Test de Weisfeiler-Lehman

Alexander Pinto - Marcelo Arenas

IIC3810

Un poco de historia

- Test de isomorfismo de grafos basado en etiquetado canónico.
- Weisfeiler & Lehman (**WL**),1968, algoritmo de etiquetado de vértices (**1-WL, 2-WL**).
- Babai & Mathon, 1979, algoritmo de etiquetado de tuplas de vértices ($\boldsymbol{k}$**-WL**).
- Cai, Furer & Immerman, 1992, variante del algoritmo ($\boldsymbol{k}$**-folklore-WL**).

Equivalencia por color

El algoritmo define una relación de equivalencia en el color (etiqueta) de cada nodo de un grafo.

- C es una función que devuelve el color de un nodo.
- $\sim_C$ es una relación de equivalencia tal que $u \sim_C v$ si y sólo si $C(u) = C(v)$.

Refinamiento

Dadas: relaciones de equivalencia R y S

- S es un refinamiento de R si toda clase de equivalencia de S es un subconjunto de una clase de equivalencia de R.
- Además, S es un refinamiento estricto de R si al menos una clase de equivalencia de S es un subconjunto propio de una clase de equivalencia de R.

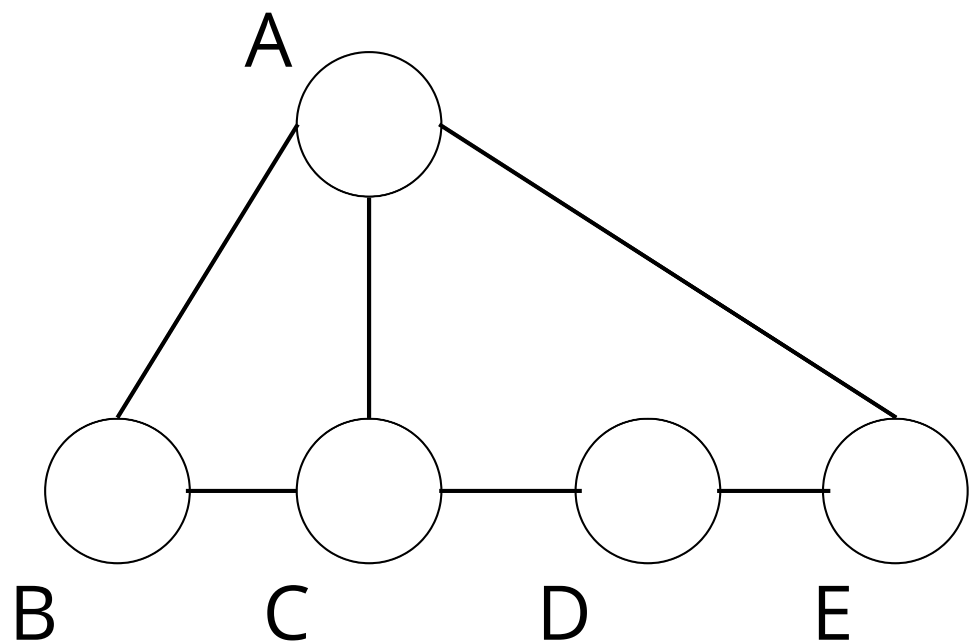
Algoritmo 1-WL

Input: $G = (V, E)$

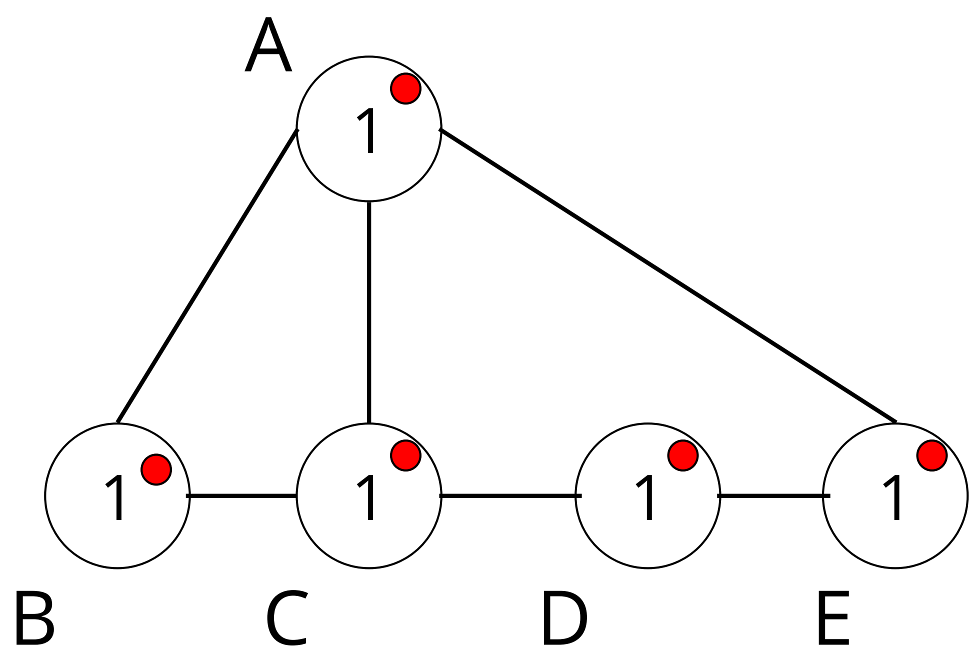
1. **for each** $v \in V$ **do**
2. $\quad C[v] \leftarrow 1$
3. **repeat**
4. $\quad C^{old} \leftarrow C$
5. $\quad$ **for each** $v \in V$ **do**
6. $\qquad C[v] \leftarrow \mathrm{hash}\Big(C^{old}[v], \{\!\!\{C^{old}[w] \mid w \in N_G(v)\}\!\!\}\Big)$
7. **until not** $\sim_C$ es un refinamiento extricto de $\sim_{C^{old}}$
8. **return** C

$N_G(v) = \{u \in V \mid \{u, v\} \in E\}$

Algoritmo 1-WL

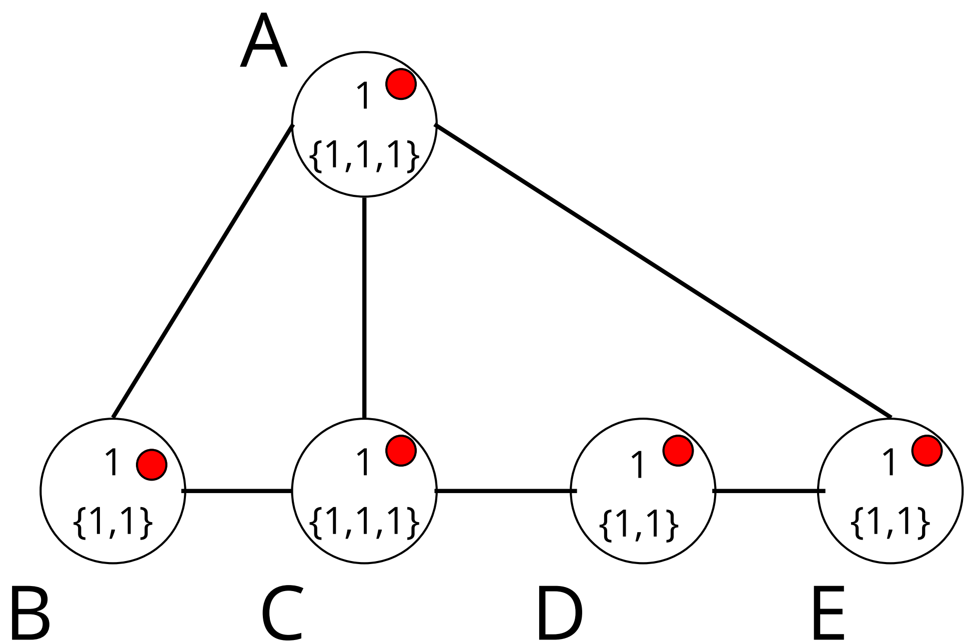


1. **for each** $v \in V$ **do**
2. $C[v] \leftarrow 1$



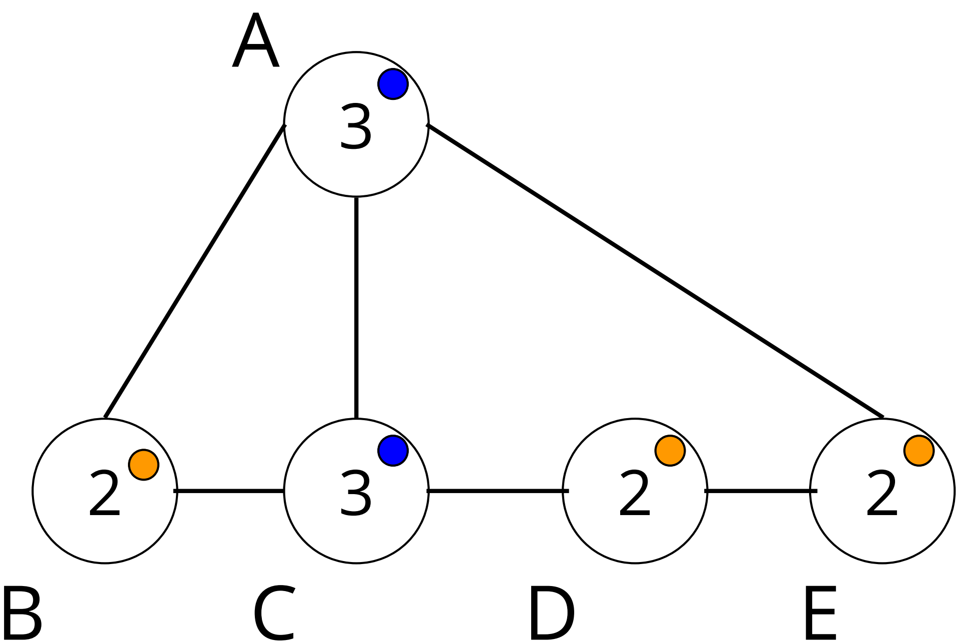
1 : C^{old}

$\{\!\{1, 1, 1\}\!\}$: $\{\!\{C^{old}[w] \mid w \in N_G(v)\}\!\}$



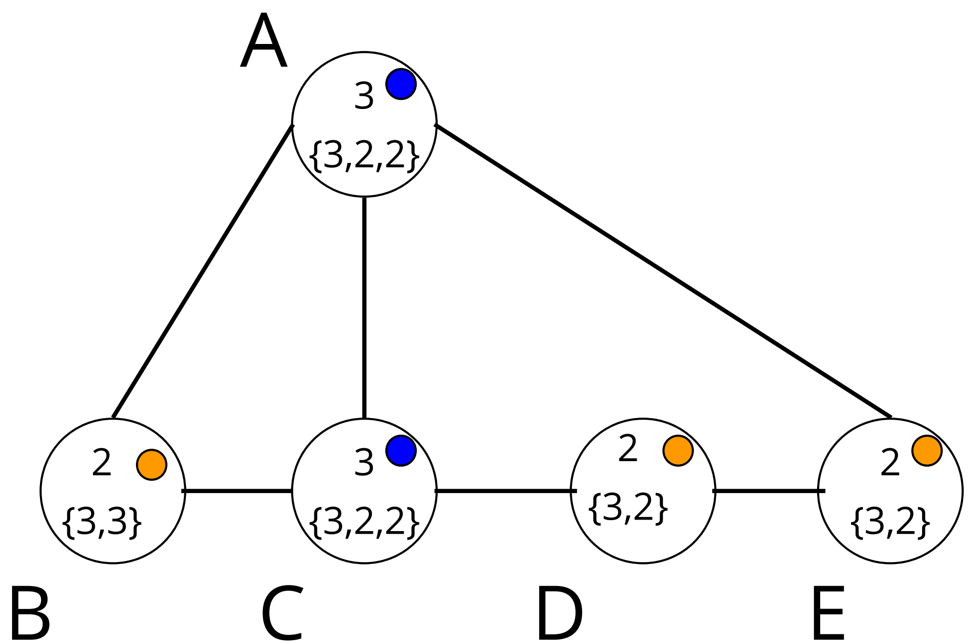
$$C[v] \leftarrow \mathrm{hash}(C^{old}[v], \{\!\{C^{old}[w] \mid w \in N_G(v)\}\!\})$$

$$3 \longleftarrow \begin{array}{ccc} 1 & : & C^{old} \\ \{\!\{1,1,1\}\!\} & : & \{\!\{C^{old}[w] \mid w \in N_G(v)\}\!\} \end{array}$$



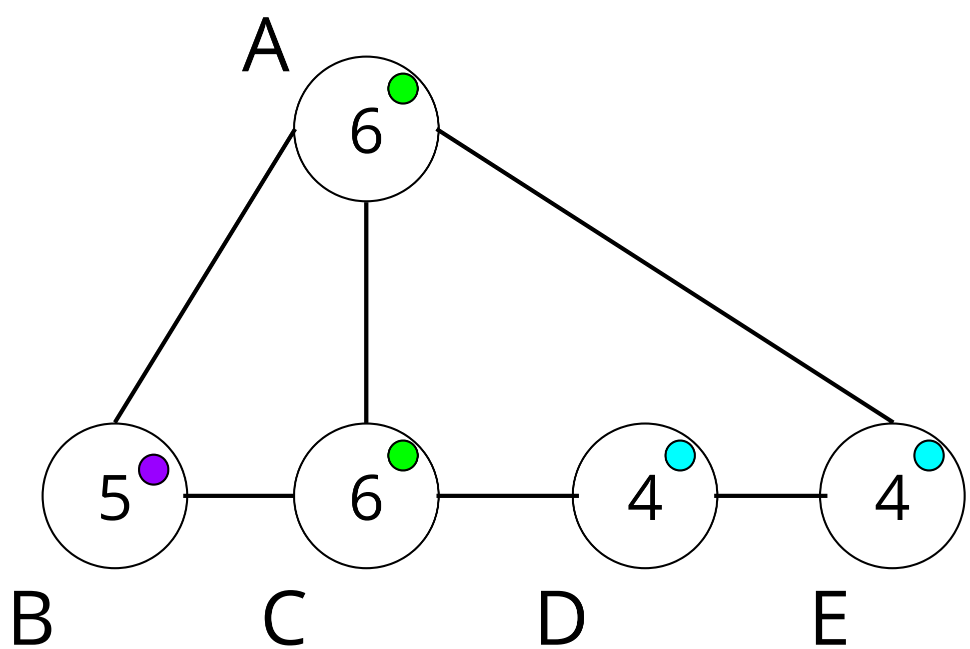
3 : C^{old}

$\{\!\{3, 2, 2\}\!\}$: $\{\!\{C^{old}[w] \mid w \in N_G(v)\}\!\}$



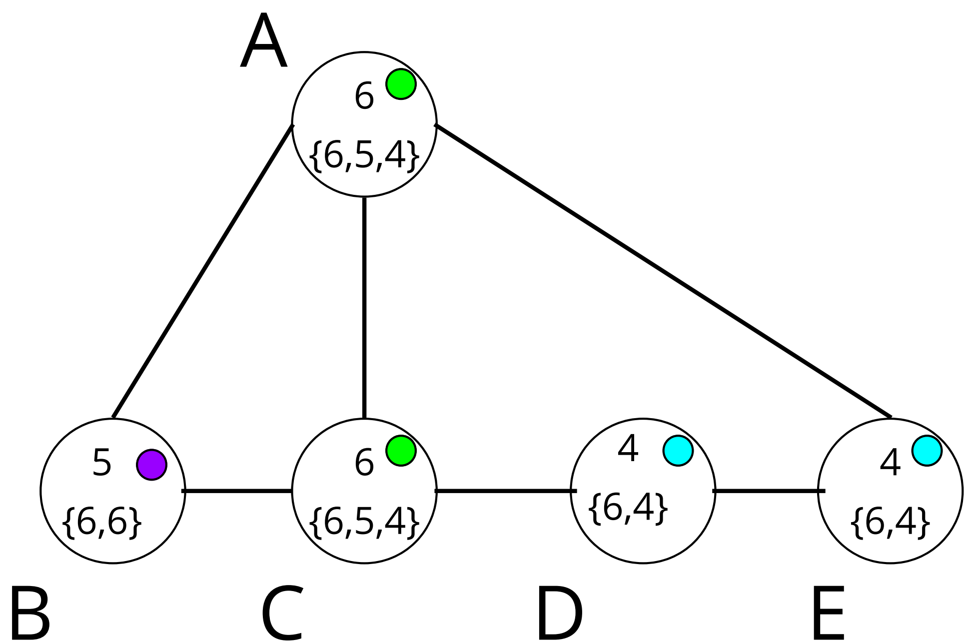
$$C[v] \leftarrow \mathrm{hash}(C^{old}[v], \{\!\!\{C^{old}[w] \mid w \in N_G(v)\}\!\!\})$$

$$6 \longleftarrow \begin{array}{ccl} 3 & : & C^{old} \\ \{\!\!\{3, 2, 2\}\!\!\} & : & \{\!\!\{C^{old}[w] \mid w \in N_G(v)\}\!\!\} \end{array}$$

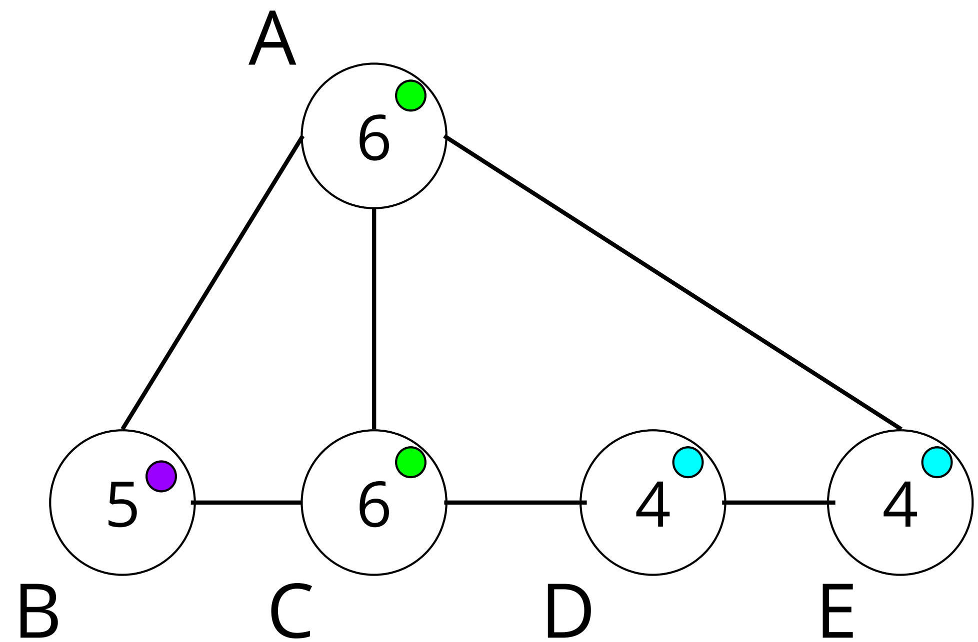


$$6 \quad : \quad C^{old}$$

$$\{\!\![6, 5, 4]\!\!\} \quad : \quad \{\!\![C^{old}[w] \mid w \in N_G(v)]\!\!\}$$



$$C[v] \leftarrow \text{hash}(C^{old}[v], \{\!\!\{C^{old}[w] \mid w \in N_G(v)\}\!\!\})$$



8. **until not** $\sim_C$ es un refinamiento extricto de $\sim_{C^{old}}$

9. **return** C

$C =$ { A $\rightarrow$ 6, B $\rightarrow$ 5, C $\rightarrow$ 6, D $\rightarrow$ 4, E $\rightarrow$ 4 }

Test de isomorfismo WL

Input: $G = (V, E)$ y $G' = (V', E')$

1. **if** $|V| \neq |V'|$ **or** $|E| \neq |E'|$ **then**

   return G no es isomorfo a G'

2. $H \leftarrow G \uplus G'$

3. $C = 1\text{-WL}(H)$

4. Ordenar $C(G)$ y $C(G')$ según color

5. **for** $i = 1$ **to** $|V|$ **do**

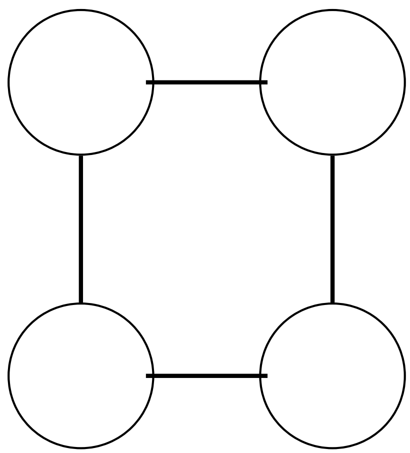
6. **if** $C(G)[i] \neq C(G')[i]$ **then**

7. **return** G no es isomorfo a G'

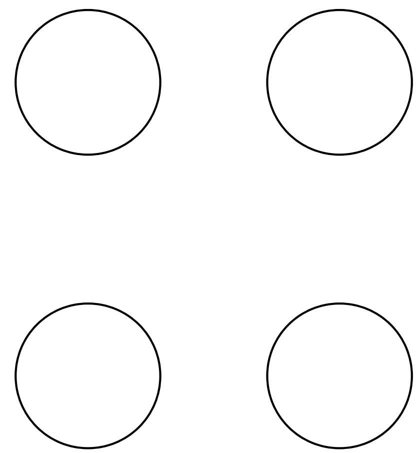
8. **return** G es *isomorfo* a G'

¿Por qué colocamos a G y G' en el mismo grafo H?

