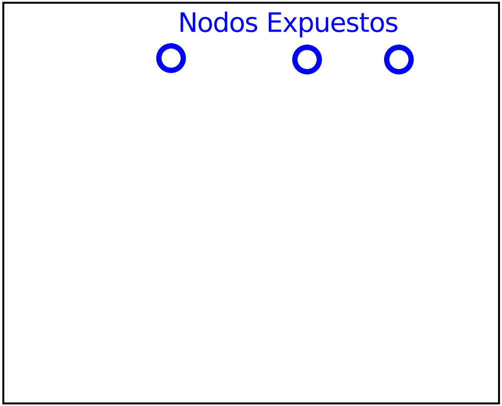
- Aquí no podemos hacer nada y tendremos que seguir buscando por otro lado.
- Vamos a formar un bosque de búsqueda.
- Consideremos el conjunto de todos los nodos no apareados. A estos les llamamos nodos expuestos.
- En cada nodo expuesto empezaremos a formar un árbol (por eso crearemos un bosque).

Bosque



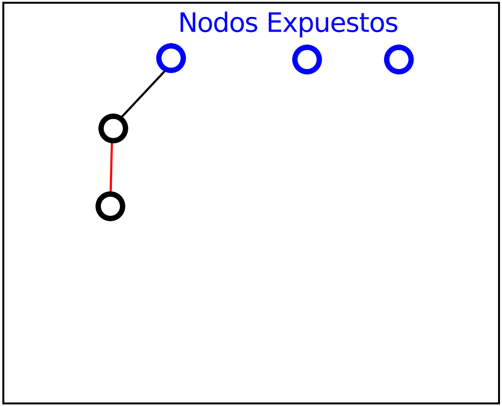
- Empezamos con los nodos expuestos

Bosque



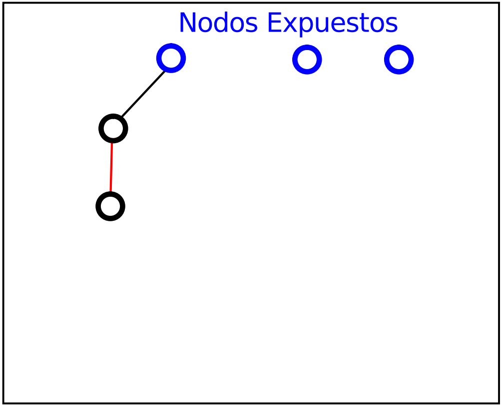
- Empezamos con los nodos expuestos
- Iremos agregando todos los posibles vecinos (negros)

Bosque



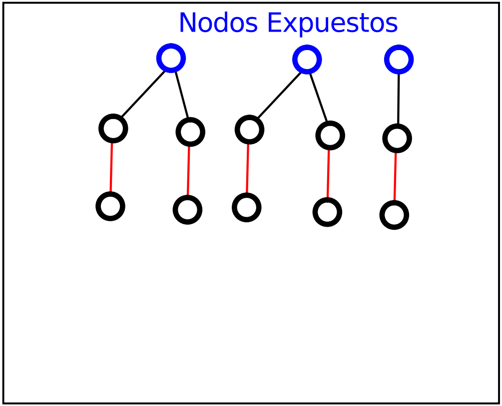
- Cada que agregamos a un vecino (negro) agregamos inmediatamente a su pareja (roja).

Bosque



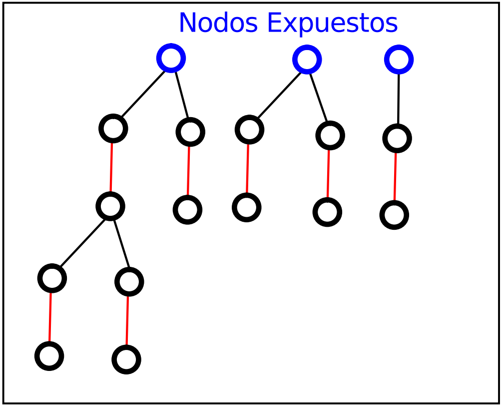
- Cada que agregamos a un vecino (negro) agregamos inmediatamente a su pareja (roja).
- Si no tiene pareja roja, terminamos, ahí hay un camino de aumento.

