

بسم الله الرحمن الرحيم

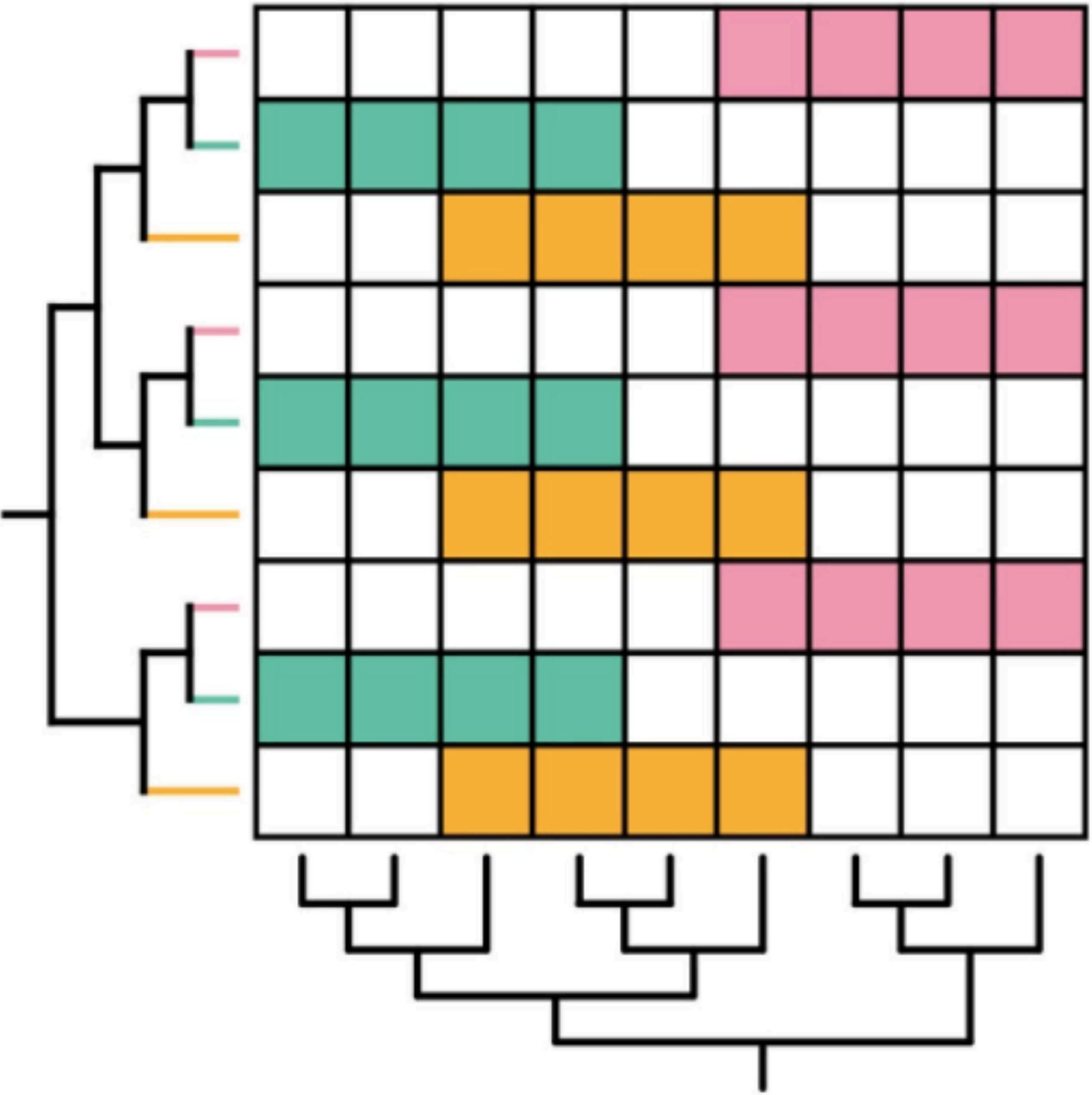