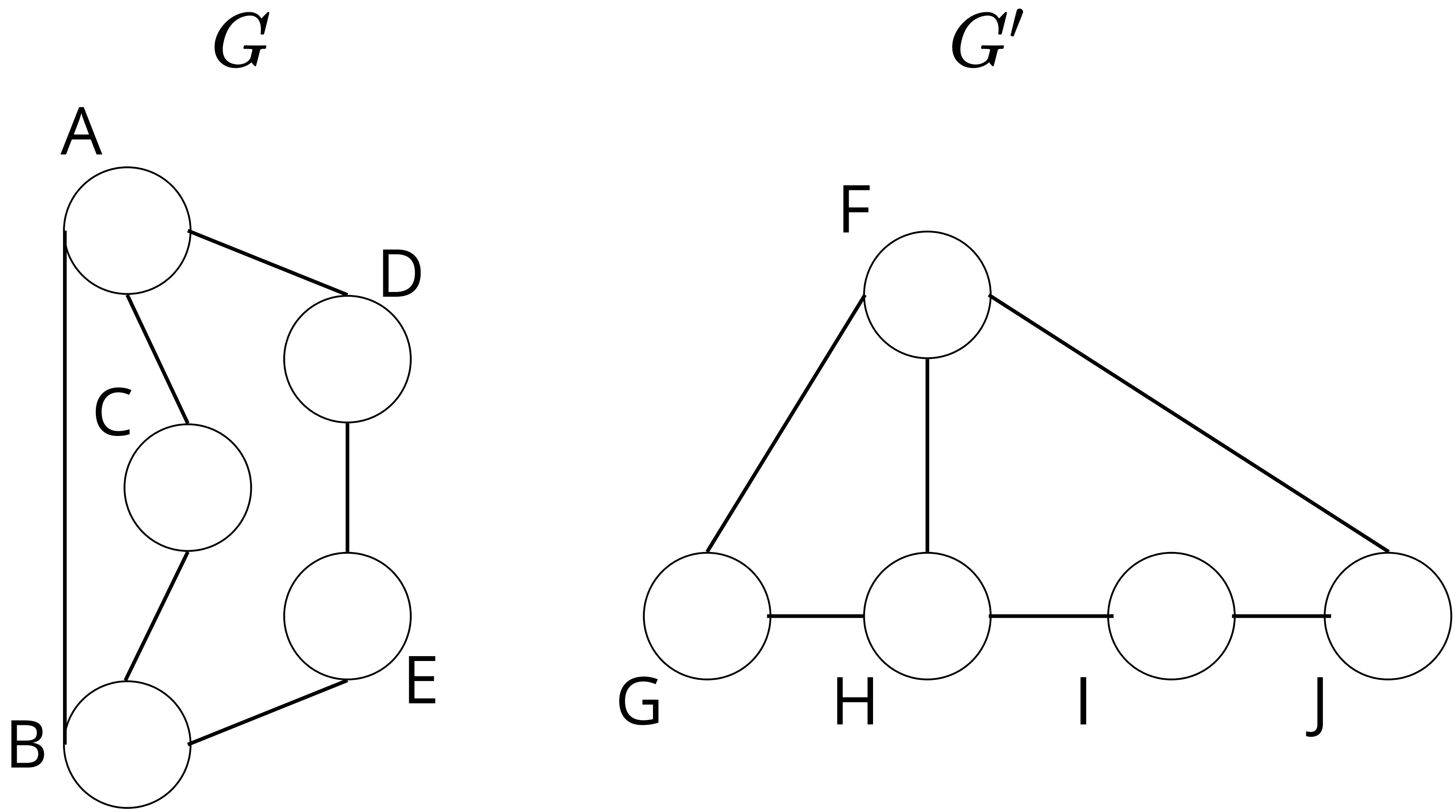
G



G'

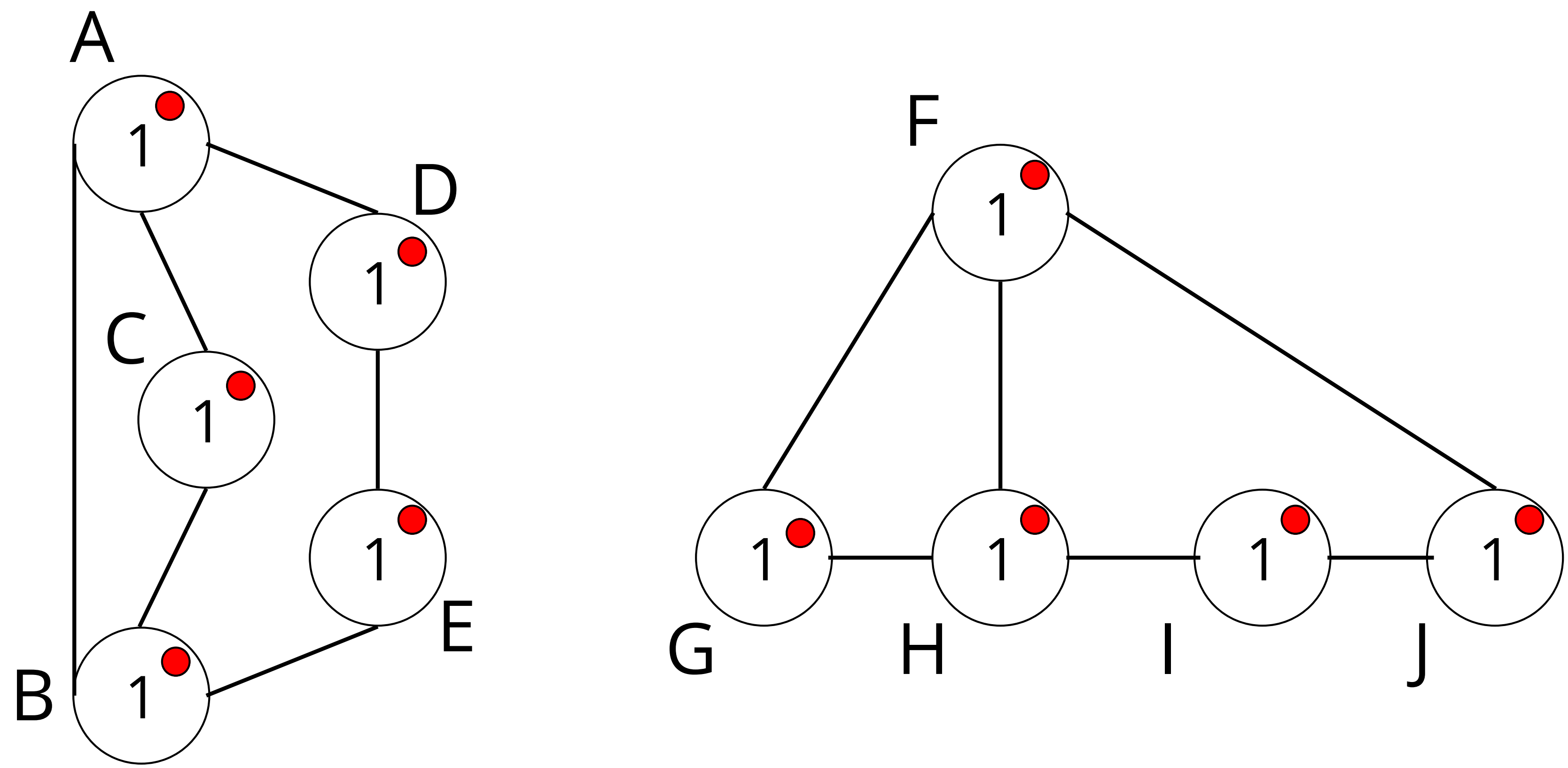


Test de isomorfismo WL



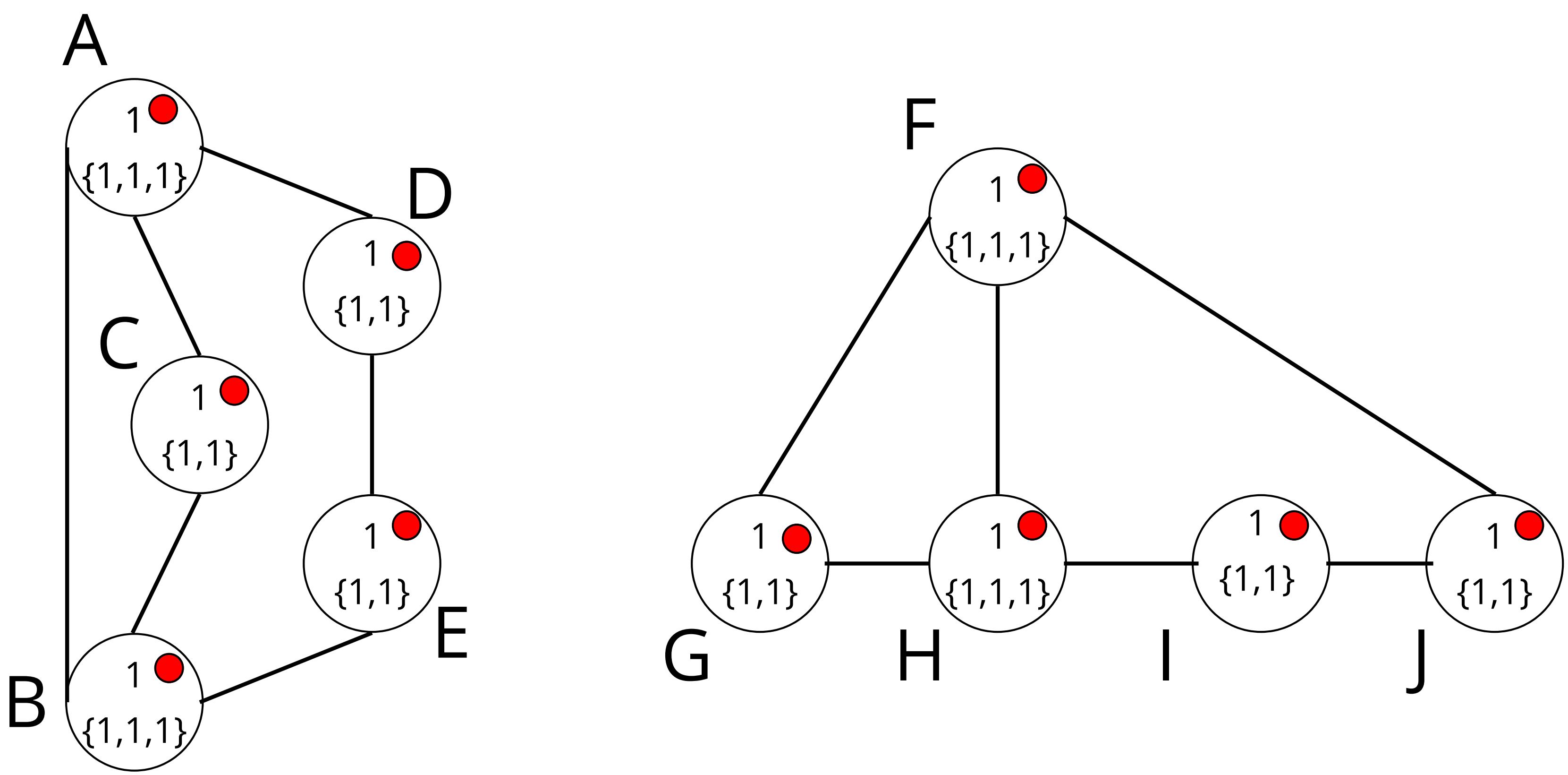
Algoritmo 1-WL

1. **for each** $v \in V$ **do**
2. $\quad C[v] \leftarrow 1$



$$1 \quad : \quad C^{old}$$

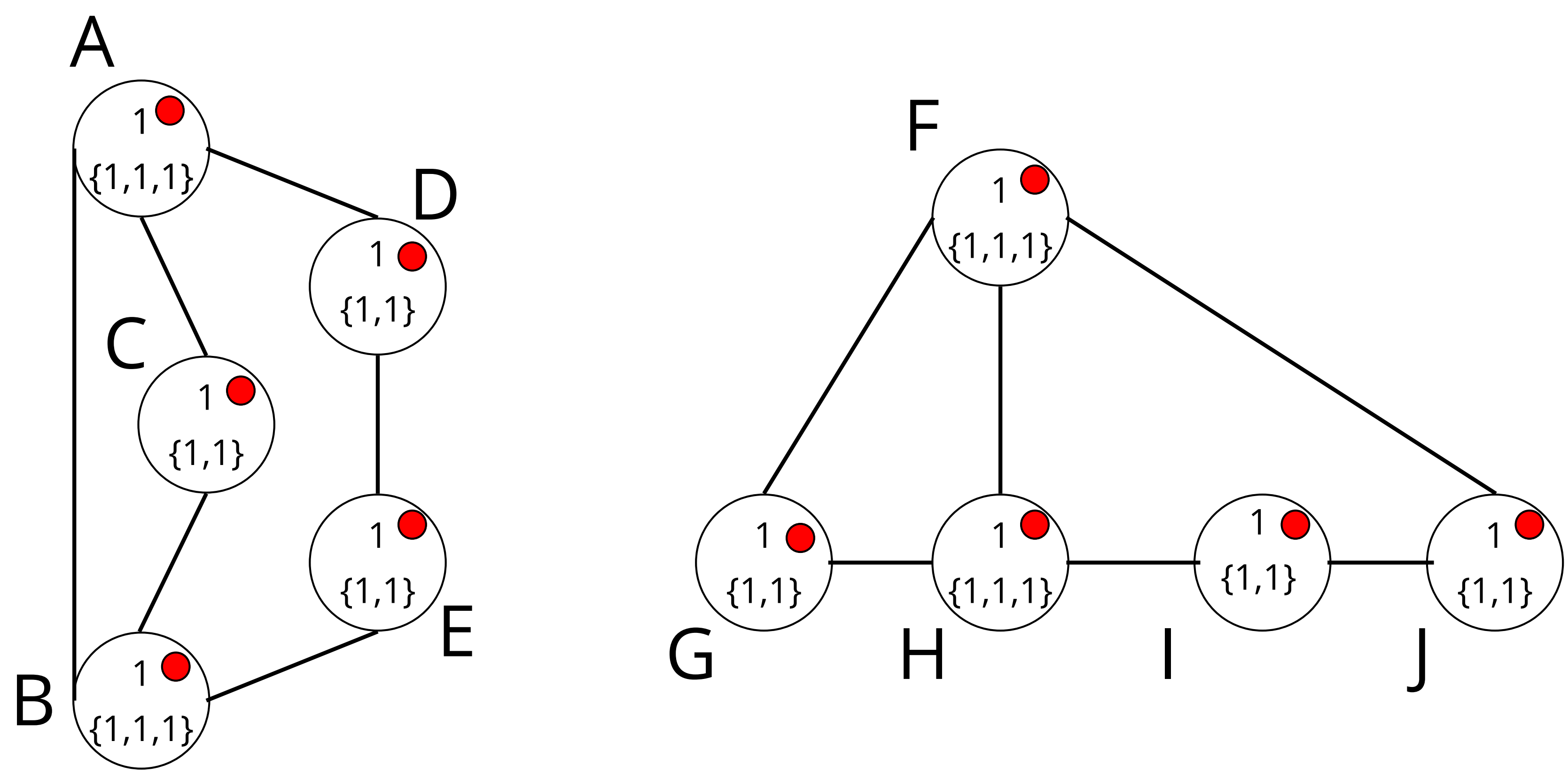
$$\{\!\{1, 1, 1\}\!\} \quad : \quad \{\!\{C^{old}[w] \mid w \in N_H(v)\}\!\}$$



$$C[v] \leftarrow \mathrm{hash}(C^{old}[v], \{\!\{C^{old}[w] \mid w \in N_H(v)\}\!\})$$

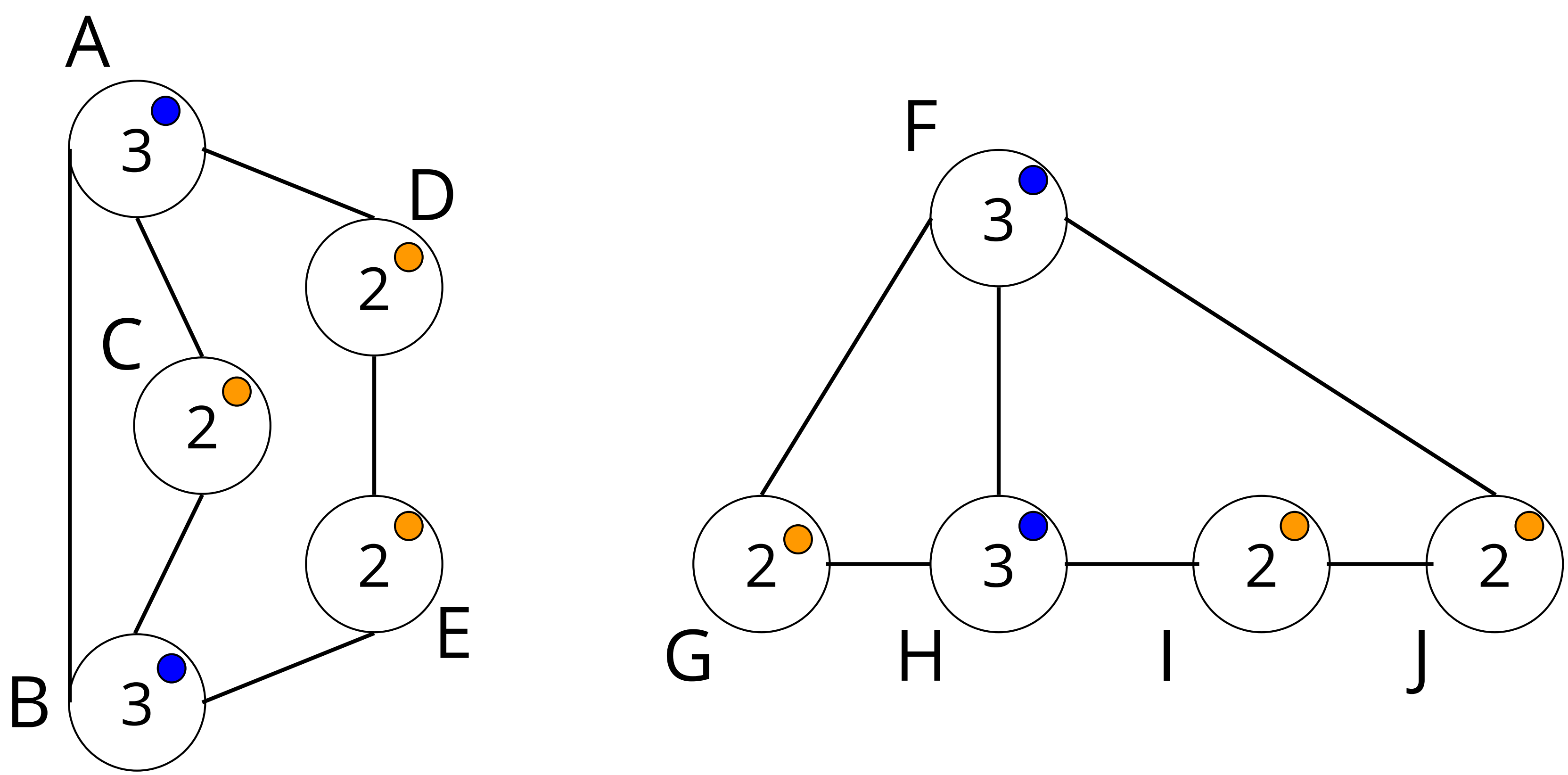
$$1 \quad : \quad C^{old}$$

$$\{\!\{1, 1, 1\}\!\} \quad : \quad \{\!\{C^{old}[w] \mid w \in N_H(v)\}\!\}$$



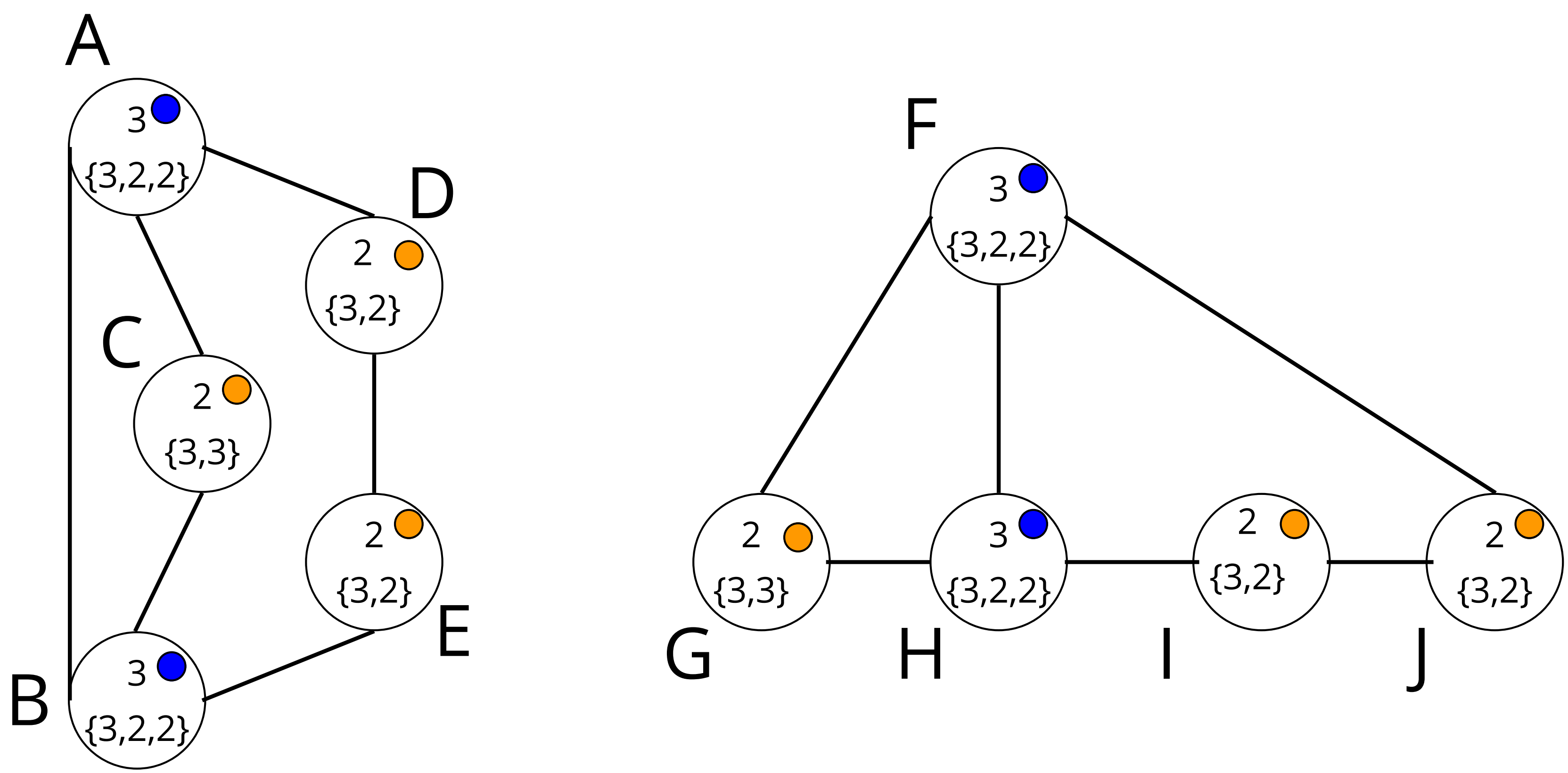
$$C[v] \leftarrow \mathrm{hash}(C^{old}[v], \{\!\!\{C^{old}[w] \mid w \in N_H(v)\}\!\!\})$$

$$3 \longleftarrow \begin{array}{ccc} 1 & : & C^{old} \\ \{\!\!\{1,1,1\}\!\!\} & : & \{\!\!\{C^{old}[w] \mid w \in N_H(v)\}\!\!\} \end{array}$$



3 : C^{old}

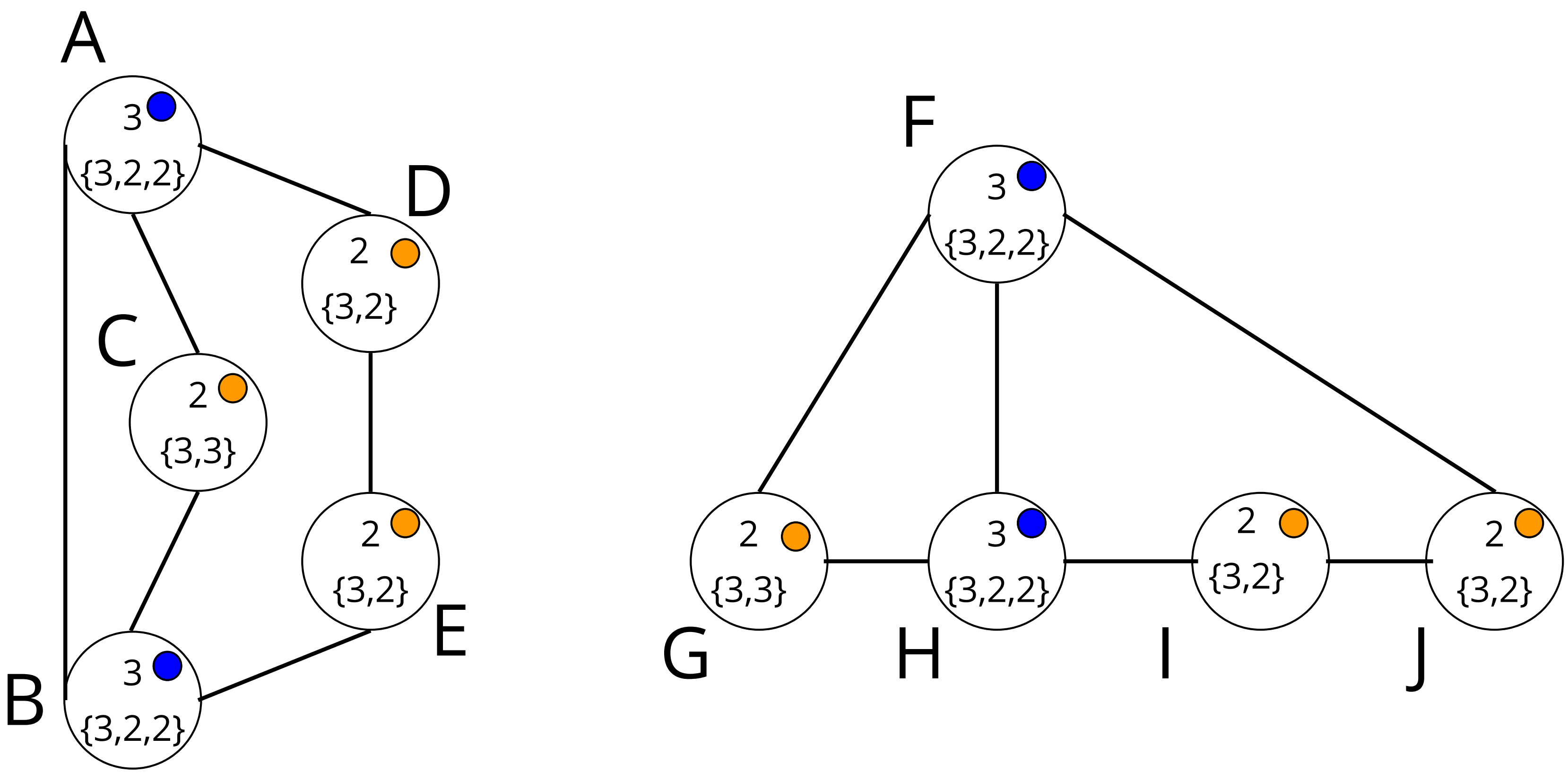
$\{\!\!\{3, 2, 2\}\!\!\}$: $\{\!\!\{C^{old}[w] \mid w \in N_H(v)\}\!\!\}$



$$C[v] \leftarrow \operatorname{hash}(C^{old}[v], \{\!\!\{C^{old}[w] \mid w \in N_H(v)\}\!\!\})$$