Bosque



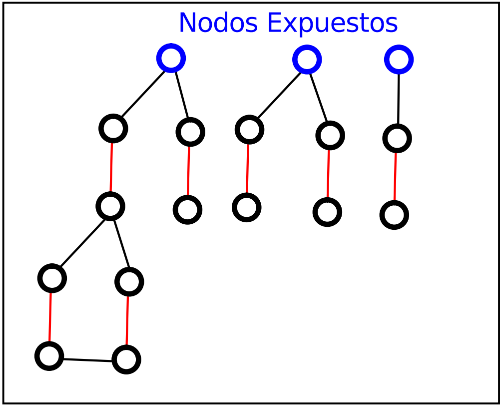
- Seguimos construyendo el árbol.

Bosque



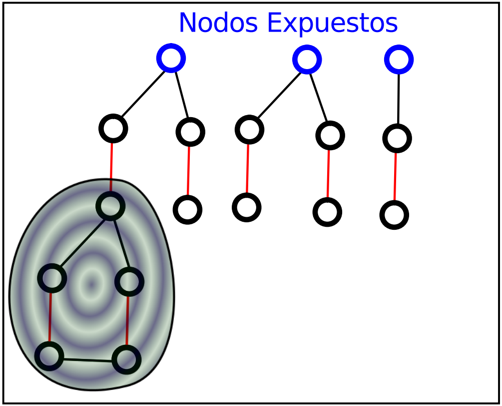
- Seguimos construyendo el árbol.

Bosque



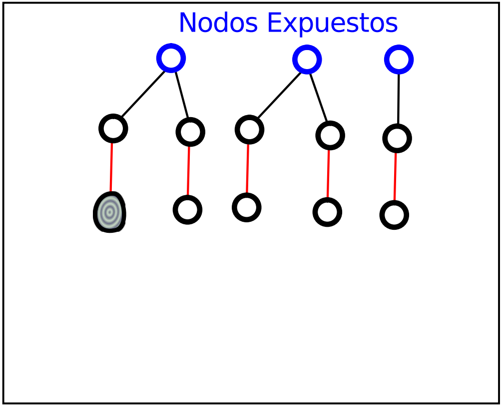
- ¿Qué ocurre si encontramos una arista en el mismo bosque?

Bosque



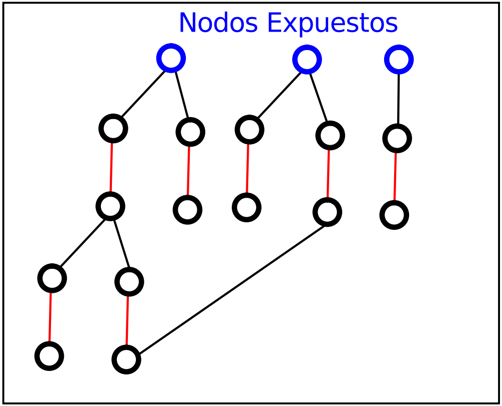
- ¡Podemos formar una flor!

Bosque



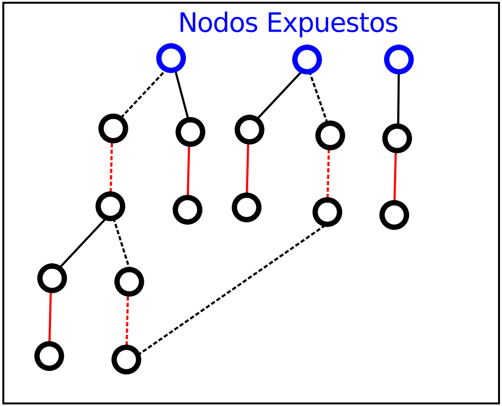
- Comprimimos la flor y continuamos.