ڙنو ميڪ محاسباتي

جلسه ٧: کار با چند درخت

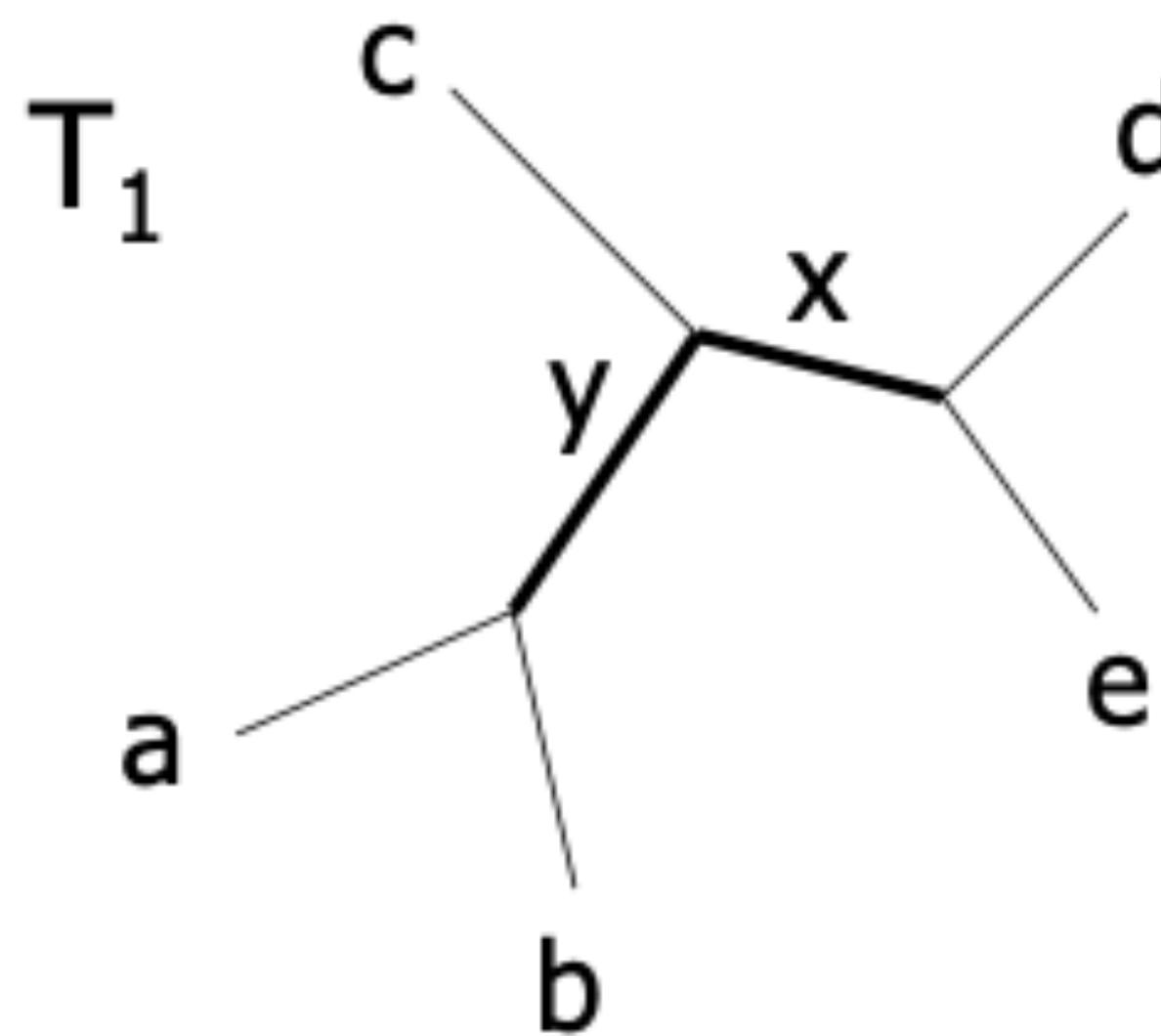