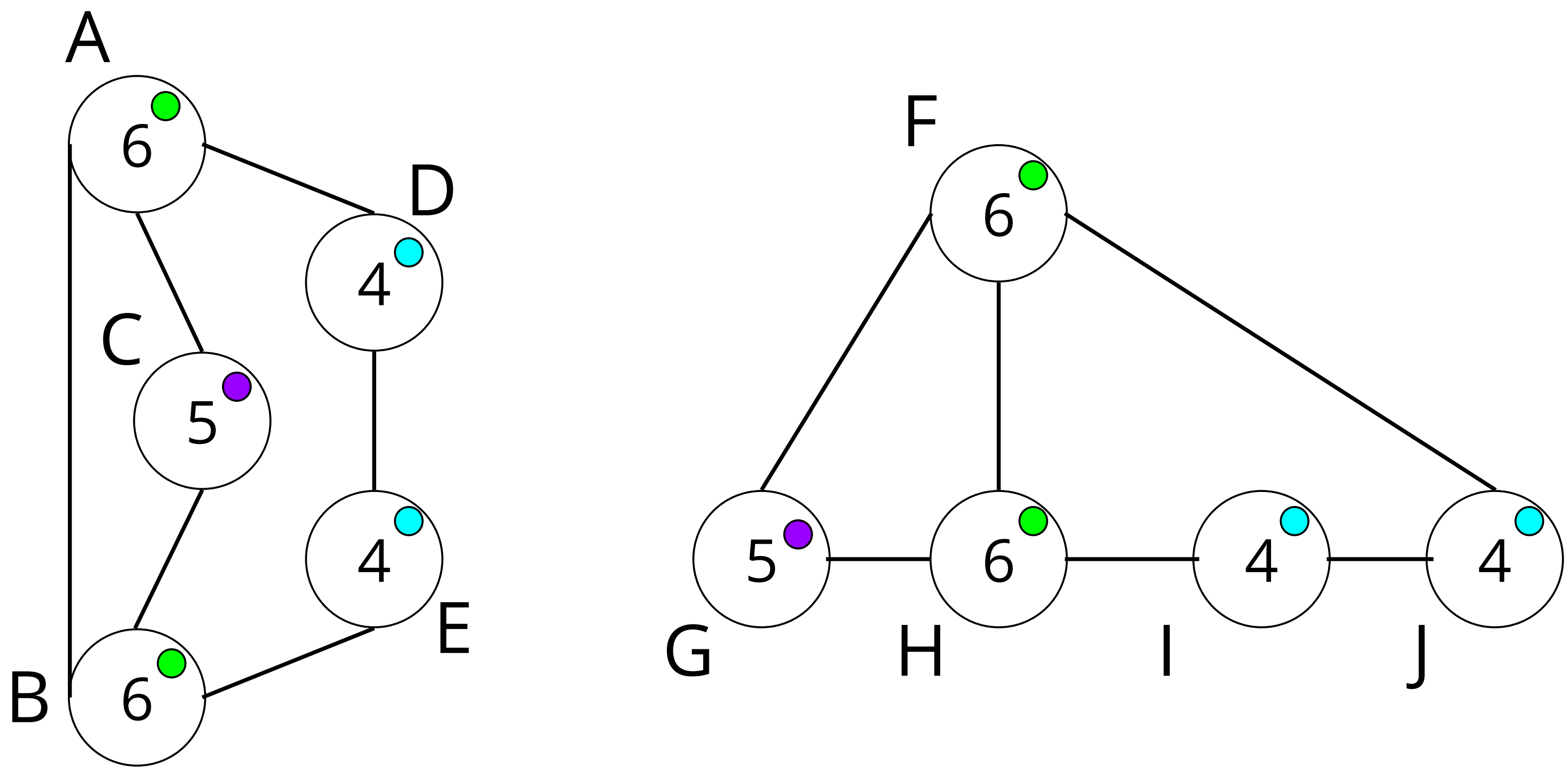
$$3 \quad : \quad C^{old}$$

$$\{\!\!\{3, 2, 2\}\!\!\} \quad : \quad \{\!\!\{C^{old}[w] \mid w \in N_H(v)\}\!\!\}$$



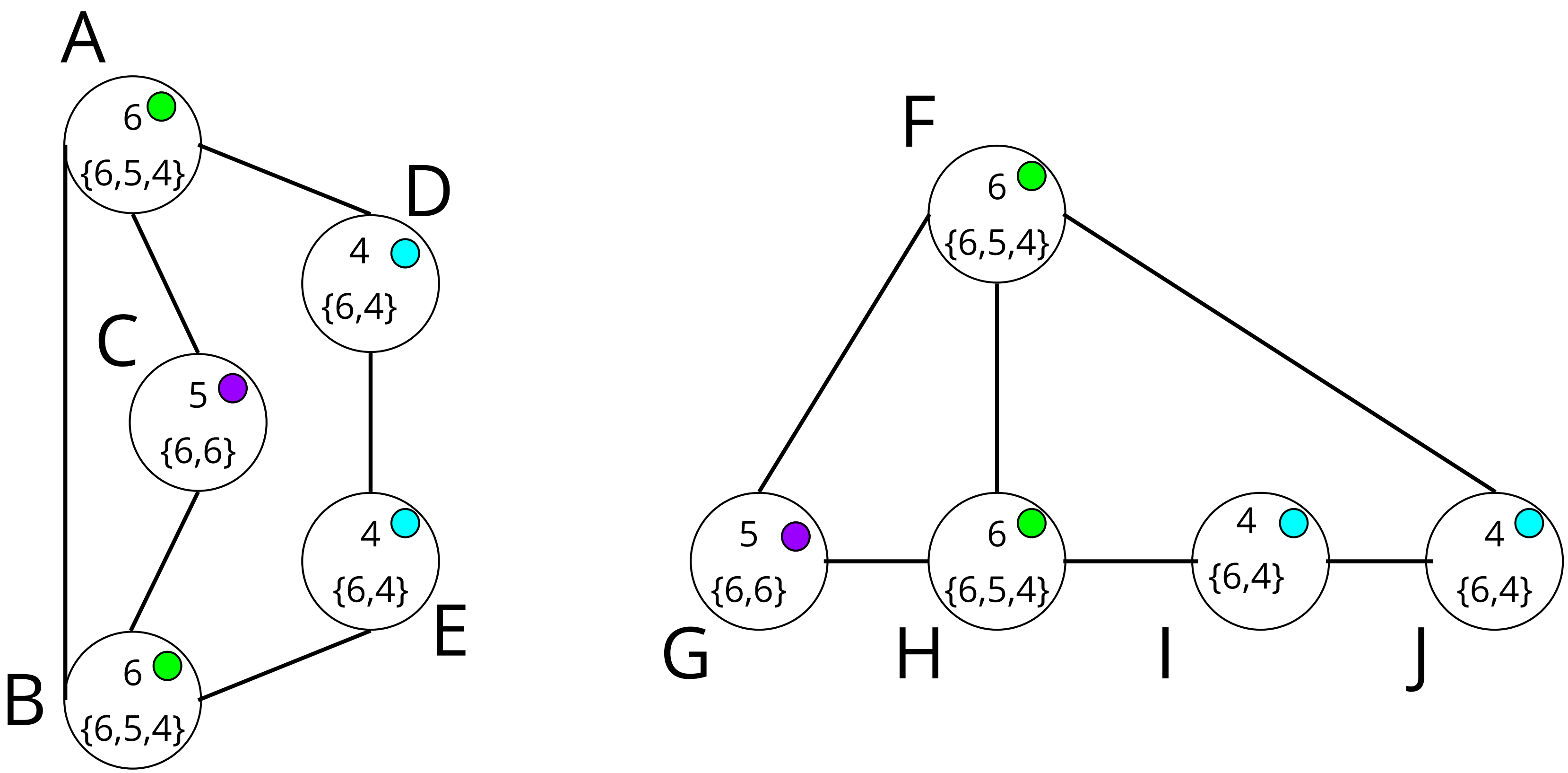
$$C[v] \leftarrow \mathrm{hash}(C^{old}[v], \{\!\!\{C^{old}[w] \mid w \in N_H(v)\}\!\!\})$$

$$6 \longleftarrow \begin{array}{ccl} 3 & : & C^{old} \\ \{\!\!\{3, 2, 2\}\!\!\} & : & \{\!\!\{C^{old}[w] \mid w \in N_H(v)\}\!\!\} \end{array}$$

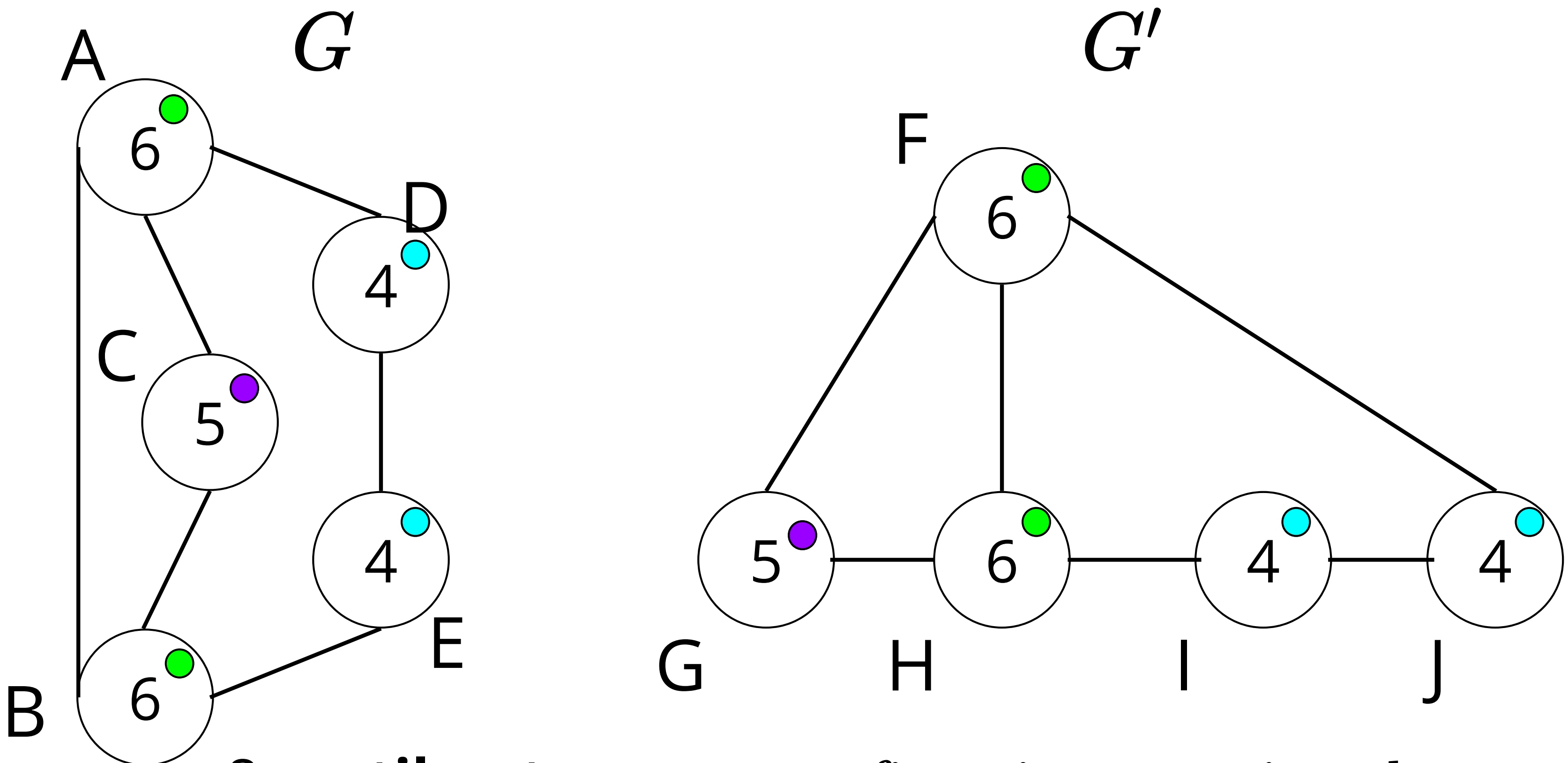


6 : C^{old}

$\{\![6,5,4]\!\}$: $\{\![C^{old}[w] \mid w \in N_H(v)]\!\}$



$$C[v] \leftarrow \mathrm{hash}(C^{old}[v], \{\!\![C^{old}[w] \mid w \in N_H(v)]\!\!\})$$



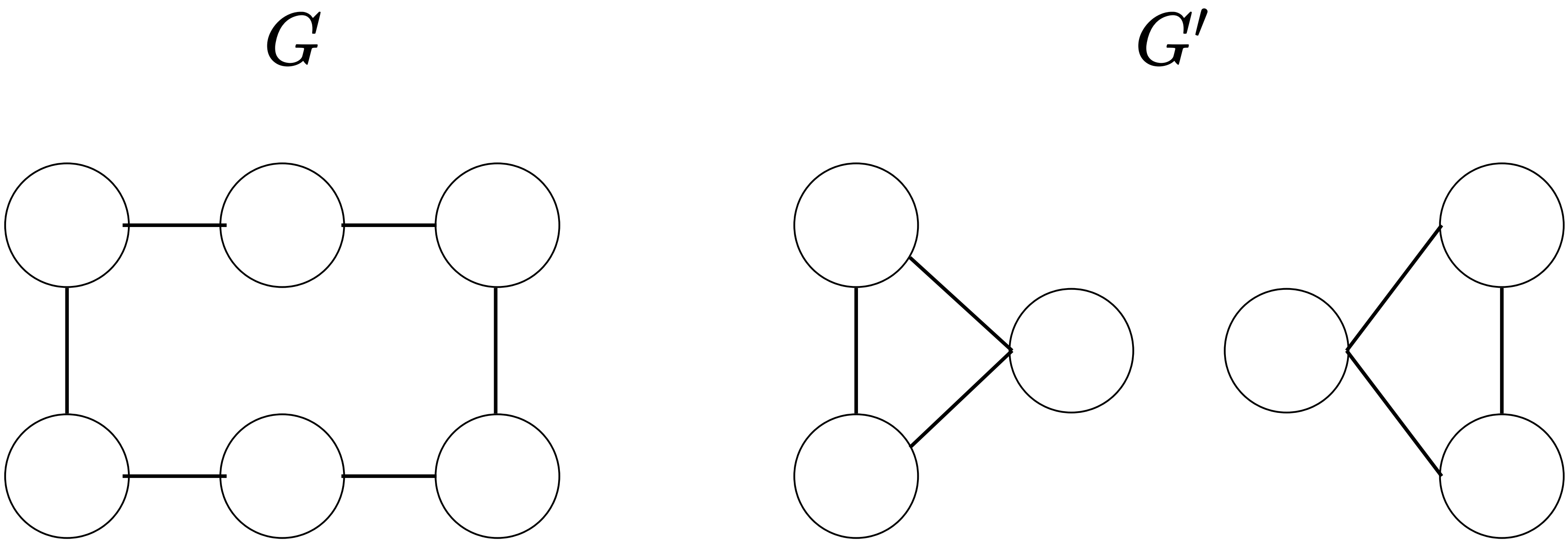
8. **until not** $\sim_C$ es un refinamiento extricto de $\sim_{C^{old}}$

9. **return** C

$C(G)$ $= \{$ D $\rightarrow$ 4, E $\rightarrow$ 4, C $\rightarrow$ 5, A $\rightarrow$ 6, B $\rightarrow$ 6 $\}$

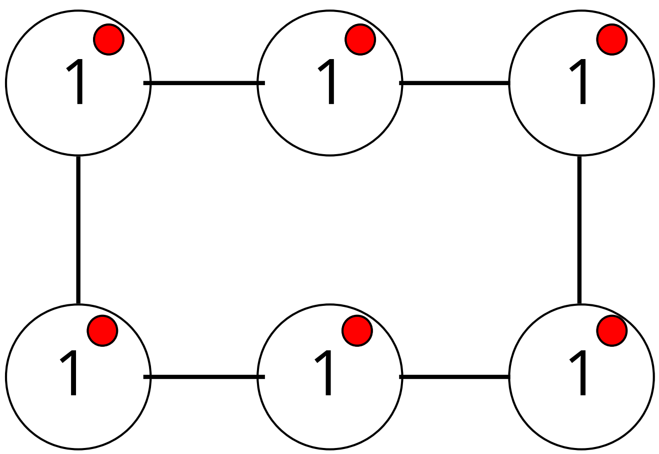
$C(G')$ $= \{$ I $\rightarrow$ 4, J $\rightarrow$ 4, G $\rightarrow$ 5, F $\rightarrow$ 6, H $\rightarrow$ 6 $\}$

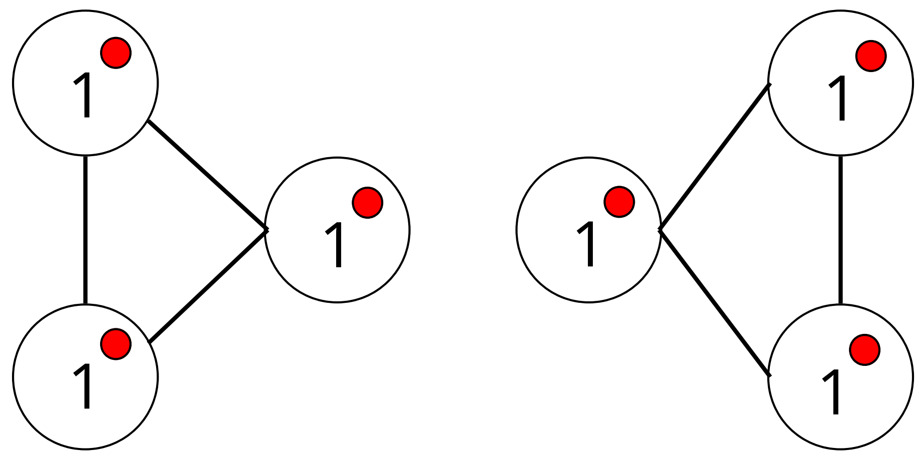
Test de isomorfismo WL



Algoritmo 1-WL

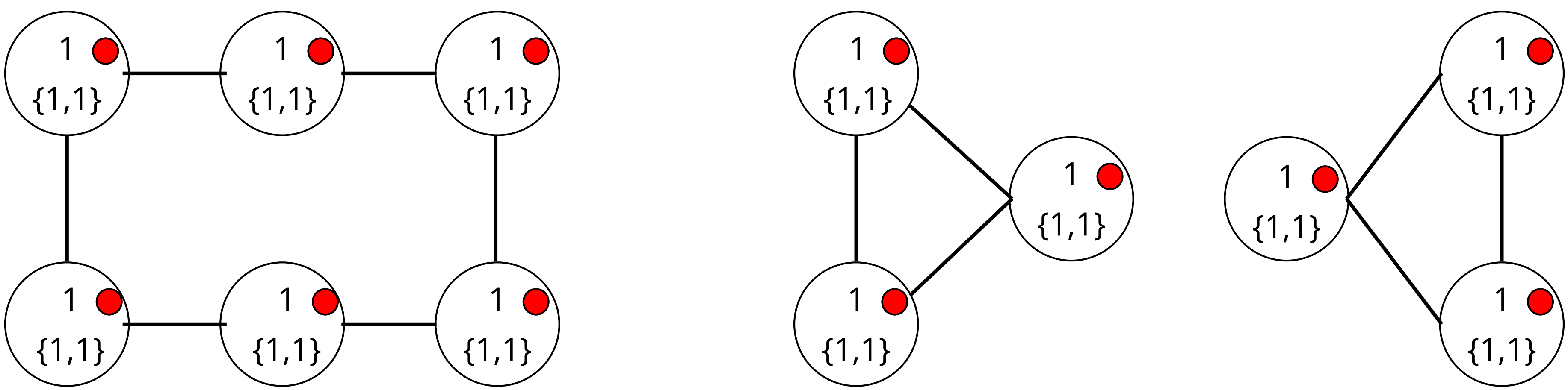
1. **for each** $v \in V$ **do**
2. $\quad C[v] \leftarrow 1$





1 : C^{old}

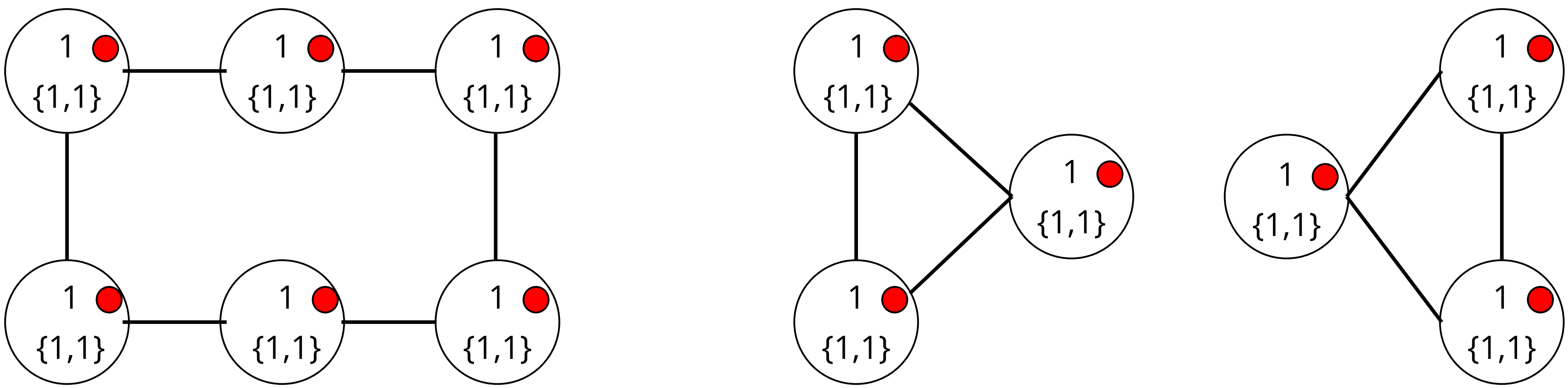
$\{\!\{1, 1\}\!\}$: $\{\!\{C^{old}[w] \mid w \in N_H(v)\}\!\}$



$$C[v] \leftarrow \text{hash}(C^{old}[v], \{\!\!\{C^{old}[w] \mid w \in N_H(v)\}\!\!\})$$