Bosque



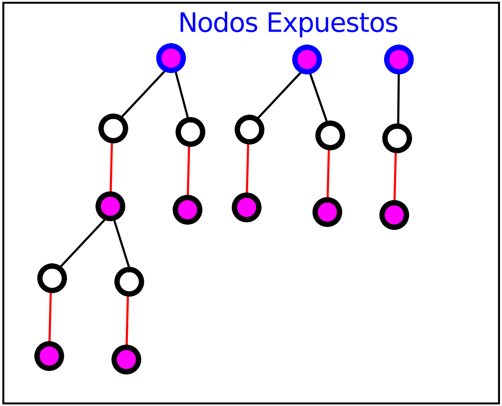
- Qué pasa si encontramos una arista entre diferentes bosques?

Bosque



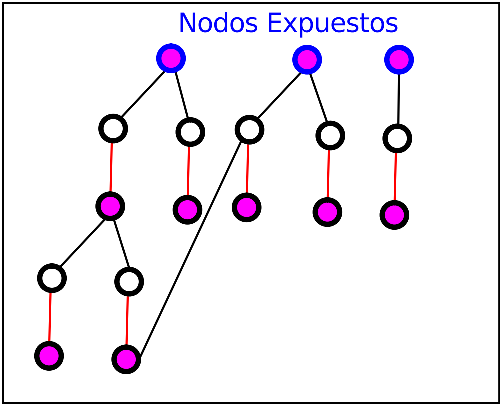
- Entonces tenemos un camino de aumento!

Bosque



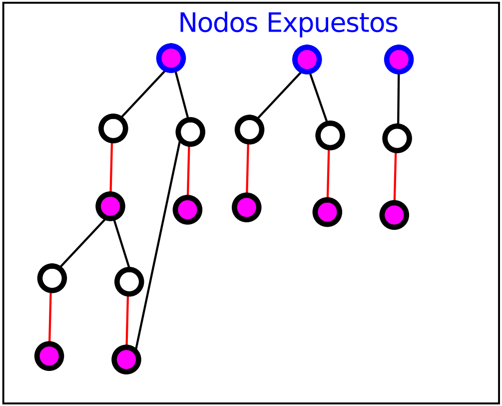
- En realidad sólo nos fijaremos en los nodos "pares".
- Los nodos impares los expandiremos inmediatamente, así que no importan mucho.

Bosque



- Si tuviéramos una arista (negra debe ser) a un nodo impar de otro bosque o de otra rama, para el caso de encontrar un camino de aumento, no afecta y podemos ignorarla.

Bosque



- Si tuviéramos una arista (negra debe ser) a un nodo impar de otro bosque o de otra rama, para el caso de encontrar un camino de aumento, no afecta y podemos ignorarla.

Implementación

- Bosque $= \{v : v$ expuesto$\}$

Implementación

- Bosque $= \{v : v \text{ expuesto}\}$
- Mientras $\exists v$ "vértice final" en Bosque con aristas negras:

Implementación

- Bosque $= \{v : v \text{ expuesto}\}$
- Mientras $\exists v$ "vértice final" en Bosque con aristas negras:
  - Para cada w vecino de v con aristas negra, haz:

Implementación

- Bosque $= \{v : v \text{ expuesto}\}$
- Mientras $\exists v$ “vértice final” en Bosque con aristas negras:
  - Para cada w vecino de v con aristas negra, haz:
  - Si w no está en Bosque, agrega $v \to w$ y luego si w' pareja de w, agrega $w \to w'$.

Implementación

- Bosque $= \{v : v \text{ expuesto}\}$
- Mientras $\exists v$ "vértice final" en Bosque con aristas negras:
  - Para cada w vecino de v con aristas negra, haz:
  - Si w no está en Bosque, agrega $v \to w$ y luego si w' pareja de w, agrega $w \to w'$.
  - Si w ya estaba en Bosque, hay dos casos:

Implementación

- Bosque $= \{v : v \text{ expuesto}\}$
- Mientras $\exists v$ "vértice final" en Bosque con aristas negras:
  - Para cada w vecino de v con aristas negra, haz:
  - Si w no está en Bosque, agrega $v \to w$ y luego si w' pareja de w, agrega $w \to w'$.
  - Si w ya estaba en Bosque, hay dos casos:
    - Si w era nodo "impar", ignóralo, no lo agregues a nada.