١٤٠١-١٤٠٠
ترم پايز



مقایسه درخت‌ها



(Split) تقسيم

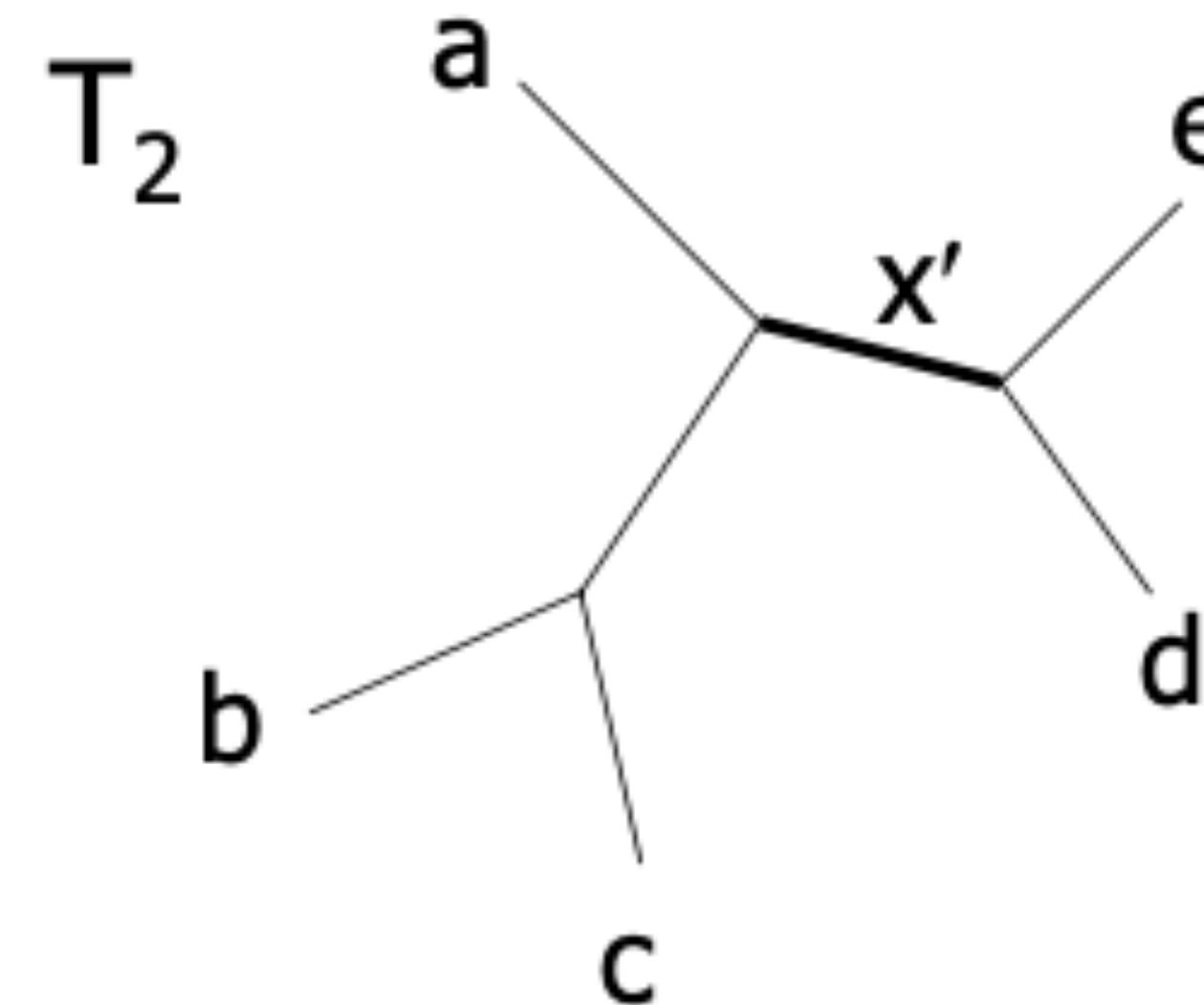
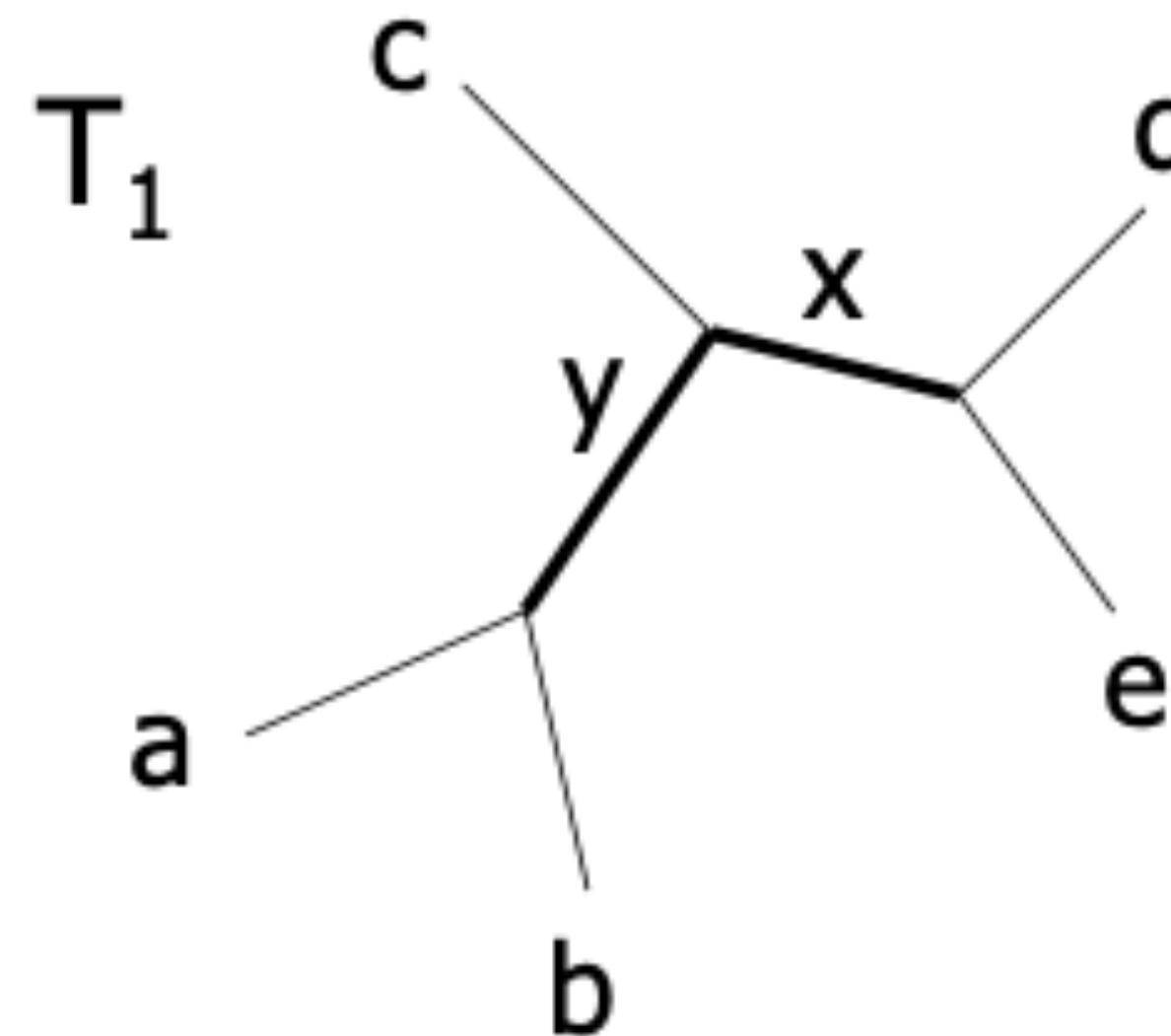