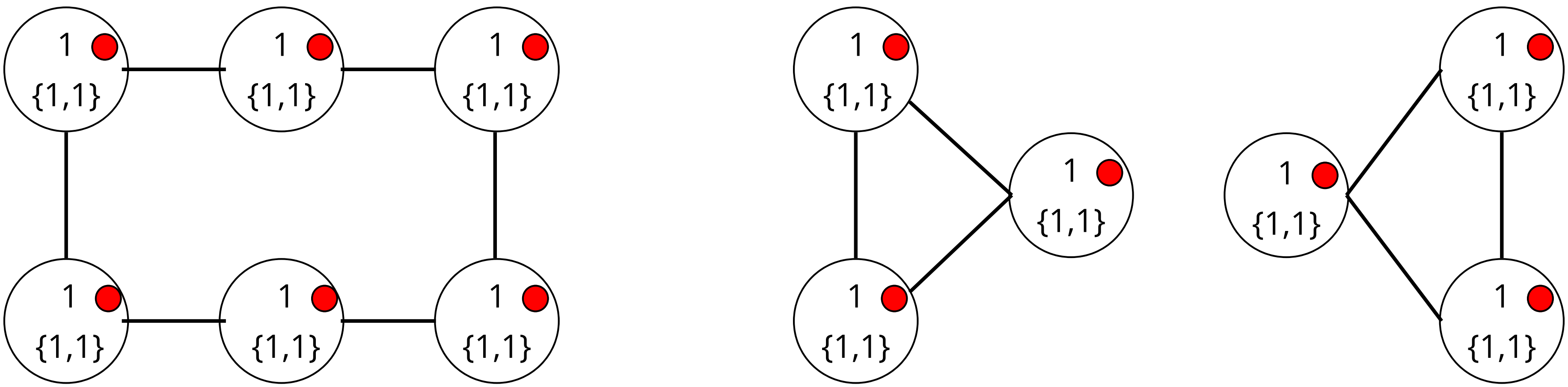
1 : C^{old}

$\{\!\!\{1, 1\}\!\!\}$: $\{\!\!\{C^{old}[w] \mid w \in N_H(v)\}\!\!\}$



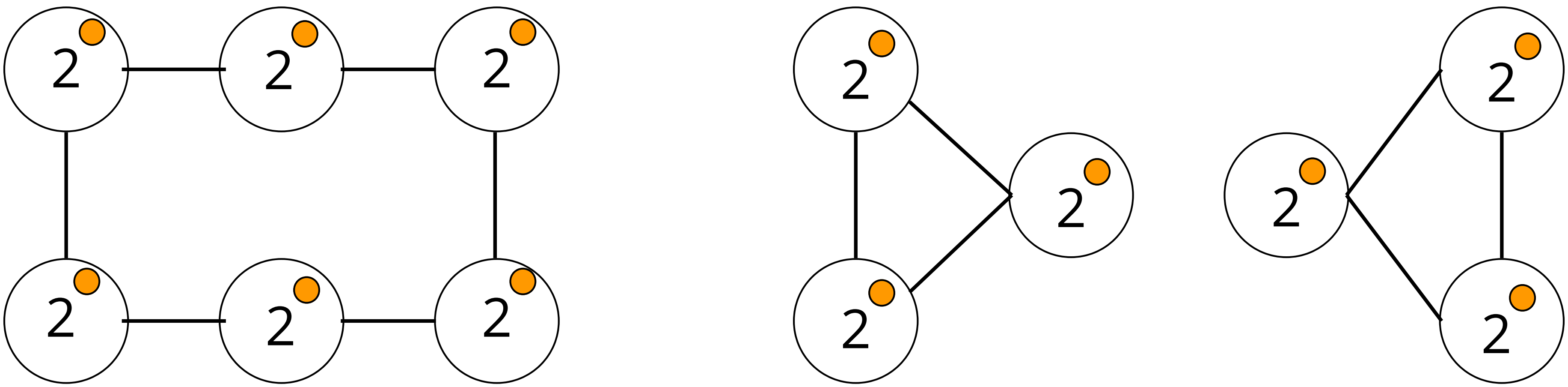
$$C[v] \leftarrow \mathrm{hash}(C^{old}[v], \{\!\!\{C^{old}[w] \mid w \in N_H(v)\}\!\!\})$$

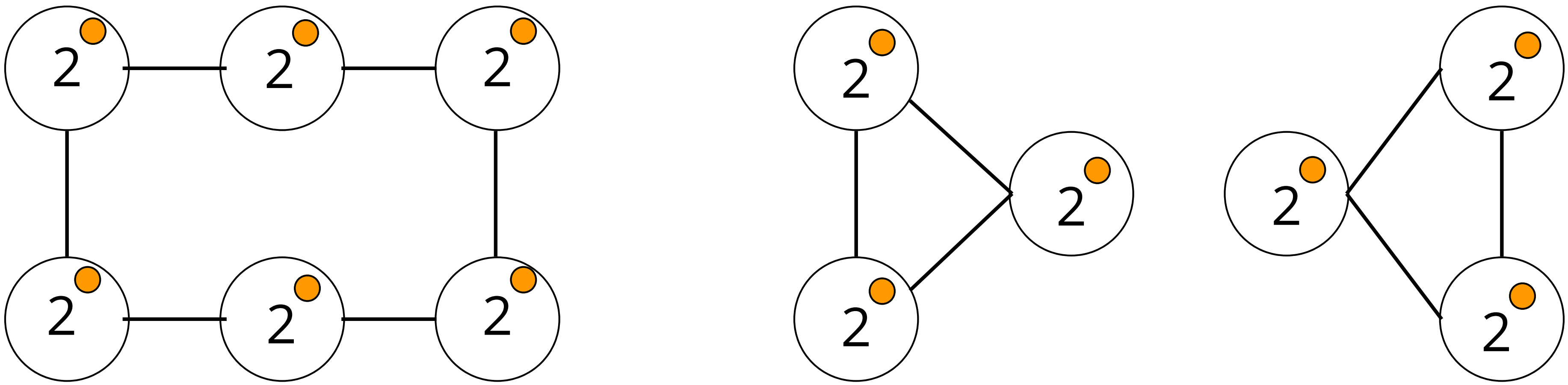
$$2 \longleftarrow \begin{array}{lcl} 1 & : & C^{old} \\ \{\!\!\{1, 1\}\!\!\} & : & \{\!\!\{C^{old}[w] \mid w \in N_H(v)\}\!\!\} \end{array}$$



$$C[v] \leftarrow \mathrm{hash}(C^{old}[v], \{\!\![C^{old}[w] \mid w \in N_H(v)]\!\!\})$$

$$2 \longleftarrow \begin{array}{lcl} 1 & : & C^{old} \\ \{\!\![1, 1]\!\!\} & : & \{\!\![C^{old}[w] \mid w \in N_H(v)]\!\!\} \end{array}$$





8. **until not** $\sim_C$ es un refinamiento extricto de $\sim_{C^{old}}$

9. **return** C

$$C(G) = \{2, 2, 2, 2, 2, 2\}$$
$$C(G') = \{2, 2, 2, 2, 2, 2\}$$

Pros de 1-WL

- Simple y fácil de implementar
- Produce formas normales para todas los grafos de n-vértices excepto una fracción $n^{-1/7}$

Cons de 1-WL

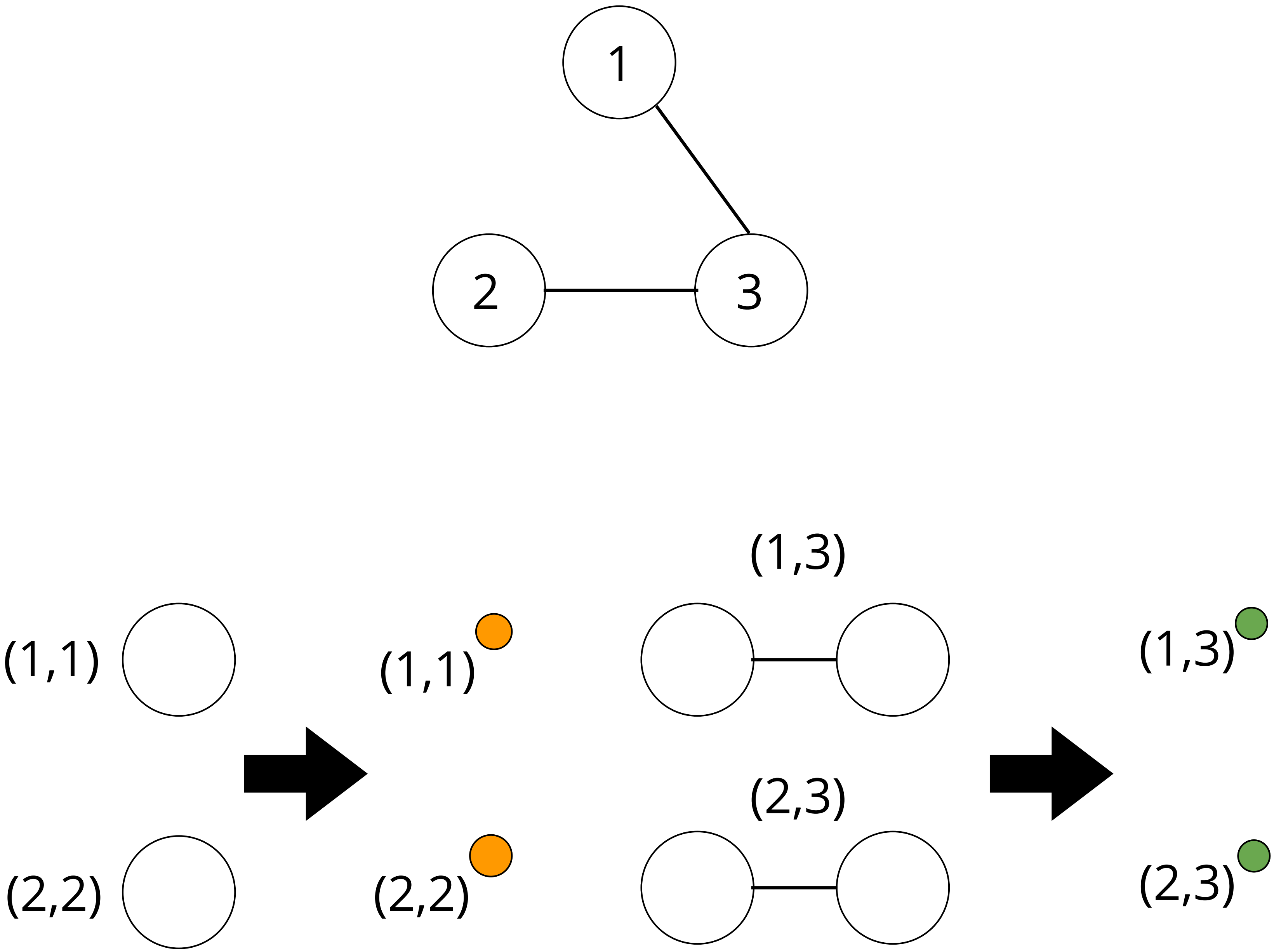
- Existe clases de grafos para los cuales el algoritmo es inútil, por ejemplo grafos regulares (el mismo grado en cada vértice)

Algoritmo k-WL

Coloración de tuplas

Dado un grafo $G = (V, E)$ y tuplas $\overline{u} = (u_1, \ldots, u_k)$ y $\overline{v} = (v_1, \ldots, v_k)$ en V^k.

$\overline{u}$ y $\overline{v}$ son pintadas del mismo color si la función $f(u_i) = v_i$ es un isomorfismo entre los subgrafos de G inducidos por $\{u_1, \ldots, u_k\}$ y $\{v_1, \ldots, v_k\}$.



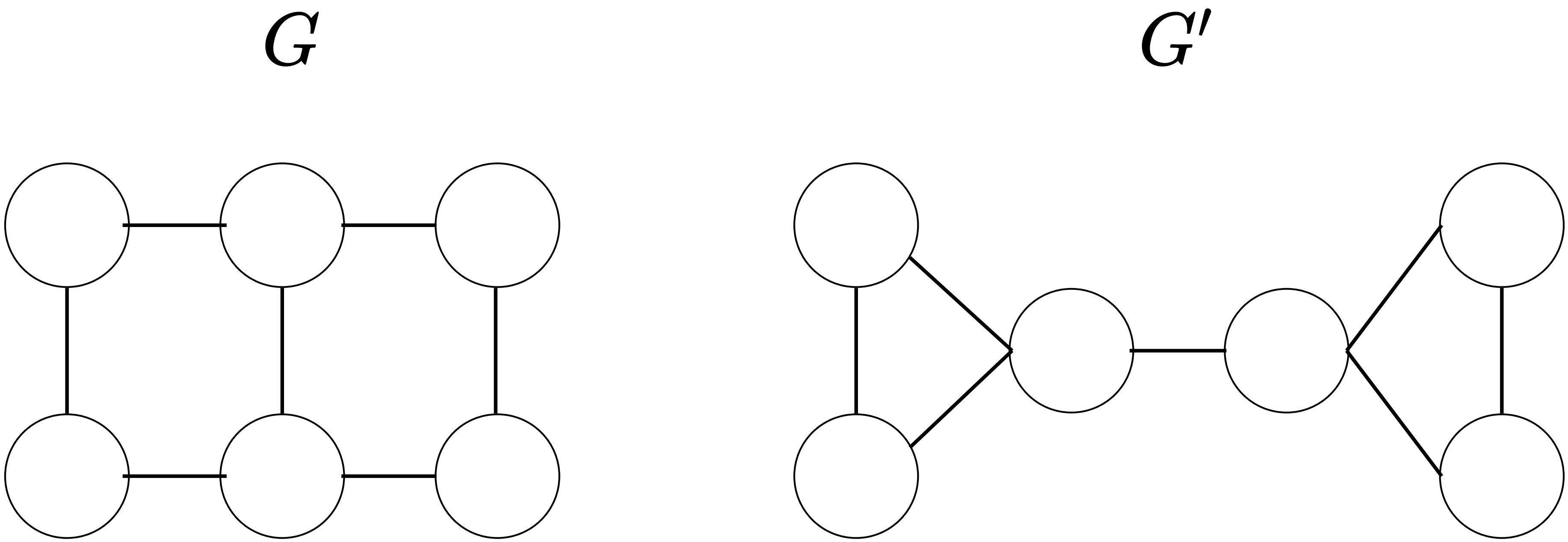
1
2
3
(1,1)
(2,2)
(1,1)
(2,2)
(1,3)
(2,3)
(1,3)
(2,3)

Algoritmo 2-WL

Input: $G = (V, E)$

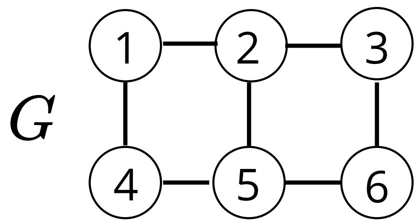
1. $C \leftarrow$ **ColoreoInicial**(G)
2. **repeat**
3. $\quad C^{old} \leftarrow C$
4. $\quad$ **for each** $(u, v) \in V^2$ **do**
5. $\quad\quad \chi[1] \leftarrow \{\!\!\{ C^{old}[(n, v)] \mid n \in V \}\!\!\}$
6. $\quad\quad \chi[2] \leftarrow \{\!\!\{ C^{old}[(u, n)] \mid n \in V \}\!\!\}$
7. $\quad\quad C[(u, v)] \leftarrow \mathrm{hash}\Big(C^{old}[(u, v)], \chi[1], \chi[2] \Big)$
8. **until not** $\sim_C$ es un refinamiento extricto de $\sim_{C^{old}}$
9. **return** C

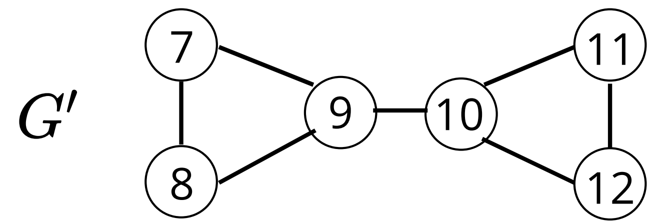
Test de isomorfismo 2-WL



¿Qué resultado da la 1-WL sobre estos grafos?

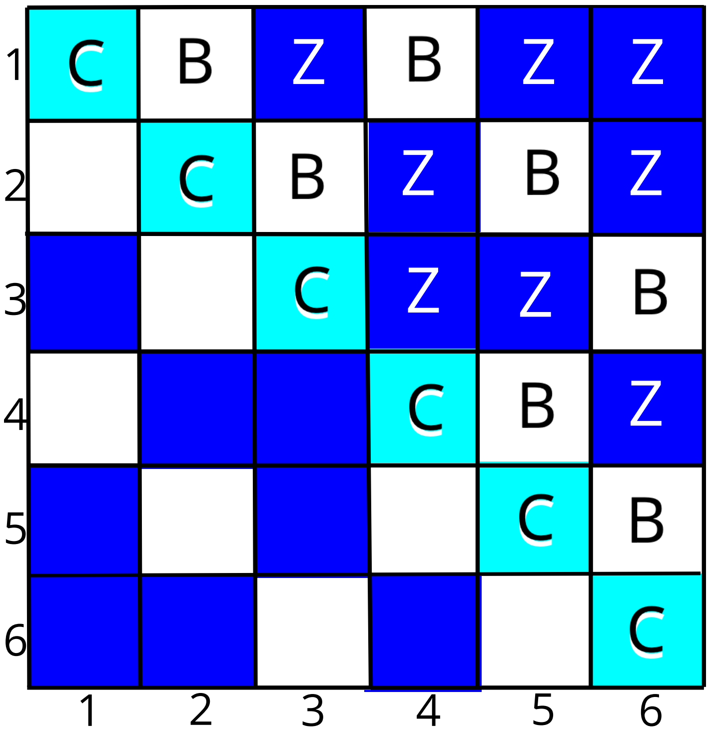
ColoreoInicial(G)

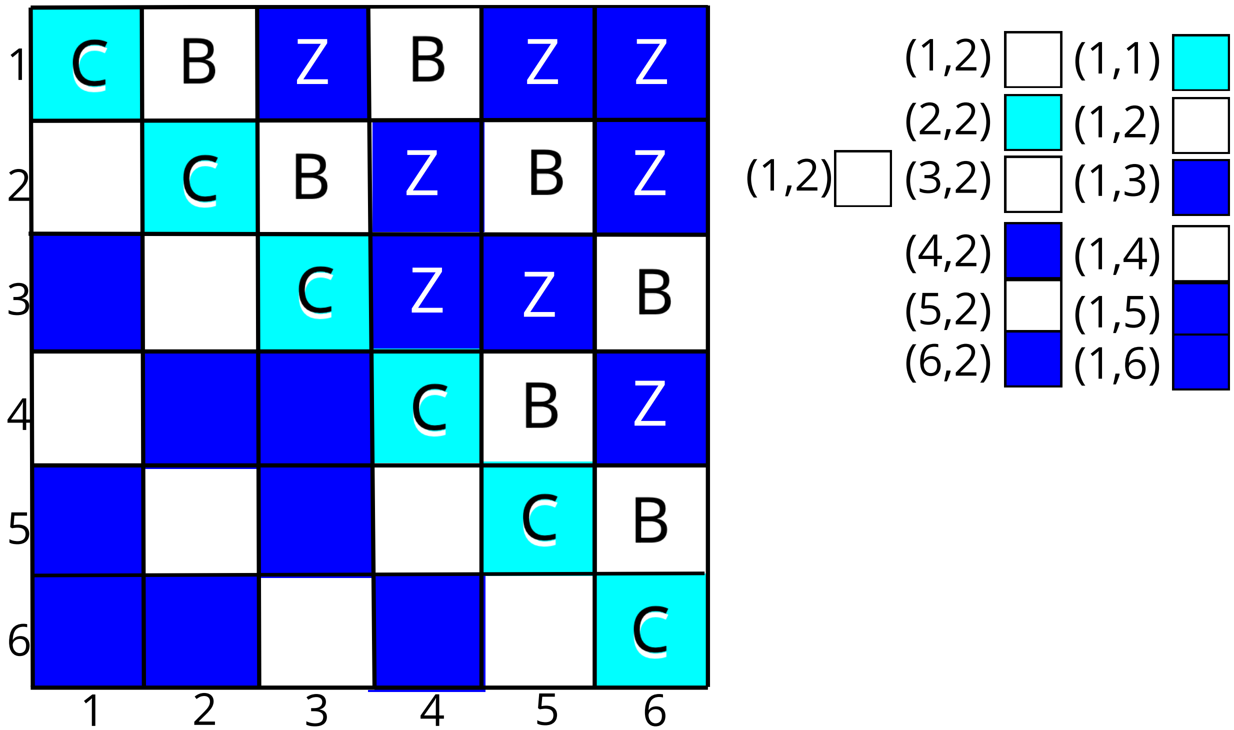




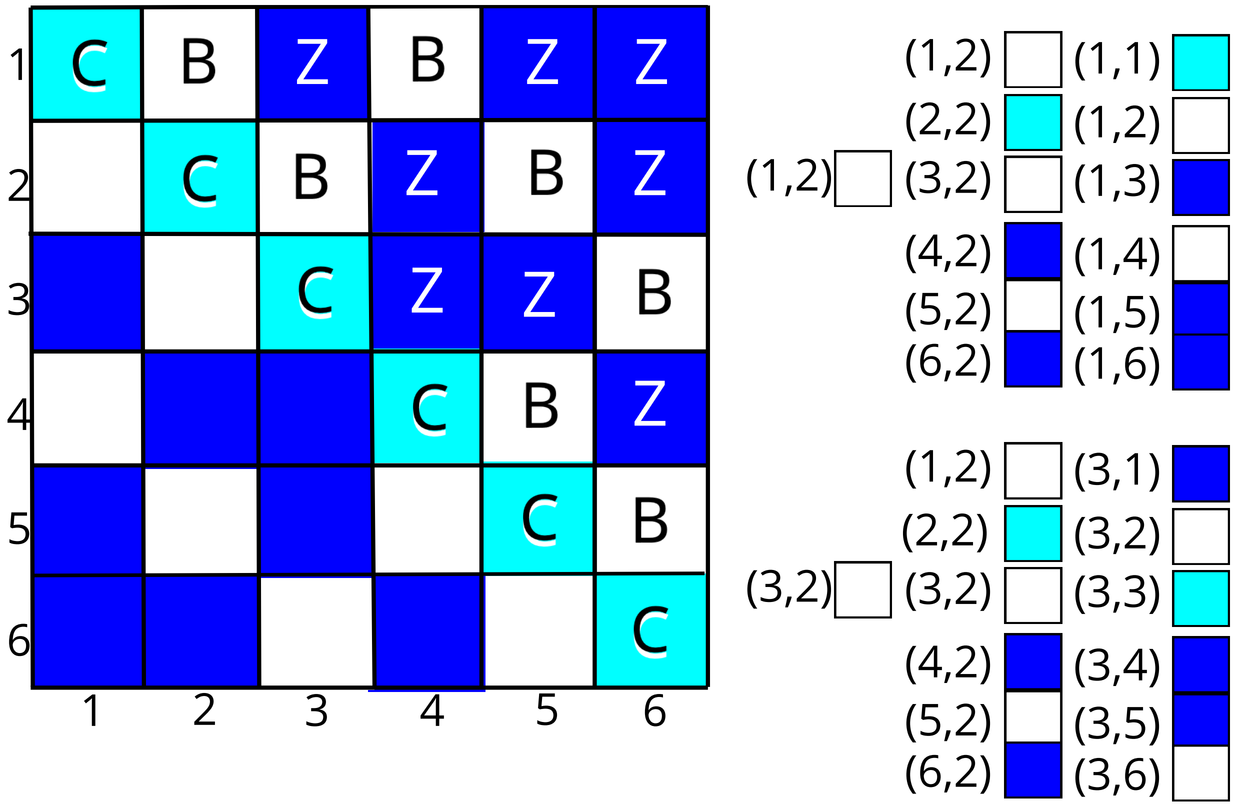
	1	2	3	4	5	6
1	C	B	Z	B	Z	Z
2		C	B	Z	B	Z
3			C	Z	Z	B
4				C	B	Z
5					C	B
6						C