Implementación

- Bosque $= \{v : v \text{ expuesto}\}$
- Mientras $\exists v$ "vértice final" en Bosque con aristas negras:
  - Para cada w vecino de v con aristas negra, haz:
  - Si w no está en Bosque, agrega $v \to w$ y luego si w' pareja de w, agrega $w \to w'$.
  - Si w ya estaba en Bosque, hay dos casos:
    - Si w era nodo "impar", ignóralo, no lo agregues a nada.
    - Si w era nodo "par" y NO está en el mismo bosque, encontraste un camino de aumento! Felicidades!

Implementación

- Bosque $= \{v : v \text{ expuesto}\}$
- Mientras $\exists v$ "vértice final" en Bosque con aristas negras:
  - Para cada w vecino de v con aristas negra, haz:
  - Si w no está en Bosque, agrega $v \to w$ y luego si w' pareja de w, agrega $w \to w'$.
  - Si w ya estaba en Bosque, hay dos casos:
    - Si w era nodo "impar", ignóralo, no lo agregues a nada.
    - Si w era nodo "par" y NO está en el mismo bosque, encontraste un camino de aumento! Felicidades!
    - Si w era nodo "par", y está en el mismo bosque, entonces encontraste una flor. Comprímela y corre de nuevo el programa en el comprimido.

Conclusiones

- El algoritmo corre en el peor caso en tiempo de $O(V^4)$. Sin embargo, usualmente es más rápido, en especial si comienzas con un apareamiento grande (avaricioso).

Conclusiones

- El algoritmo corre en el peor caso en tiempo de $O(V^4)$. Sin embargo, usualmente es más rápido, en especial si comienzas con un apareamiento grande (avaricioso).
- Podemos ir “marcando” los nodos ya explorados en un arreglo (para ver si llega arista luego a ellos), y podemos ir marcando si ya lo marcamos como nodo impar, como nodo par, o ambos.

Conclusiones

- El algoritmo corre en el peor caso en tiempo de $O(V^4)$. Sin embargo, usualmente es más rápido, en especial si comienzas con un apareamiento grande (avaricioso).
- Podemos ir "marcando" los nodos ya explorados en un arreglo (para ver si llega arista luego a ellos), y podemos ir marcando si ya lo marcamos como nodo impar, como nodo par, o ambos.
- Debemos tener una función "descomprimir camino" para cuando comprimimos una flor y encontramos un camino ahí adentro.

Conclusiones

- El algoritmo corre en el peor caso en tiempo de $O(V^4)$. Sin embargo, usualmente es más rápido, en especial si comienzas con un apareamiento grande (avaricioso).
- Podemos ir "marcando" los nodos ya explorados en un arreglo (para ver si llega arista luego a ellos), y podemos ir marcando si ya lo marcamos como nodo impar, como nodo par, o ambos.
- Debemos tener una función "descomprimir camino" para cuando comprimimos una flor y encontramos un camino ahí adentro.
- Hay pequeñas mejoras de este algoritmo, pero se basan en la misma idea de flores. De hecho, se llaman: Blossom II, Blossom III, Blossom IV y Blossom V.

Conclusiones

- El algoritmo corre en el peor caso en tiempo de $O(V^4)$. Sin embargo, usualmente es más rápido, en especial si comienzas con un apareamiento grande (avaricioso).
- Podemos ir "marcando" los nodos ya explorados en un arreglo (para ver si llega arista luego a ellos), y podemos ir marcando si ya lo marcamos como nodo impar, como nodo par, o ambos.
- Debemos tener una función "descomprimir camino" para cuando comprimimos una flor y encontramos un camino ahí adentro.
- Hay pequeñas mejoras de este algoritmo, pero se basan en la misma idea de flores. De hecho, se llaman: Blossom II, Blossom III, Blossom IV y Blossom V.
- Varios de estos algoritmos funcionan para gráficas con pesos!