	7	8	9	10	11	12
7	C	B	B	Z	Z	Z
8		C	B	Z	Z	Z
9			C	B	Z	Z
10				C	B	B
11					C	B
12						C

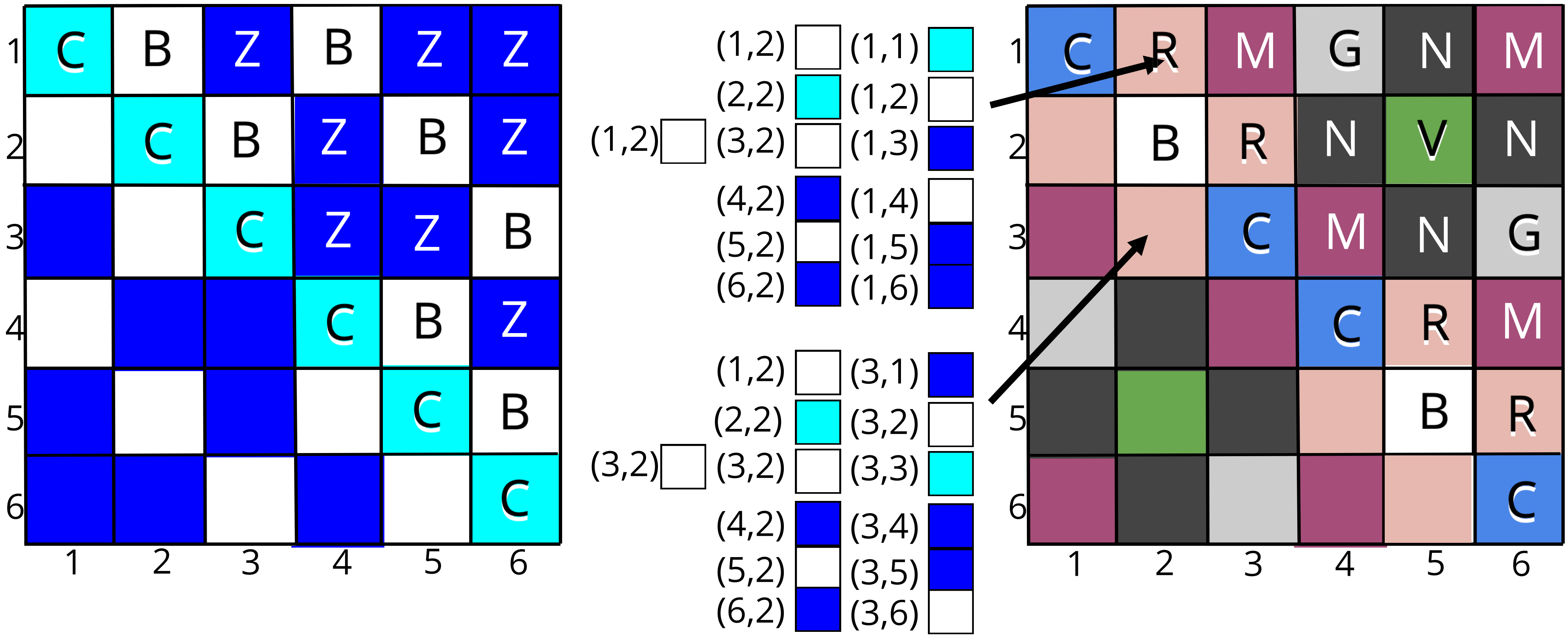

1 C B Z B Z Z
2 C B Z B Z
3 C Z Z B
4 C B Z
5 C B
6 C
1 2 3 4 5 6



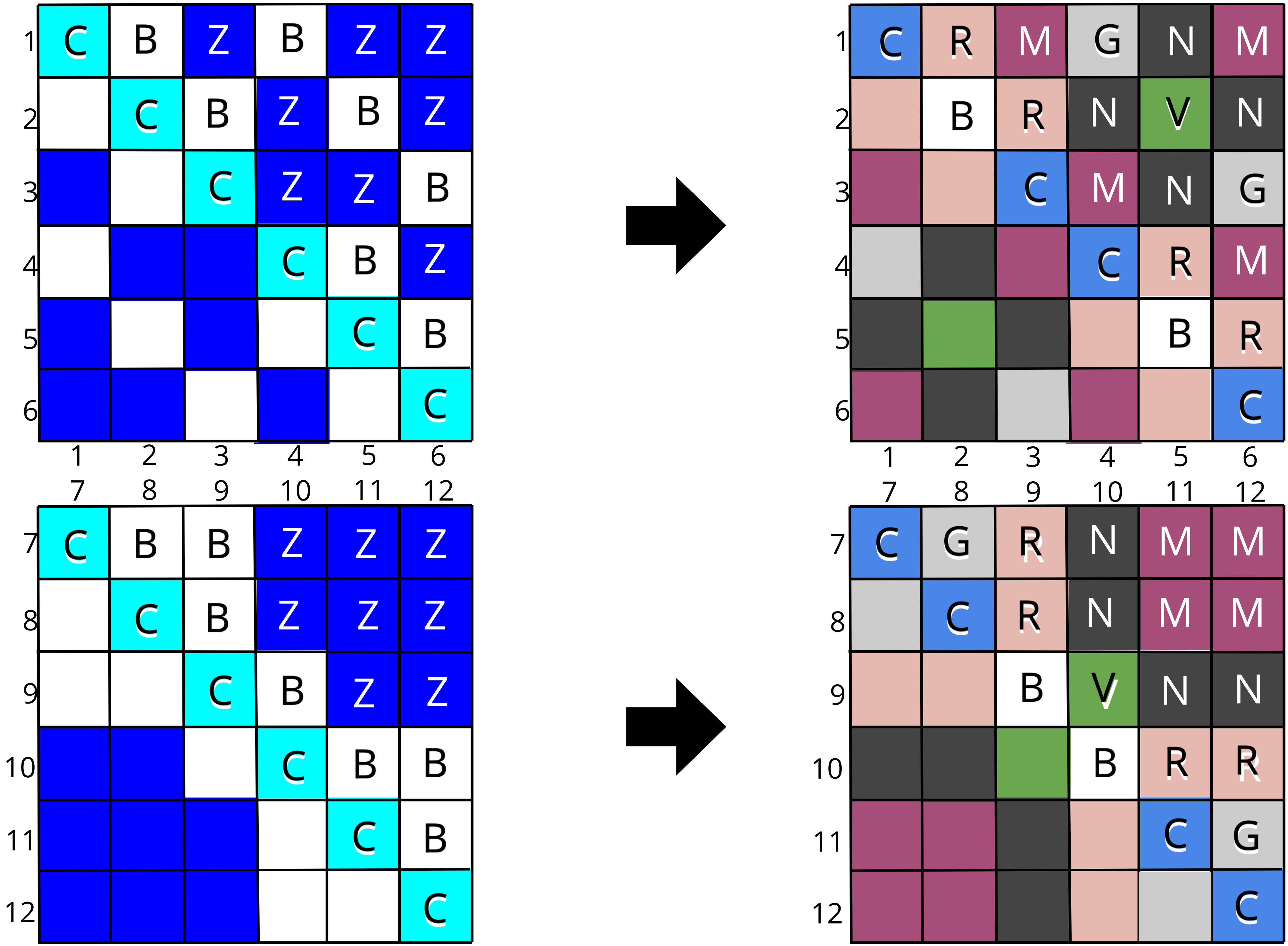
C B Z B Z Z
C B Z B Z
C Z Z B
C B Z
C B
C
1 2 3 4 5 6
(1,2)
(1,2)
(2,2)
(3,2)
(4,2)
(5,2)
(6,2)
(1,1)
(1,2)
(1,3)
(1,4)
(1,5)
(1,6)

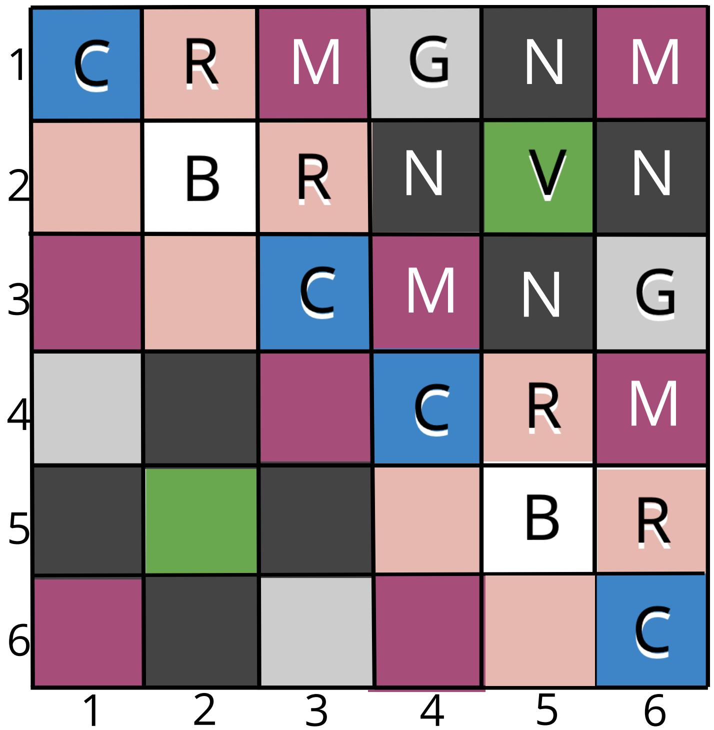


C B Z B Z Z
C B Z B Z
C Z Z B
C B Z
C B
C
1 2 3 4 5 6
(1,2)
(1,2) (2,2) (3,2) (4,2) (5,2) (6,2)
(1,1) (1,2) (1,3) (1,4) (1,5) (1,6)
(3,2)
(1,2) (2,2) (3,2) (4,2) (5,2) (6,2)
(3,1) (3,2) (3,3) (3,4) (3,5) (3,6)

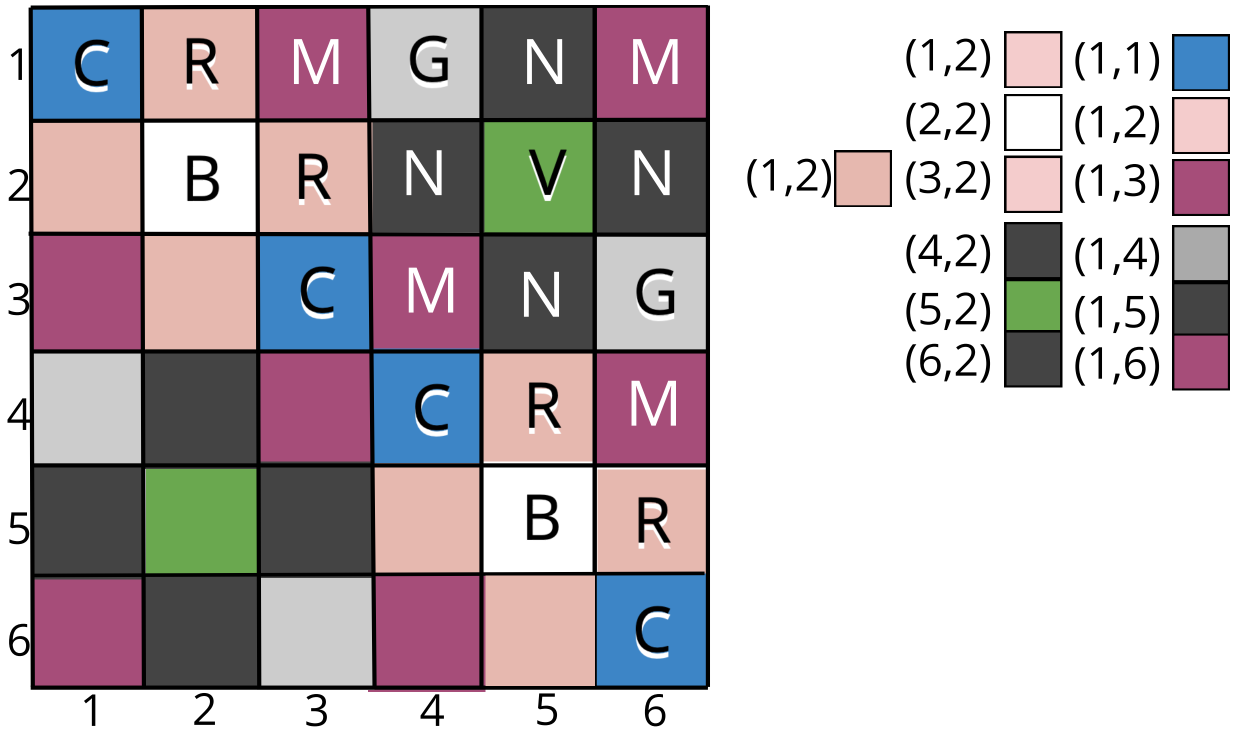


C B Z B Z Z
C B Z B Z
C Z Z B
C B Z
C B
C
(1,2) (2,2) (3,2) (4,2) (5,2) (6,2)
(1,1) (1,2) (1,3) (1,4) (1,5) (1,6)
(3,2) (2,2) (3,2) (4,2) (5,2) (6,2)
(3,1) (3,2) (3,3) (3,4) (3,5) (3,6)
C R M G N M
B R N V N
C M N G
C R M
B R
C
1 2 3 4 5 6

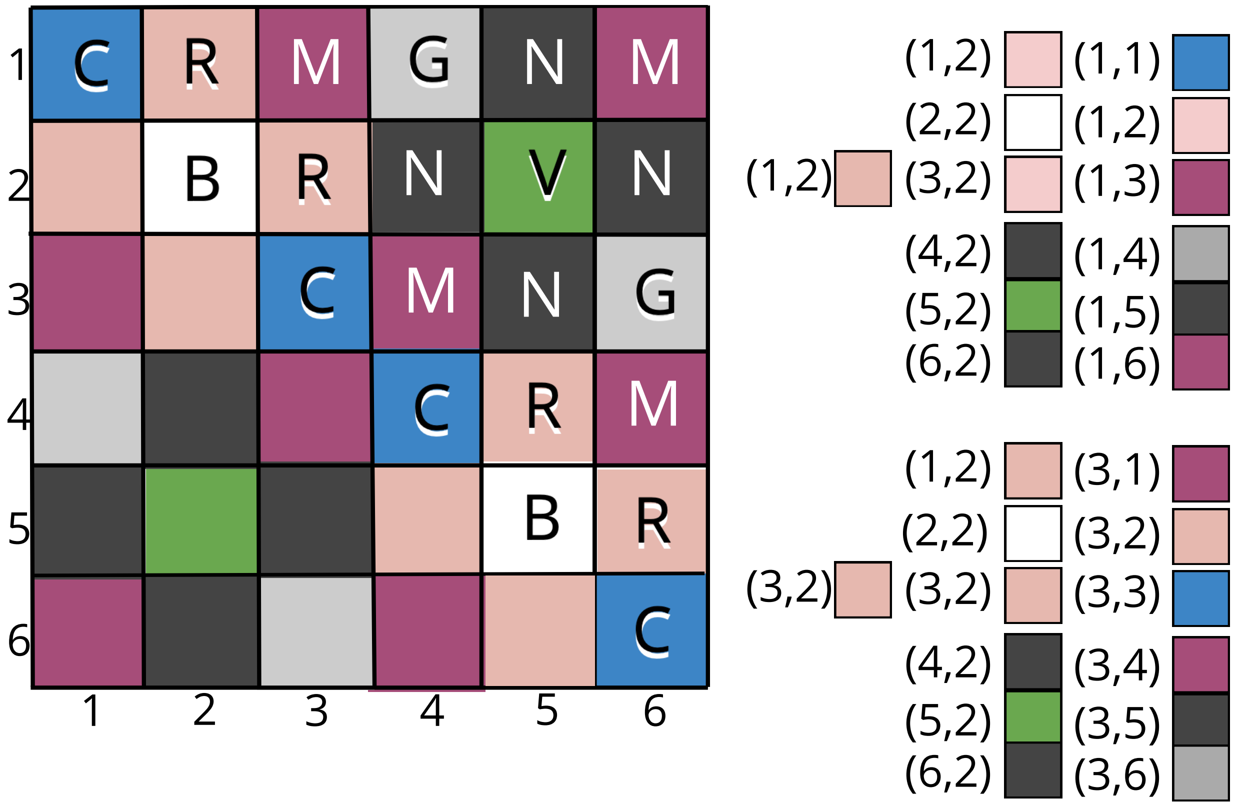

1 C B Z B Z Z
2 C B Z B Z
3 C Z Z B
4 C B Z
5 C B
6 C
1 2 3 4 5 6
7 8 9 10 11 12
7 C B B Z Z Z
8 C B Z Z Z
9 C B Z Z
10 C B B
11 C B
12 C
1 C R M G N M
2 B R N V N
3 C M N G
4 C R M
5 B R
6 C
1 2 3 4 5 6
7 8 9 10 11 12
7 C G R N M M
8 C R N M M
9 B V N N
10 B R R
11 C G
12 C



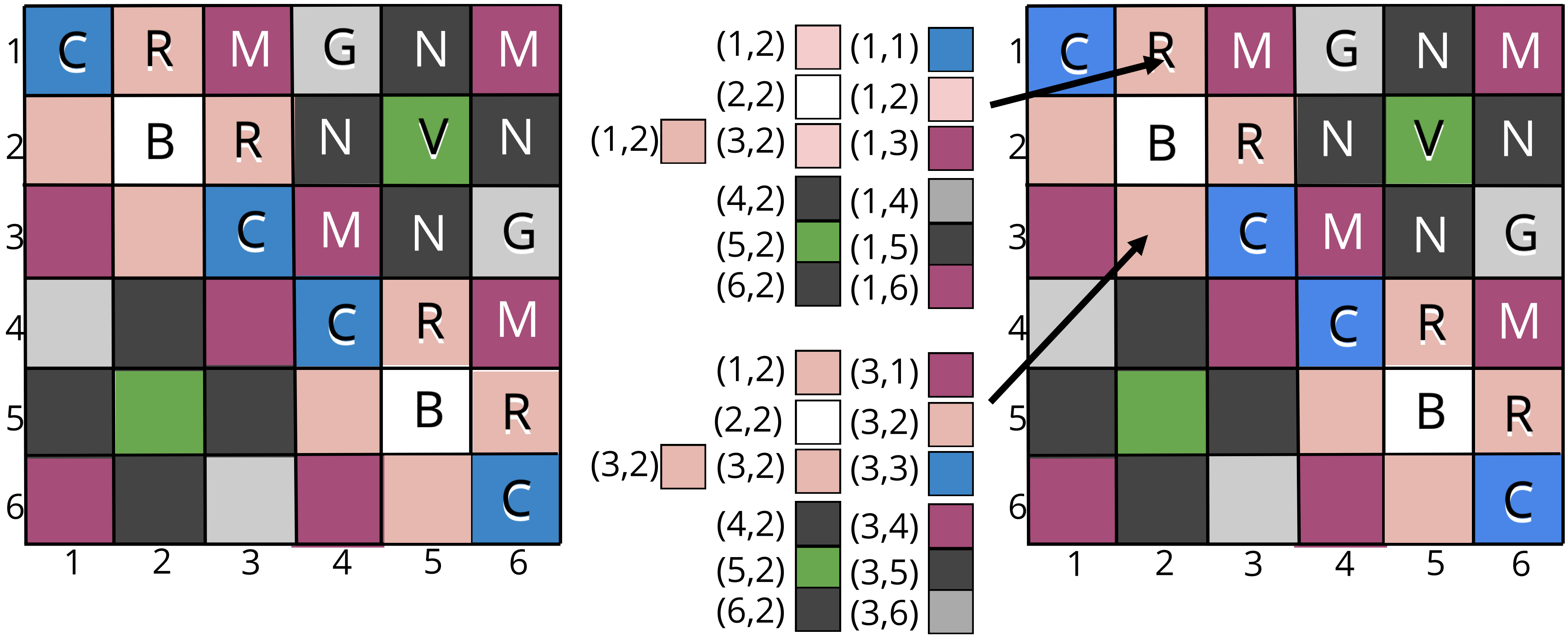
1
C R M G N M
2
B R N V N
3
C M N G
4
C R M
5
B R
6
C
1 2 3 4 5 6



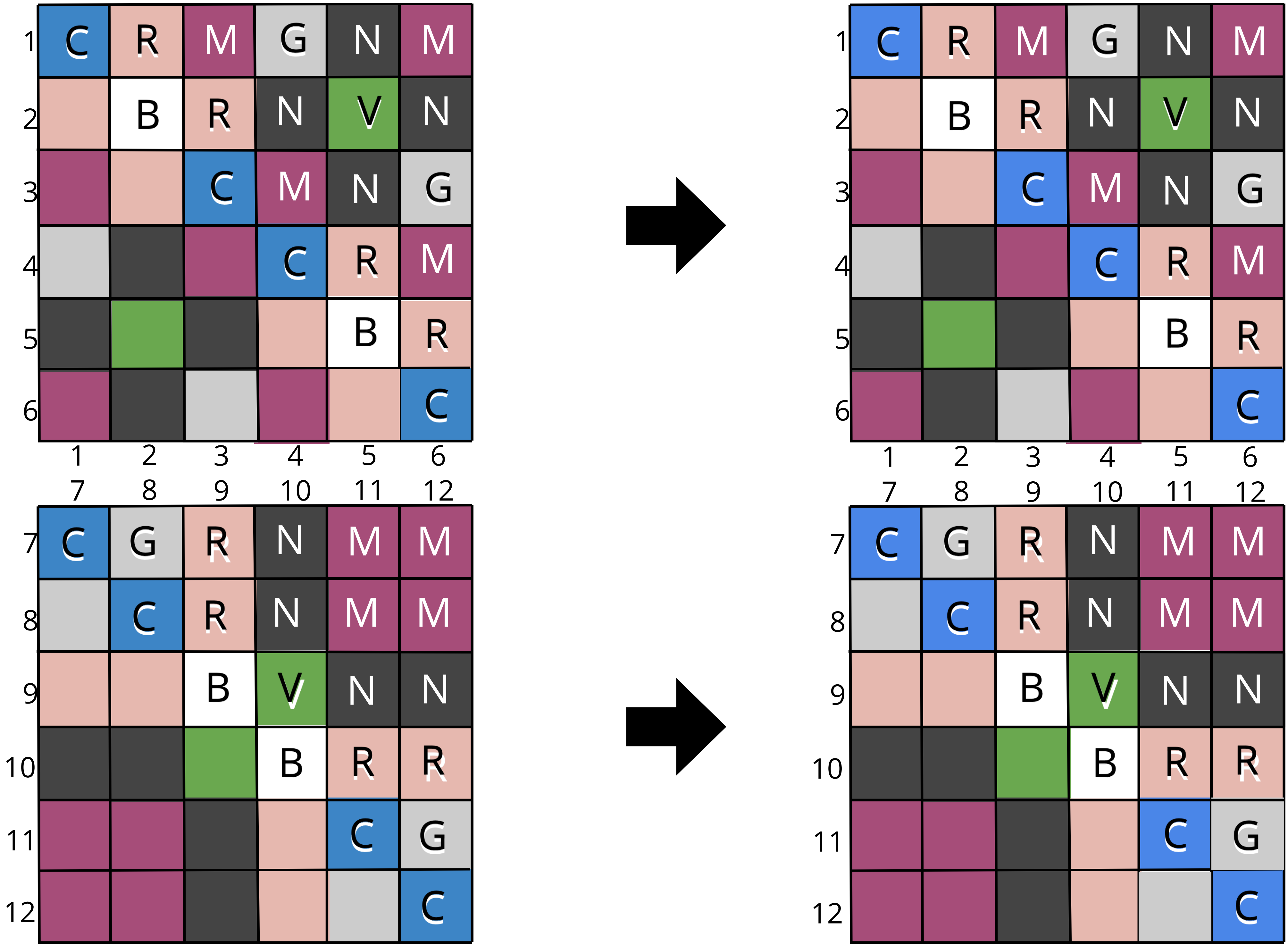
1 C R M G N M
2 B R N V N
3 C M N G
4 C R M
5 B R
6 C
1 2 3 4 5 6
(1,2)
(1,2) (1,1)
(2,2) (1,2)
(3,2) (1,3)
(4,2) (1,4)
(5,2) (1,5)
(6,2) (1,6)

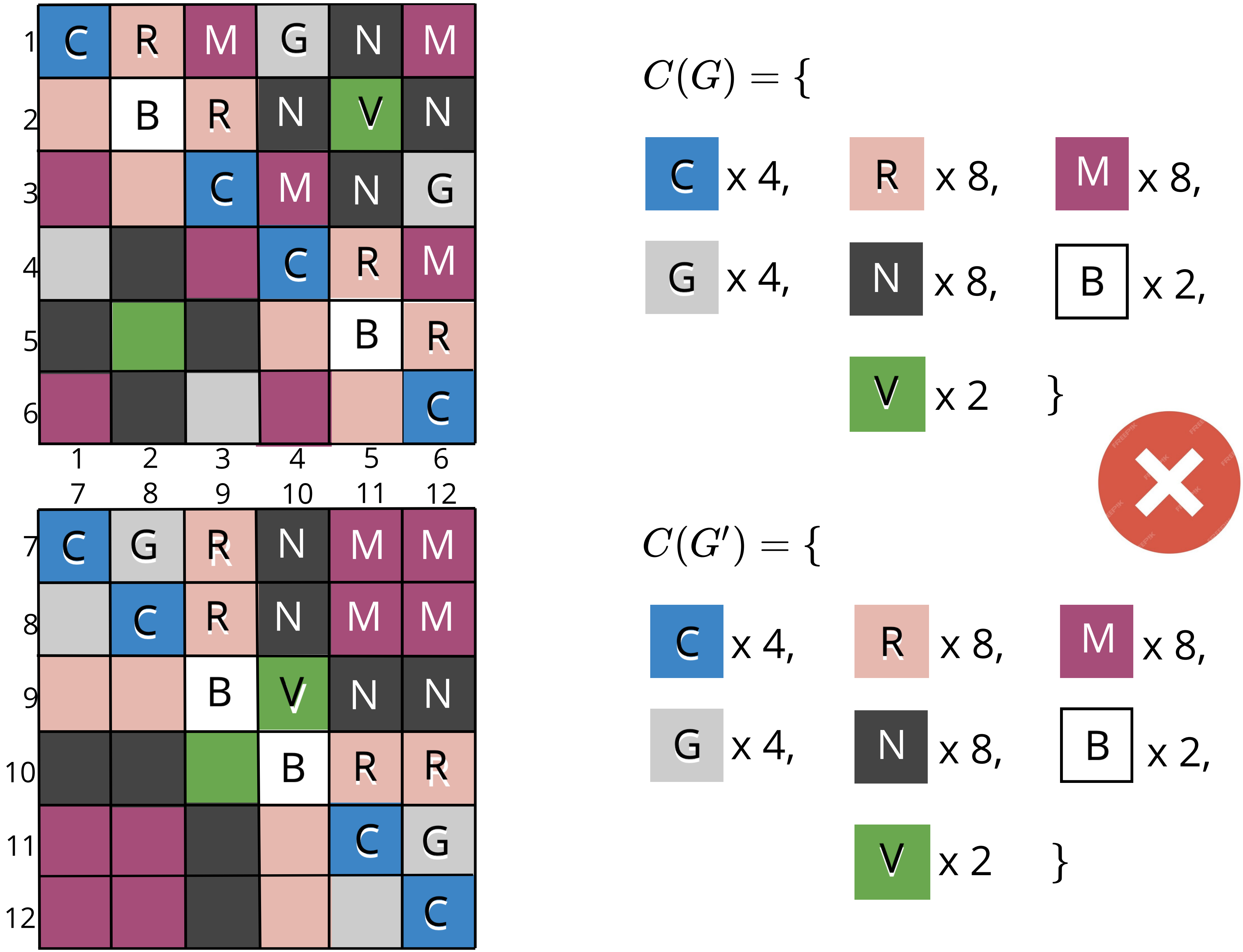


C R M G N M
B R N V N
C M N G
C R M
B R
C
1 2 3 4 5 6
(1,2) (1,1)
(2,2) (1,2)
(1,2) (3,2) (1,3)
(4,2) (1,4)
(5,2) (1,5)
(6,2) (1,6)
(1,2) (3,1)
(2,2) (3,2)
(3,2) (3,2) (3,3)
(4,2) (3,4)
(5,2) (3,5)
(6,2) (3,6)

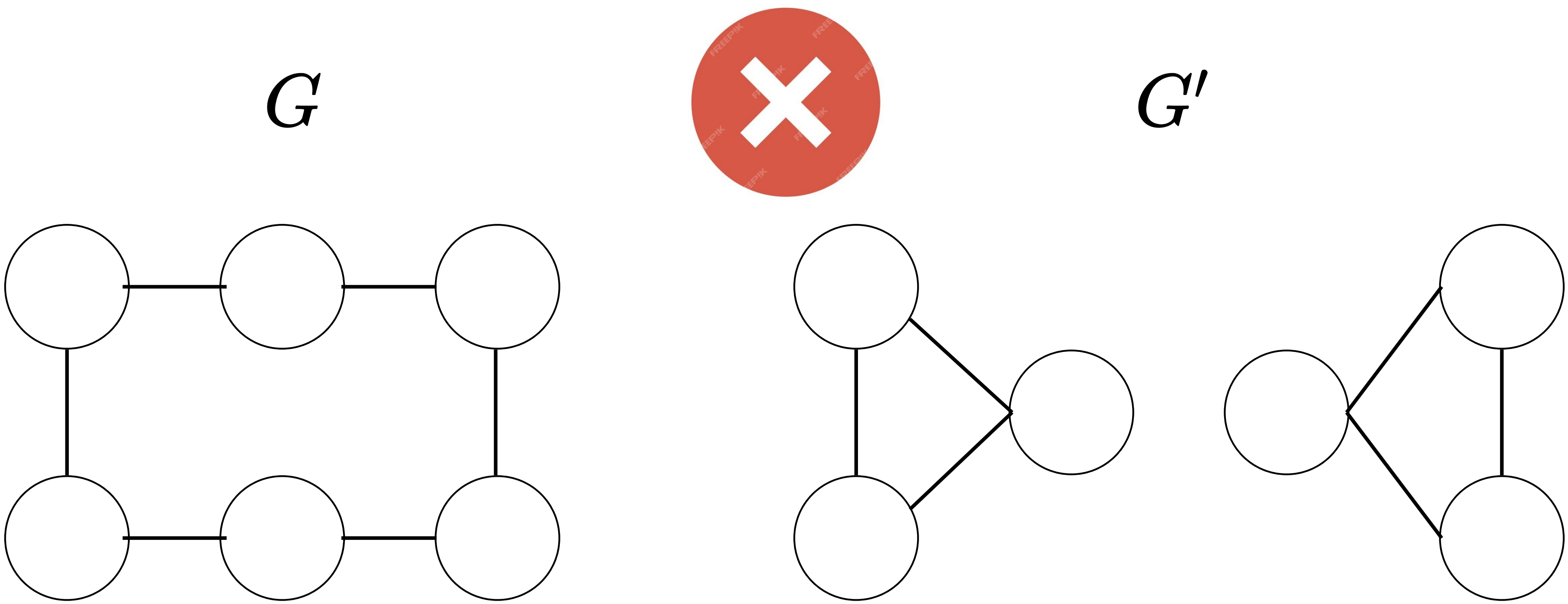


1 C R M G N M
2 B R N V N
3 C M N G
4 C R M
5 B R
6 C
1 2 3 4 5 6
(1,2)
(1,2) (2,2) (3,2) (4,2) (5,2) (6,2)
(1,1) (1,2) (1,3) (1,4) (1,5) (1,6)
(3,2)
(1,2) (2,2) (3,2) (4,2) (5,2) (6,2)
(3,1) (3,2) (3,3) (3,4) (3,5) (3,6)
1 C R M G N M
2 B R N V N
3 C M N G
4 C R M
5 B R
6 C
1 2 3 4 5 6


1 C R M G N M
2 B R N V N
3 C M N G
4 C R M
5 B R
6 C
1 2 3 4 5 6
7 8 9 10 11 12
7 C G R N M M
8 C R N M M
9 B V N N
10 B R R
11 C G
12 C
1 C R M G N M
2 B R N V N
3 C M N G
4 C R M
5 B R
6 C
1 2 3 4 5 6
7 8 9 10 11 12
7 C G R N M M
8 C R N M M
9 B V N N
10 B R R
11 C G
12 C


C R M G N M
B R N V N
C M N G
C R M
B R
C
1 2 3 4 5 6
7 8 9 10 11 12
C G R N M M
C R N M M
B V N N
B R R
C G
C
C(G) = {
C x 4, R x 8, M x 8,
G x 4, N x 8, B x 2,
V x 2 }
C(G′) = {
C x 4, R x 8, M x 8,
G x 4, N x 8, B x 2,
V x 2 }

Test de isomorfismo 2-WL



¿Qué resultado da 2-WL sobre estos grafos?

1-WL es equivalente a 2-WL

(Weisfeiller & Lehman, 1969)

Algoritmo k-WL

Input: $G = (V, E)$

0. $C \leftarrow$ **ColoreoInicial**(G)

2. **repeat**

3. $\quad C^{old} \leftarrow C$

4. $\quad$ **for each** $\overline{v} \in V^k$ **do**

5. $\qquad$ **for** $i = 1$ to k **do**

6. $\qquad\quad \chi[i] \leftarrow \{\!\!\{ C^{old}[\overline{w}] \mid \overline{w} \in N_i(\overline{v}) \}\!\!\}$

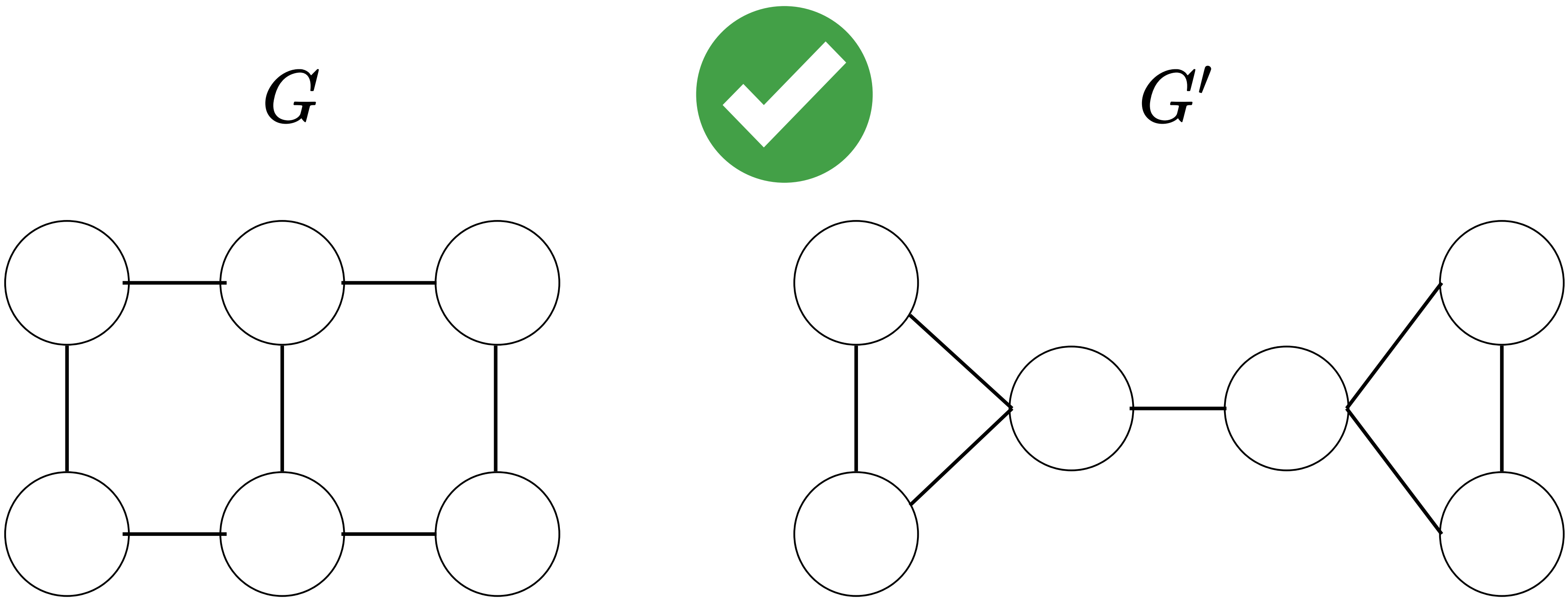
7. $\qquad C[\overline{v}] \leftarrow \text{hash}\Big(c^{old}[\overline{v}], \chi[1], ..., \chi[k] \Big)$

8. **until not** $\sim_C$ es un refinamiento extricto de $\sim_{C^{old}}$

9. **return** C

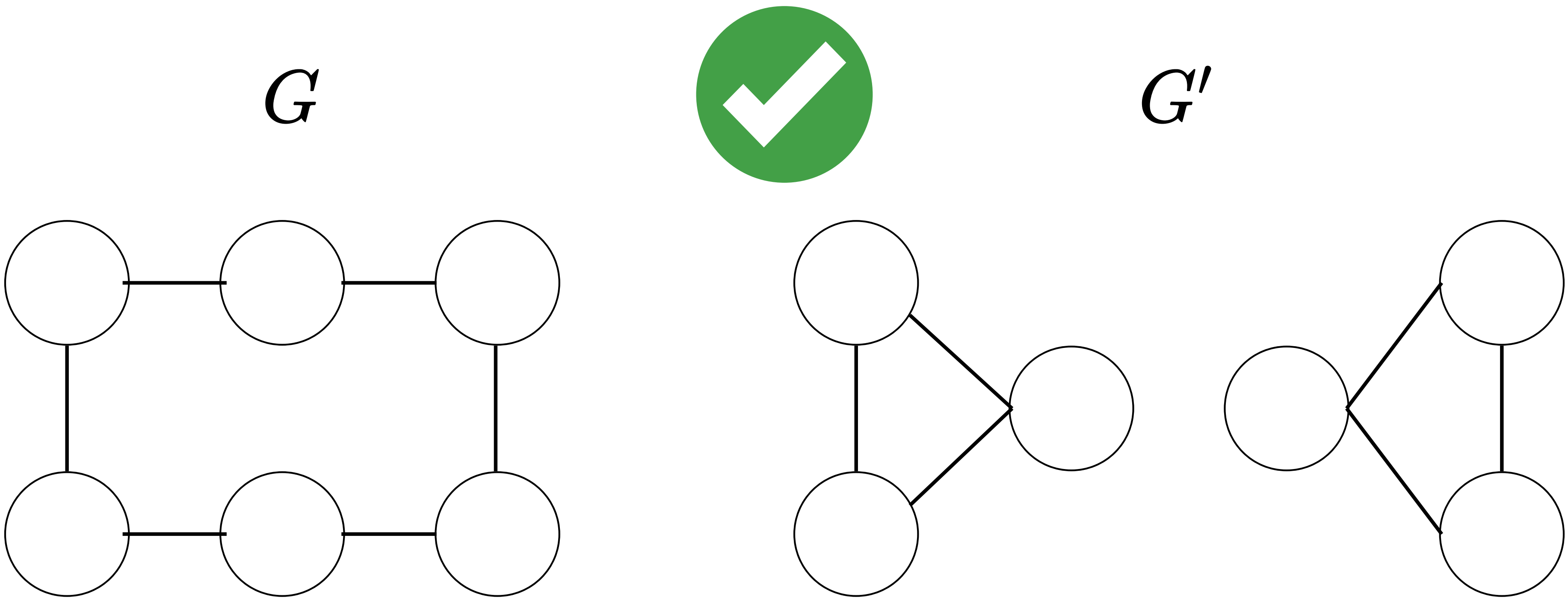
$$N_i(\overline{v}) = \{ (v_1, ..., v_{i-1}, m, v_{i+1}, ..., v_k) \mid m \in V \}$$

Test de isomorfismo 3-WL



¿Qué resultado da la 3-WL sobre estos grafos?

Test de isomorfismo 3-WL



¿3-WL también da el resultado correcto en este caso?

Algoritmo k-folklore-WL
(k-WL especial)

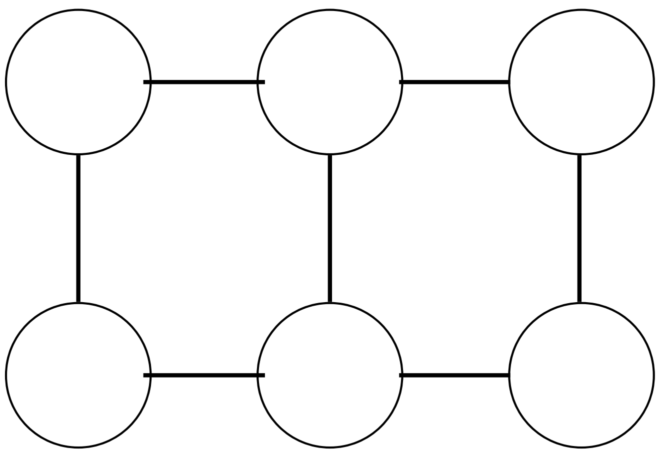
Algoritmo k-folklore-WL

Input: $G = (V, E)$

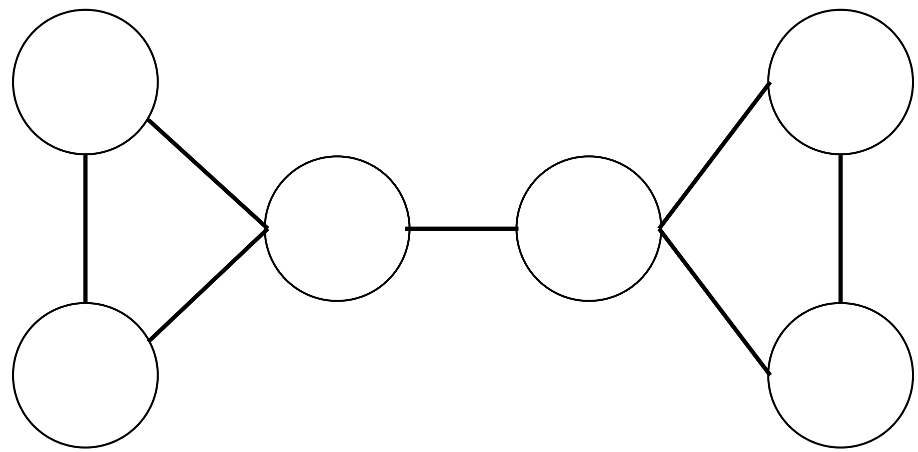
1. $C \leftarrow$ **ColoreoInicial**(G)
2. **repeat**
3. $\quad C^{old} \leftarrow C$
4. $\quad$ **for each** $(v_1, ..., v_k) \in V^k$ **do**
5. $\quad\quad$ **for each** $w \in V$ **do**
6. $\quad\quad\quad$ **for** $j = 1$ **to** k **do**
7. $\quad\quad\quad\quad \Delta_w \leftarrow [C^{old}(w, v_2, \ldots, v_k), \ldots, C^{old}(v_1, \ldots, v_{k-1}, w)]$
8. $\quad\quad C[\overline{v}] \leftarrow \text{hash}\Big(C^{old}[\overline{v}], \{\!\!\{\Delta_w \mid w \in V\}\!\!\}\Big)$
9. **until not** $\sim_C$ es un refinamiento extricto de $\sim_{C^{old}}$
10. **return** C

Test de isomorfismo 2-folklore-WL

G

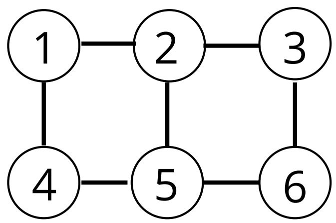


G'

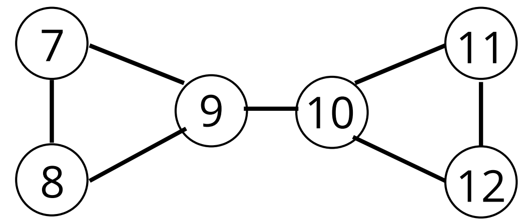


ColoreoInicial(G)

G



G'

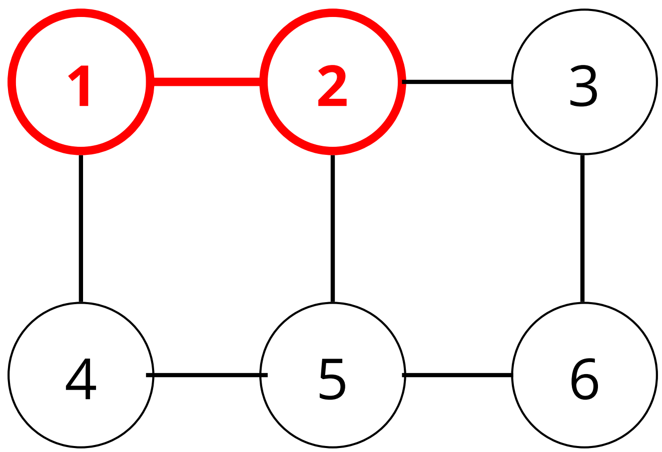


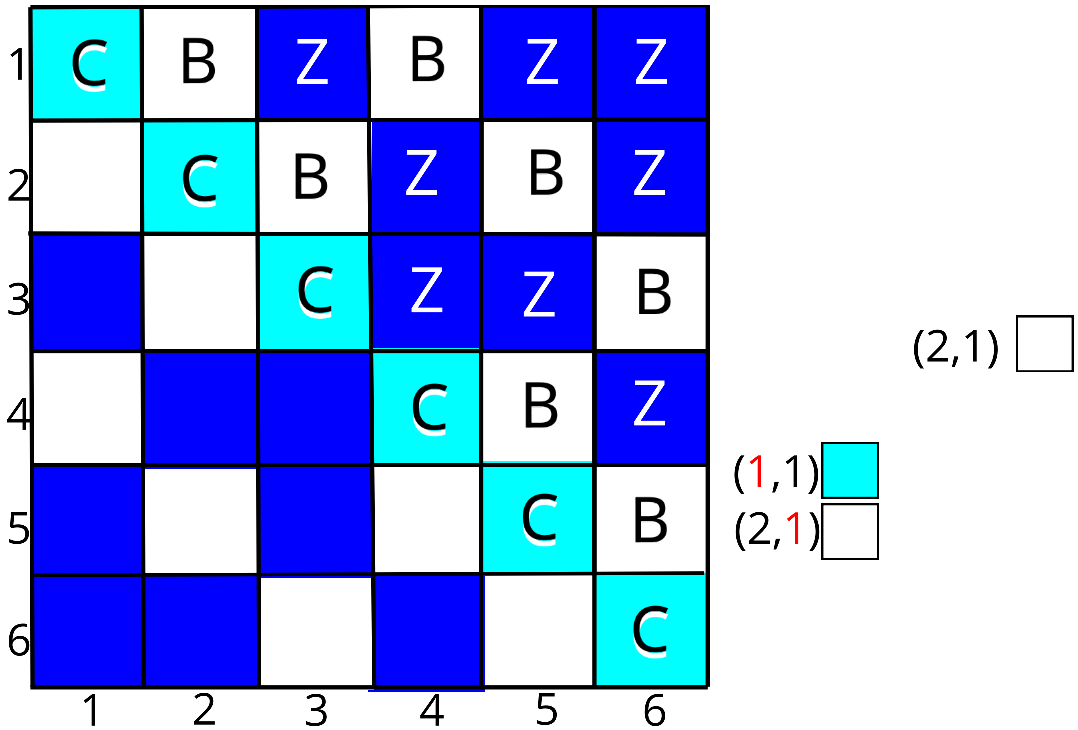
	1	2	3	4	5	6
1	C	B	Z	B	Z	Z
2		C	B	Z	B	Z
3			C	Z	Z	B
4				C	B	Z
5					C	B
6						C

	7	8	9	10	11	12
7	C	B	B	Z	Z	Z
8		C	B	Z	Z	Z
9			C	B	Z	Z
10				C	B	B
11					C	B
12						C

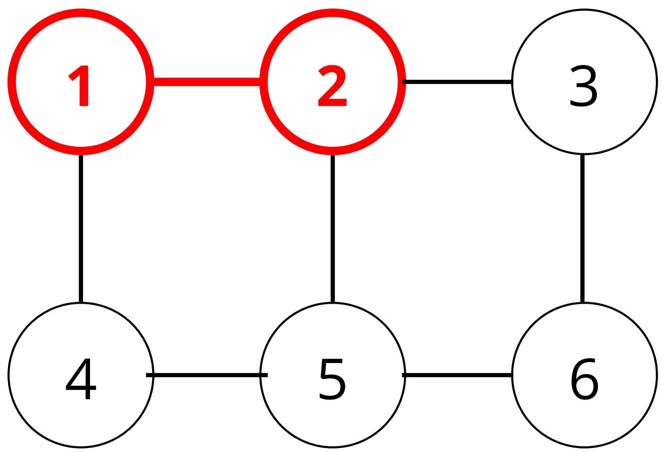
1	C	B	Z	B	Z	Z
2		C	B	Z	B	Z
3			C	Z	Z	B
4				C	B	Z
5					C	B
6						C
	1	2	3	4	5	6

(2,1) $\square$

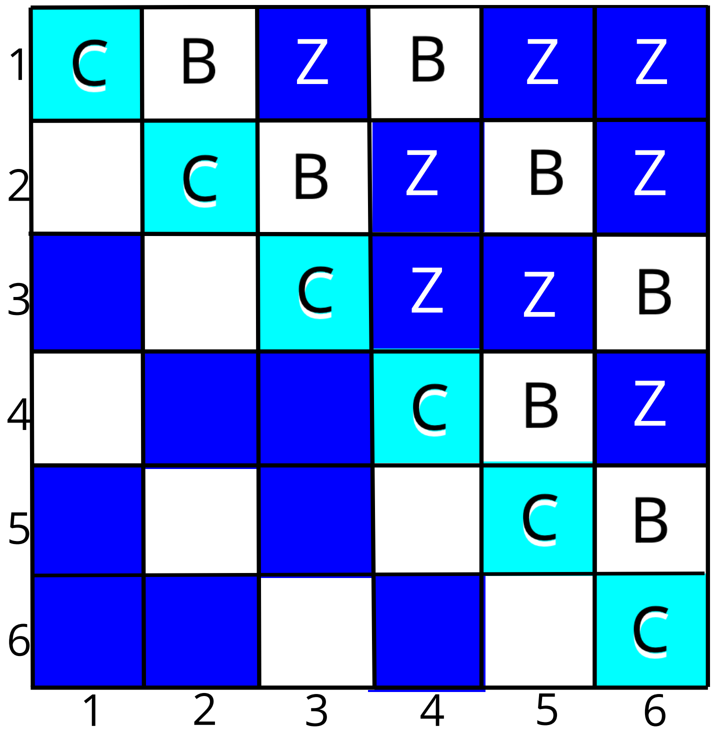




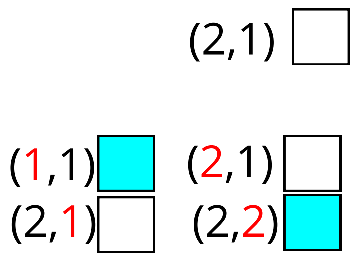
1 C B Z B Z Z
2 C B Z B Z
3 C Z Z B
4 C B Z
5 C B
6 C
1 2 3 4 5 6
(2,1)
(1,1)
(2,1)



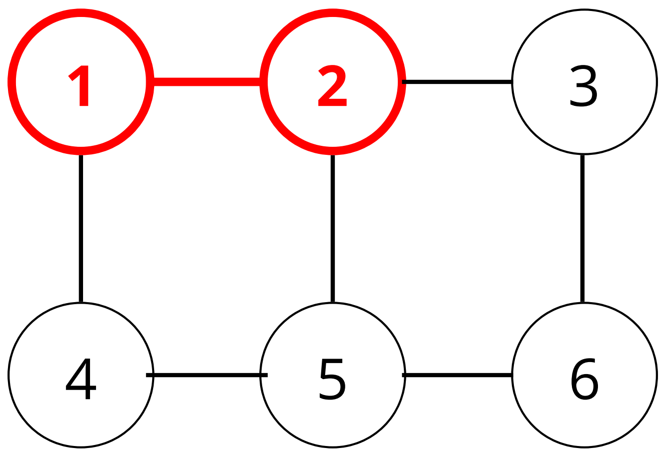
1
2
3
4
5
6



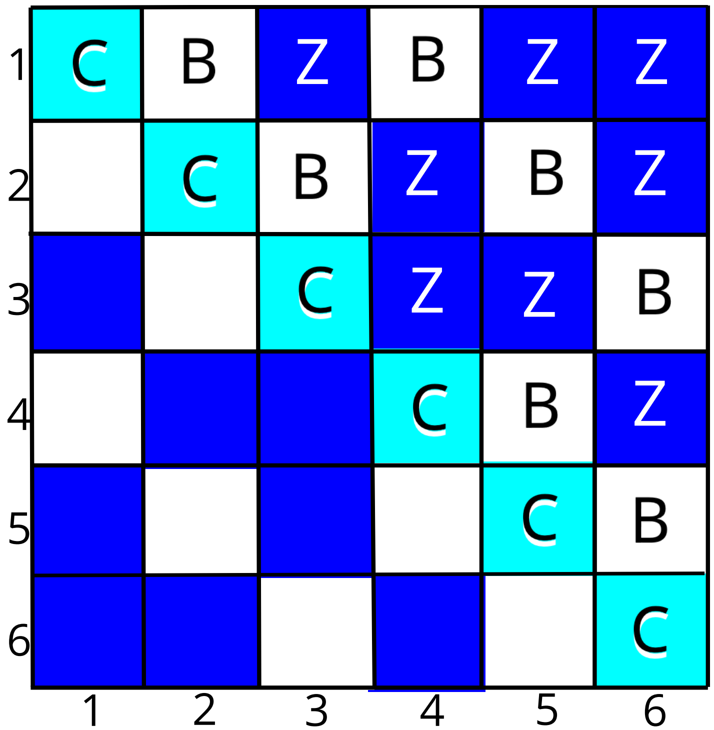
1 C B Z B Z Z
2 C B Z B Z
3 C Z Z B
4 C B Z
5 C B
6 C
1 2 3 4 5 6

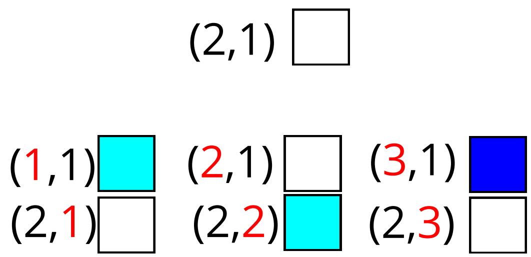


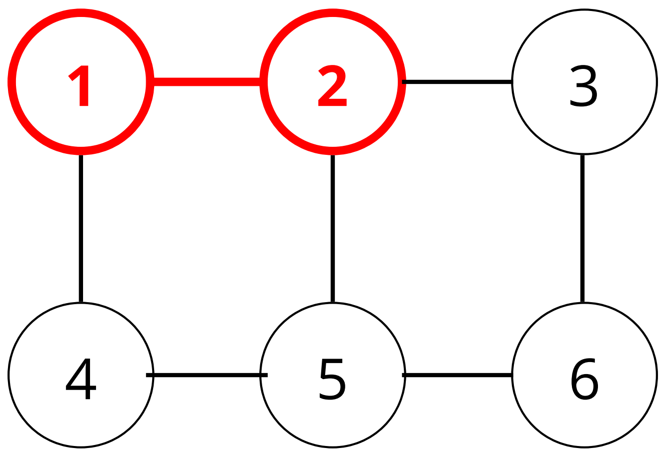
(2,1)
(1,1)
(2,1)
(2,1)
(2,2)

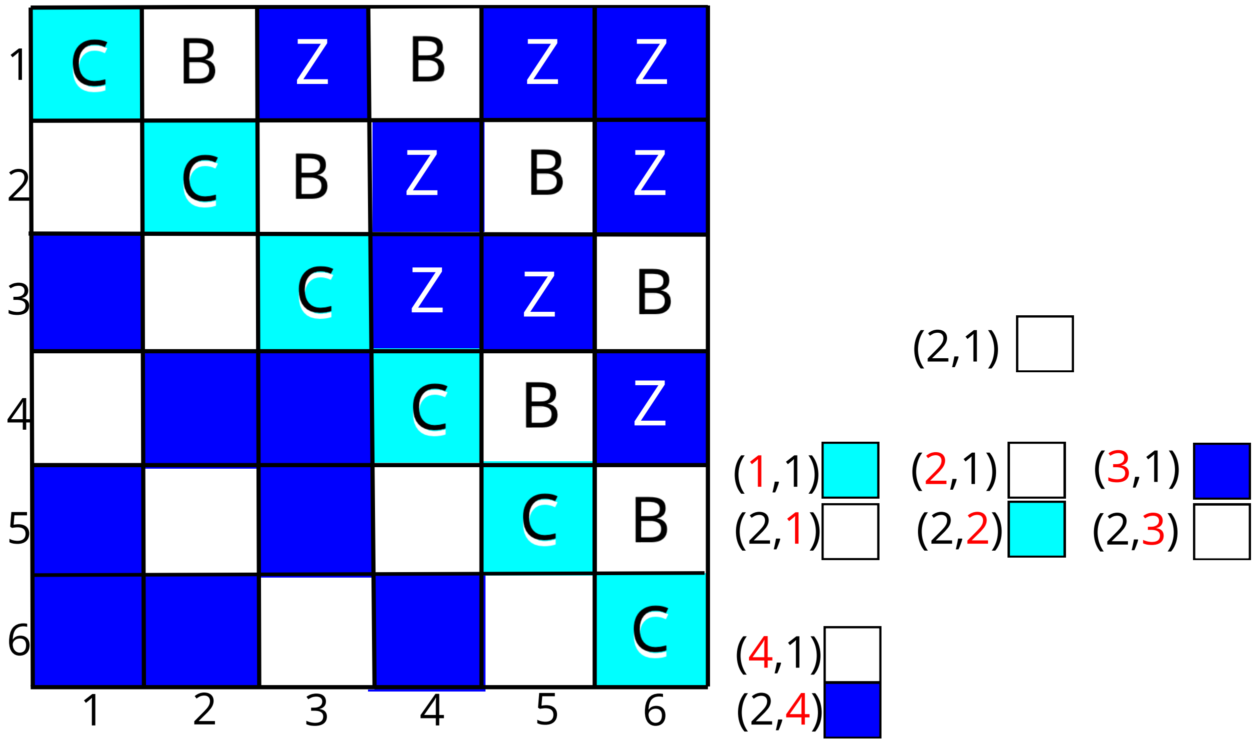


1
2
3
4
5
6

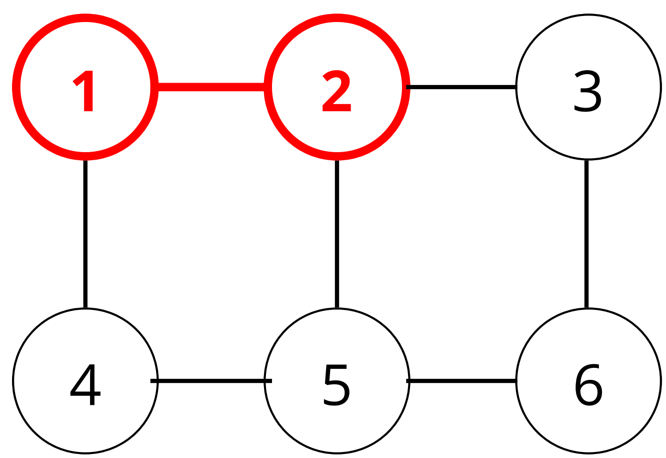

1 C B Z B Z Z
2 C B Z B Z
3 C Z Z B
4 C B Z
5 C B
6 C
1 2 3 4 5 6


(2,1)
(1,1) (2,1) (3,1)
(2,1) (2,2) (2,3)

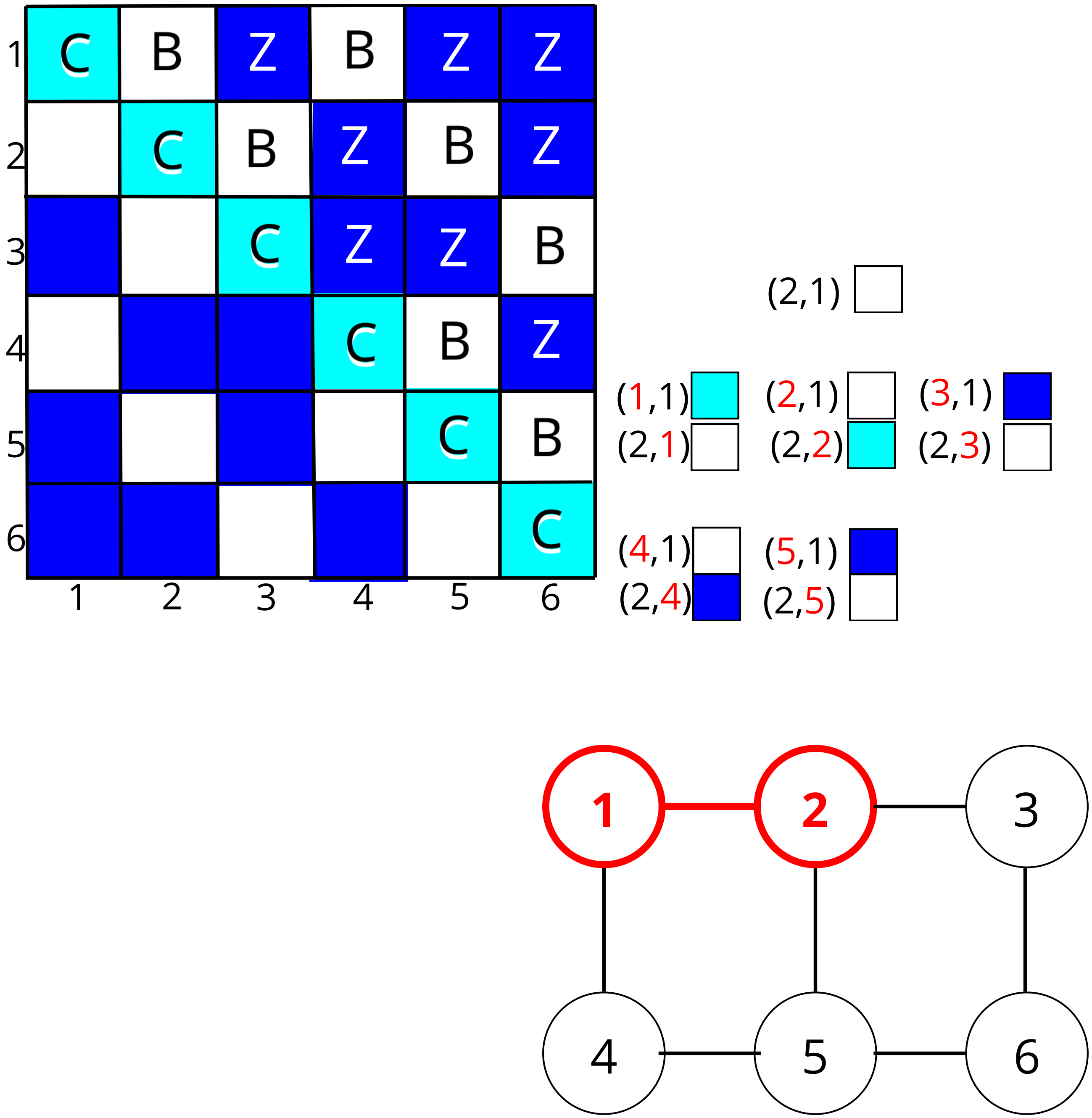

1 2 3
4 5 6



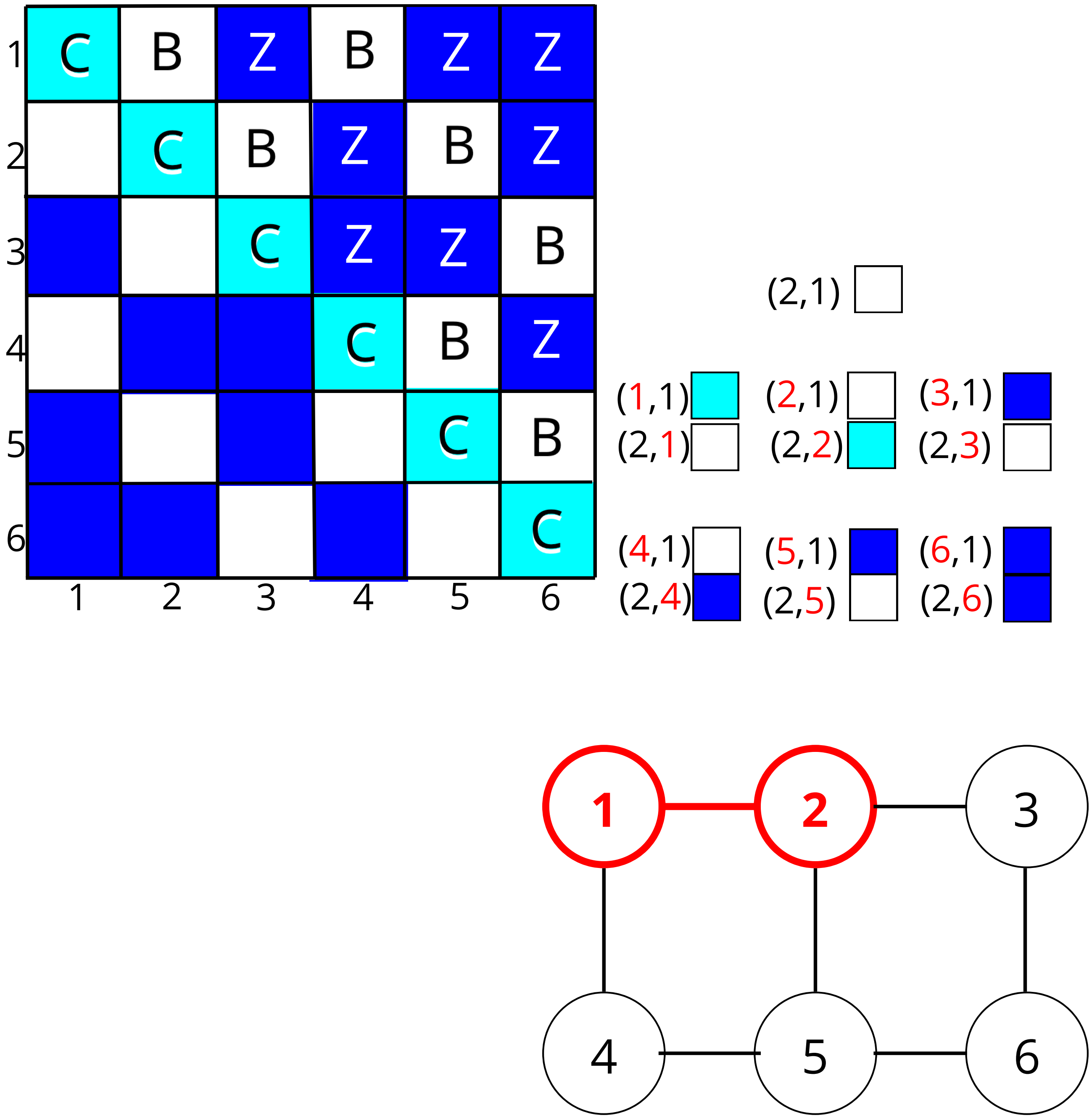
C B Z B Z Z
C B Z B Z
C Z Z B
C B Z
C B
C
1 2 3 4 5 6
(2,1)
(1,1) (2,1) (3,1)
(2,1) (2,2) (2,3)
(4,1)
(2,4)



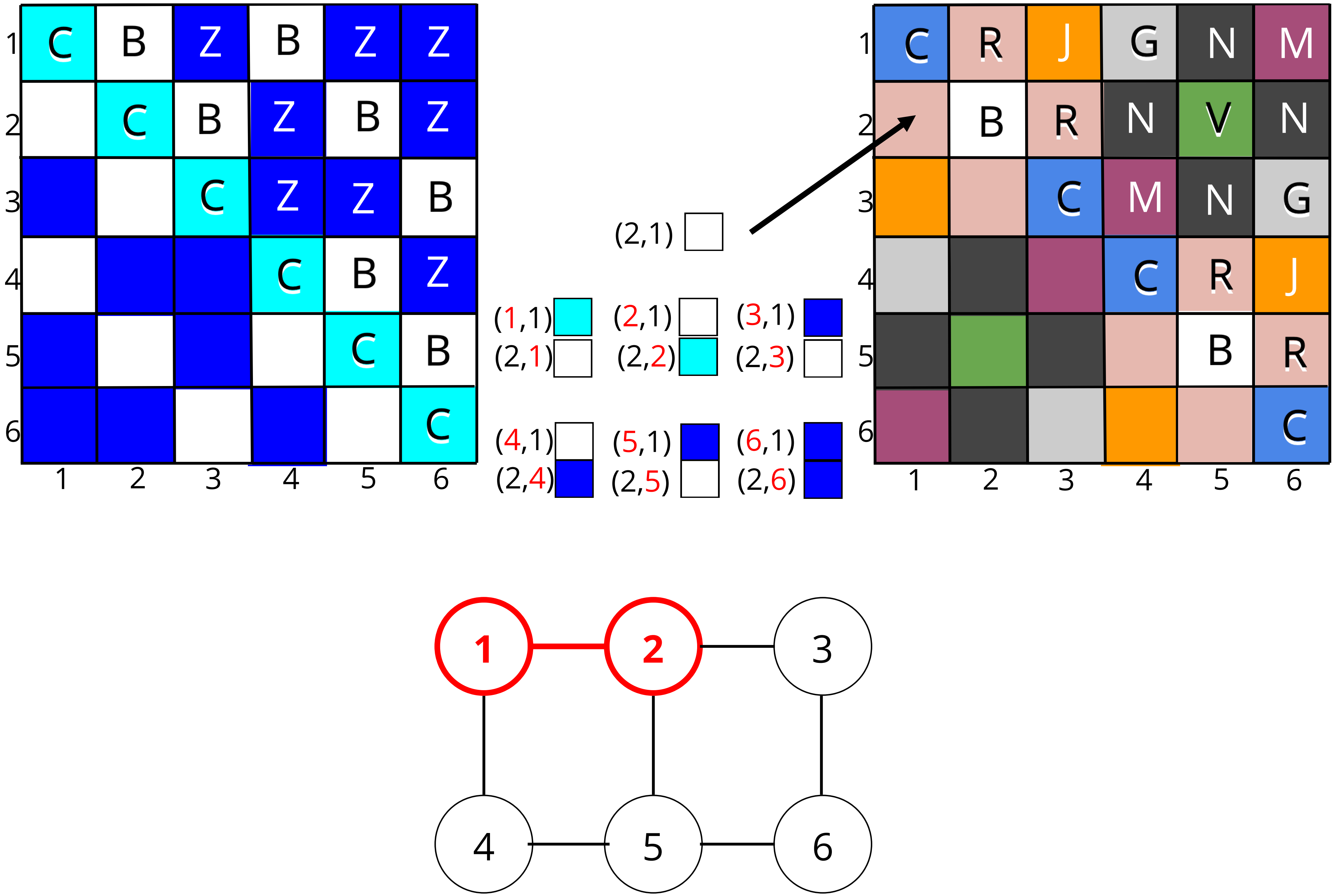
1
2
3
4
5
6



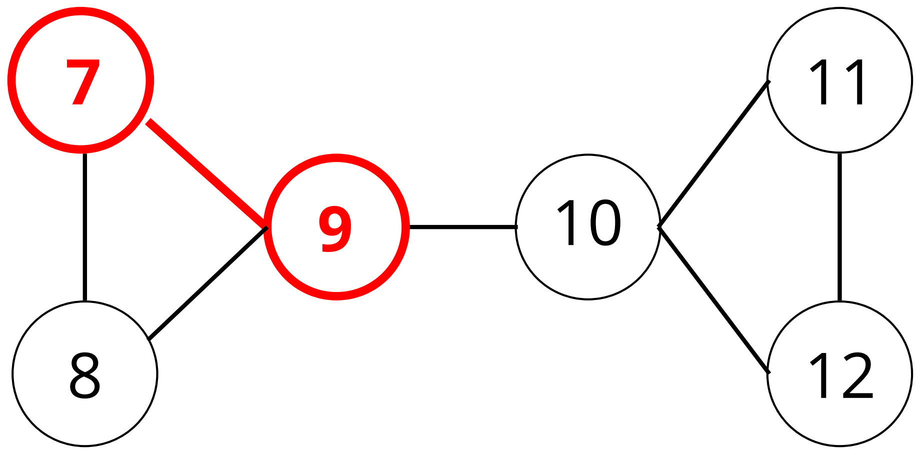
C B Z B Z Z
C B Z B Z
C Z Z B
C B Z
C B
C
1 2 3 4 5 6
(2,1)
(1,1) (2,1) (3,1)
(2,1) (2,2) (2,3)
(4,1) (5,1)
(2,4) (2,5)
1 2 3
4 5 6

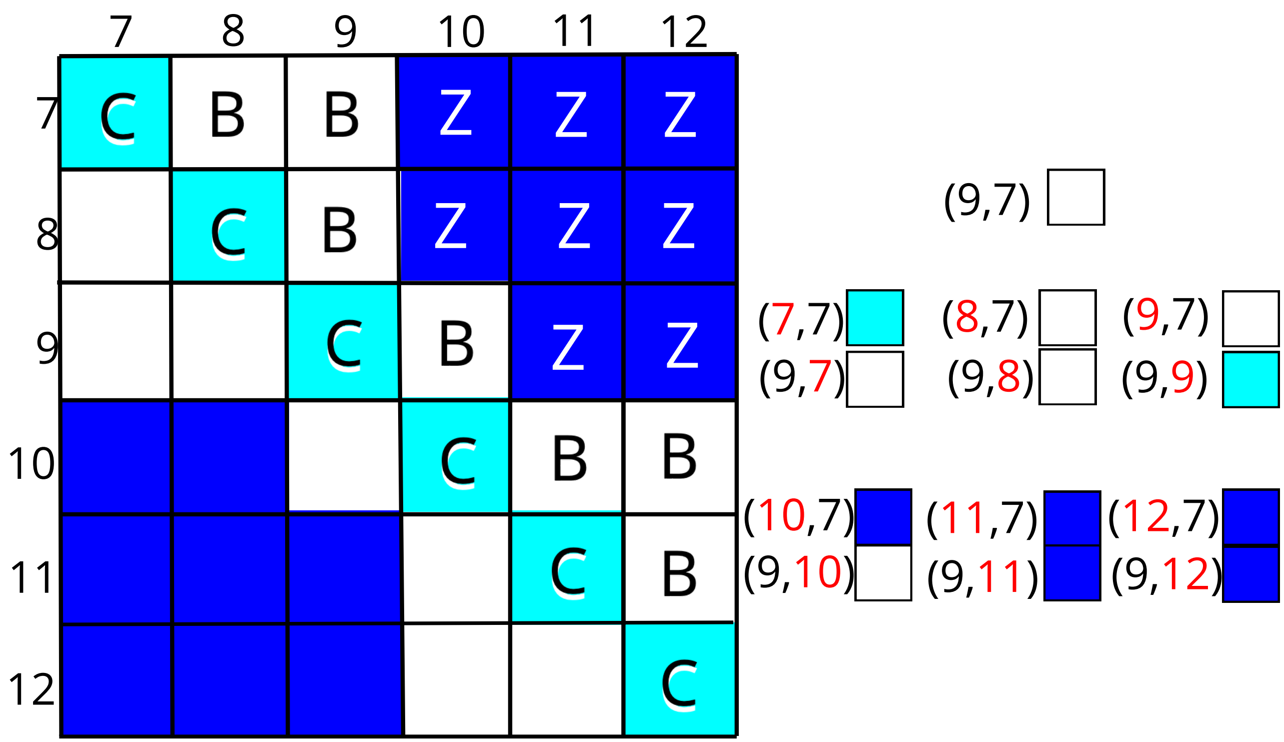


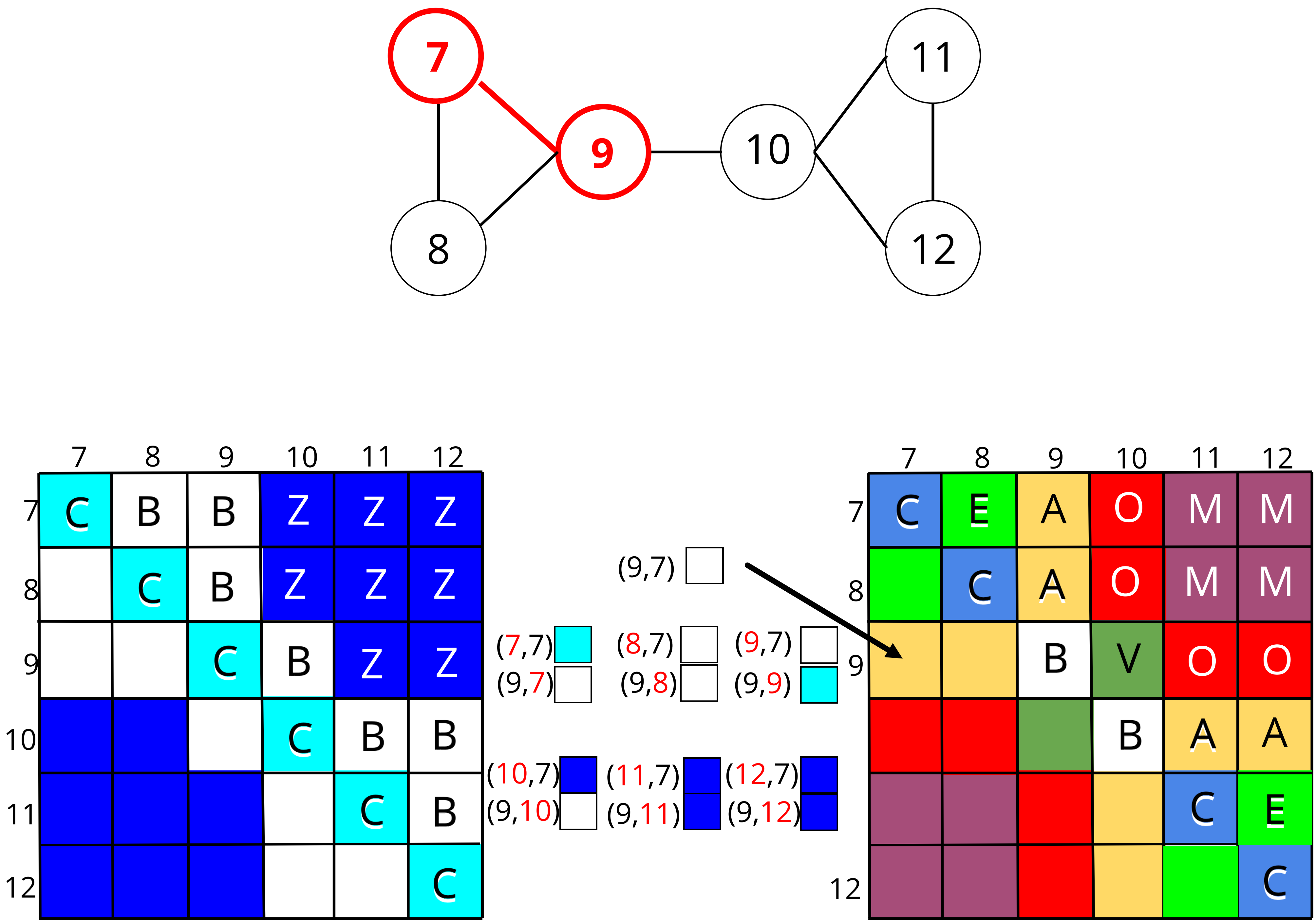
C B Z B Z Z
C B Z B Z
C Z Z B
C B Z
C B
C
1 2 3 4 5 6
(2,1)
(1,1) (2,1) (3,1)
(2,1) (2,2) (2,3)
(4,1) (5,1) (6,1)
(2,4) (2,5) (2,6)
1 2 3
4 5 6

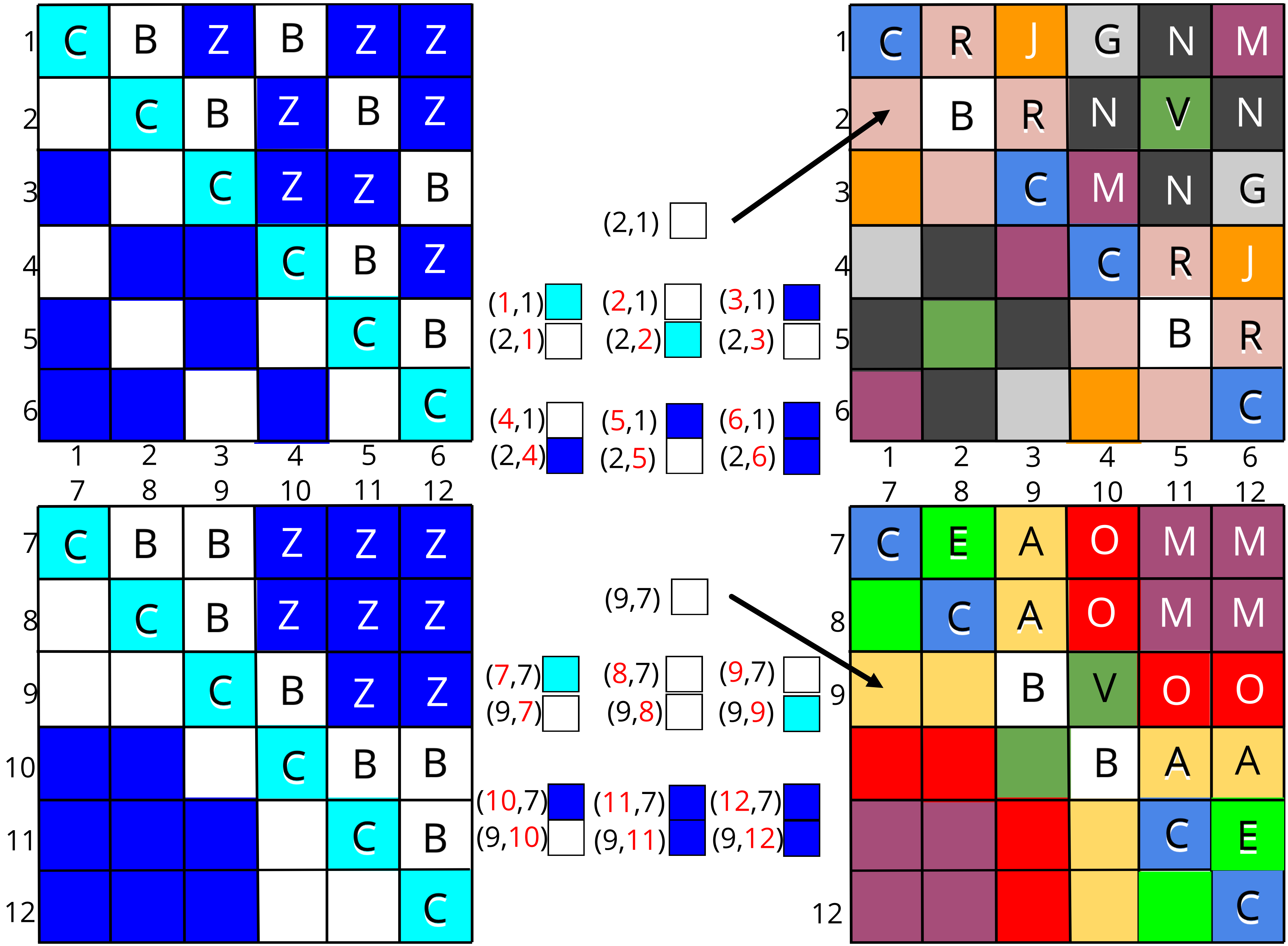


C B Z B Z Z
C B Z B Z
C Z Z B
C B Z
C B
C
(2,1)
(1,1) (2,1) (3,1)
(2,1) (2,2) (2,3)
(4,1) (5,1) (6,1)
(2,4) (2,5) (2,6)
C R J G N M
B R N V N
C M N G
C R J
B R
C
1
2
3
4
5
6

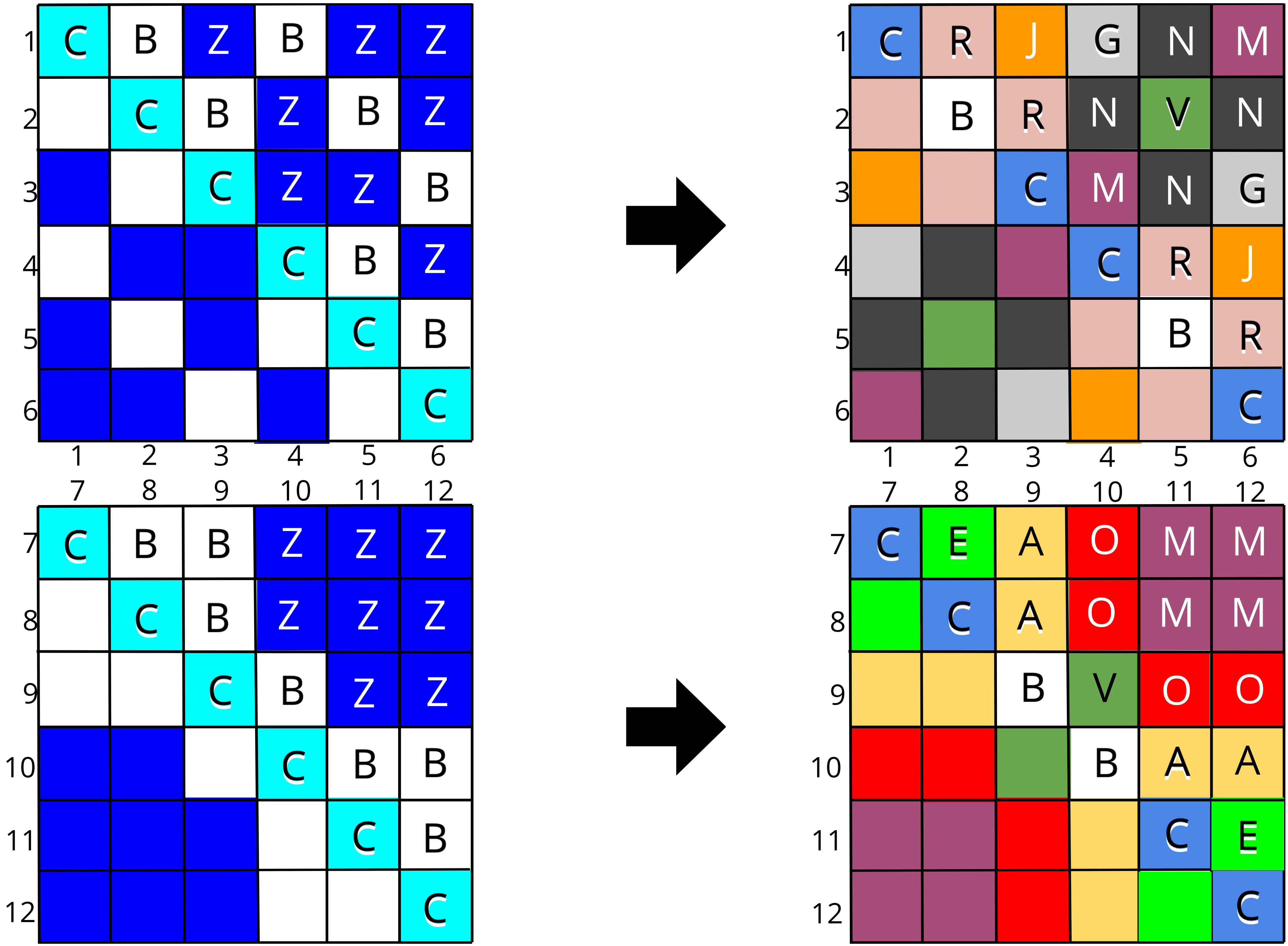

7
8
9
10
11
12

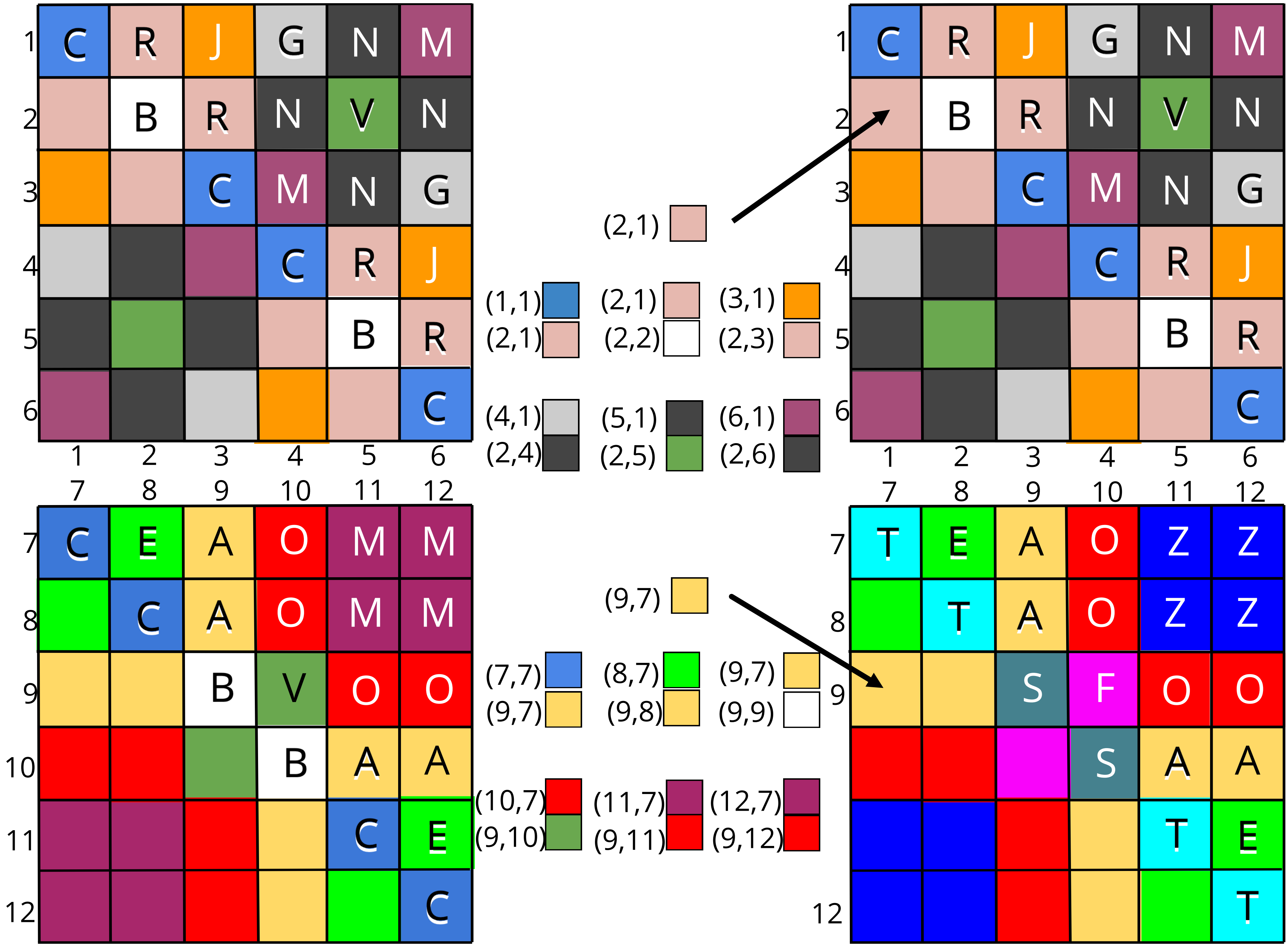

7 8 9 10 11 12
7 C B B Z Z Z
8 C B Z Z Z
9 C B Z Z
10 C B B
11 C B
12 C
(9,7)
(7,7) (8,7) (9,7)
(9,7) (9,8) (9,9)
(10,7) (11,7) (12,7)
(9,10) (9,11) (9,12)


7
9
8
10
11
12
7 8 9 10 11 12
C B B Z Z Z
C B Z Z Z
C B Z Z
C B B
C B
C
(9,7)
(7,7) (8,7) (9,7)
(9,7) (9,8) (9,9)
(10,7) (11,7) (12,7)
(9,10) (9,11) (9,12)
C E A O M M
C A O M M
B V O O
B A A
C E
C

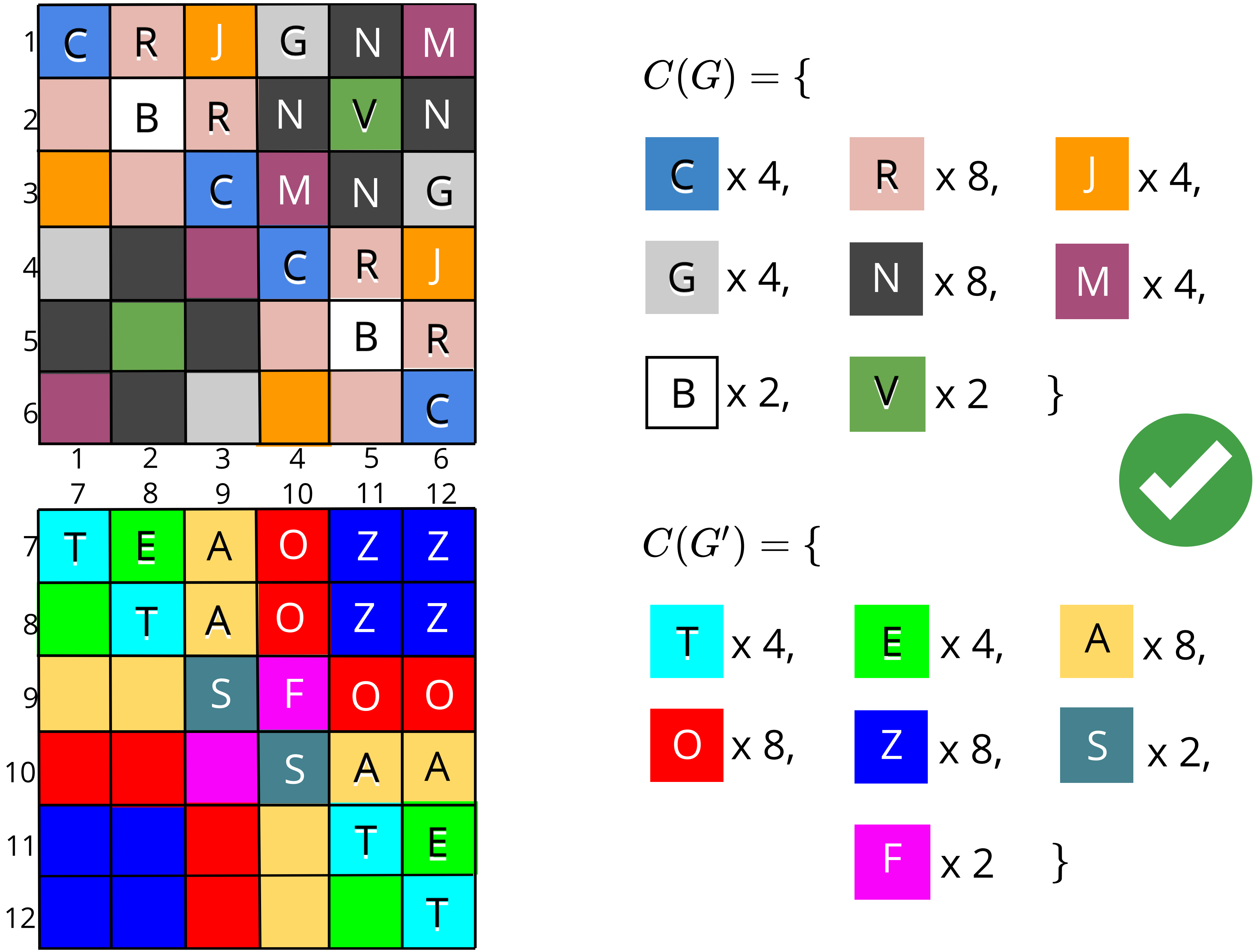


(2,1)
(1,1) (2,1) (3,1)
(2,1) (2,2) (2,3)
(4,1) (5,1) (6,1)
(2,4) (2,5) (2,6)
(9,7)
(7,7) (8,7) (9,7)
(9,7) (9,8) (9,9)
(10,7) (11,7) (12,7)
(9,10) (9,11) (9,12)


1 C B Z B Z Z
2 C B Z B Z
3 C Z Z B
4 C B Z
5 C B
6 C
1 C R J G N M
2 B R N V N
3 C M N G
4 C R J
5 B R
6 C
1 2 3 4 5 6
7 8 9 10 11 12
7 C B B Z Z Z
8 C B Z Z Z
9 C B Z Z
10 C B B
11 C B
12 C
7 C E A O M M
8 C A O M M
9 B V O O
10 B A A
11 C E
12 C



C R J G N M
B R N V N
C M N G
C R J
B R
C
C E A O M M
C A O M M
B V O O
B A A
C E
C
(2,1)
(1,1) (2,1) (3,1)
(2,1) (2,2) (2,3)
(4,1) (5,1) (6,1)
(2,4) (2,5) (2,6)
(9,7)
(7,7) (8,7) (9,7)
(9,7) (9,8) (9,9)
(10,7) (11,7) (12,7)
(9,10) (9,11) (9,12)
T E A O Z Z
T A O Z Z
S F O O
S A A
T E
T



$C(G) = \{$
C x 4, R x 8, J x 4,
G x 4, N x 8, M x 4,
B x 2, V x 2 $\}$
$C(G') = \{$
T x 4, E x 4, A x 8,
O x 8, Z x 8, S x 2,
F x 2 $\}$

$(k-1)$-folklore-WL es equivalente a k-WL

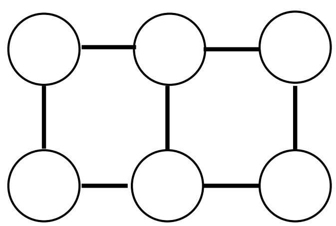
(Cai, Furer & Immerman, 1992)

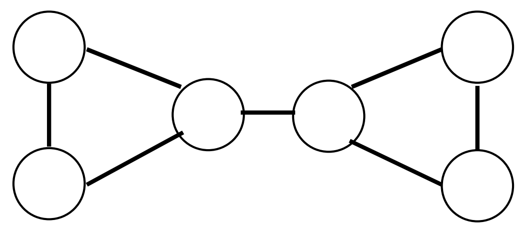
¿Dónde estamos?

- El test WL debe aplicarse a la unión disjunta de los dos grafos de entrada.
- Al incrementar k, el algoritmo k-WL tiene un mayor poder de distinguir dos grafos no-isomorfos.
- 1-WL es equivalente a 2-WL.
- $(k-1)$-folklore-WL es equivalente a k-WL.

¿Cuál es el k necesario?

- No hay un valor fijo k tal que k-WL decide correctamente si dos grafos son isomorfos.
- 4-WL decide correctamente si dos grafos planos son isomorfos (Kiefer, Ponomarenko, Schweitzer, 2019)





¿Se puede extender este resultado a otras clases de grafos?