abc | ed

ab | cde

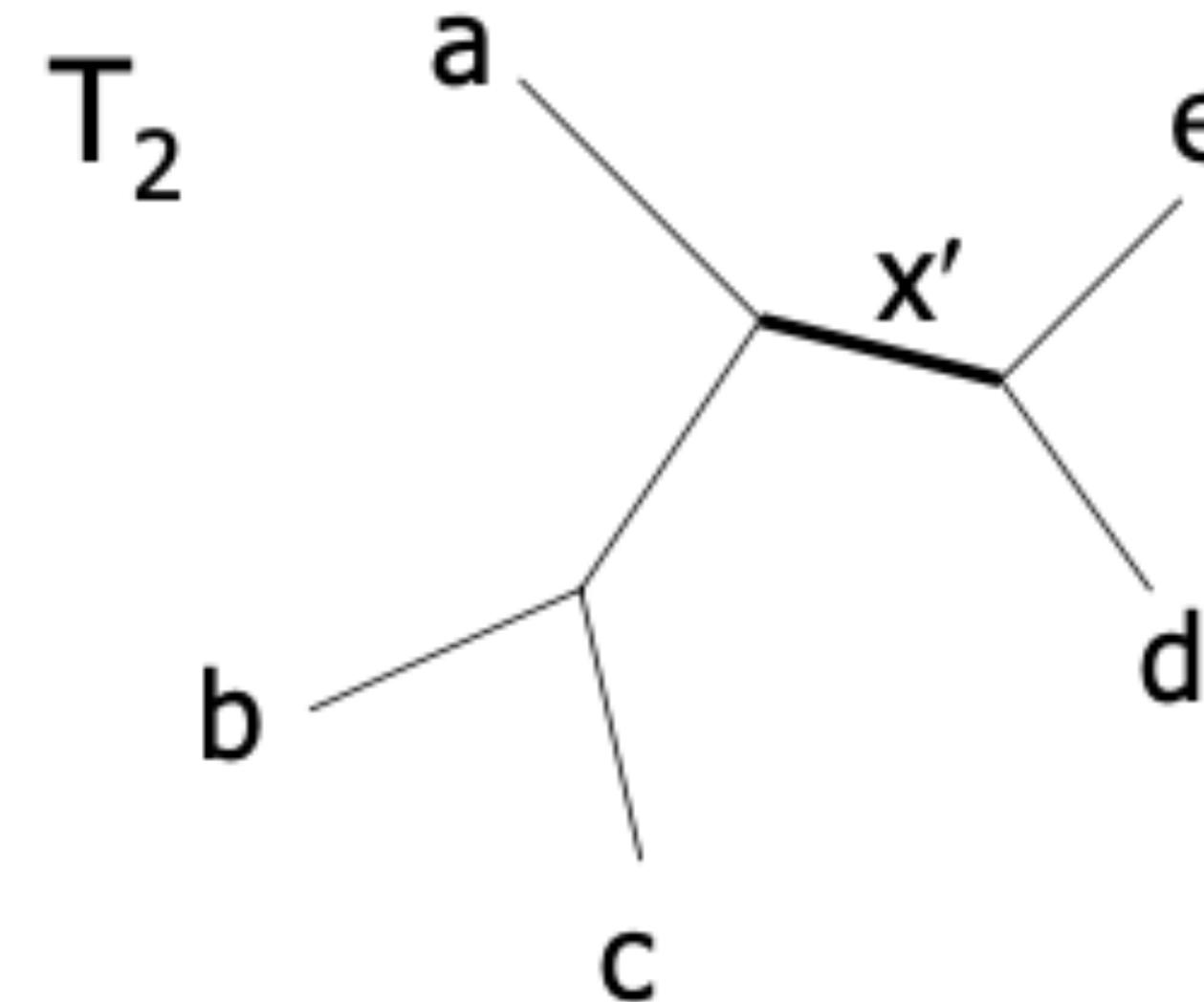
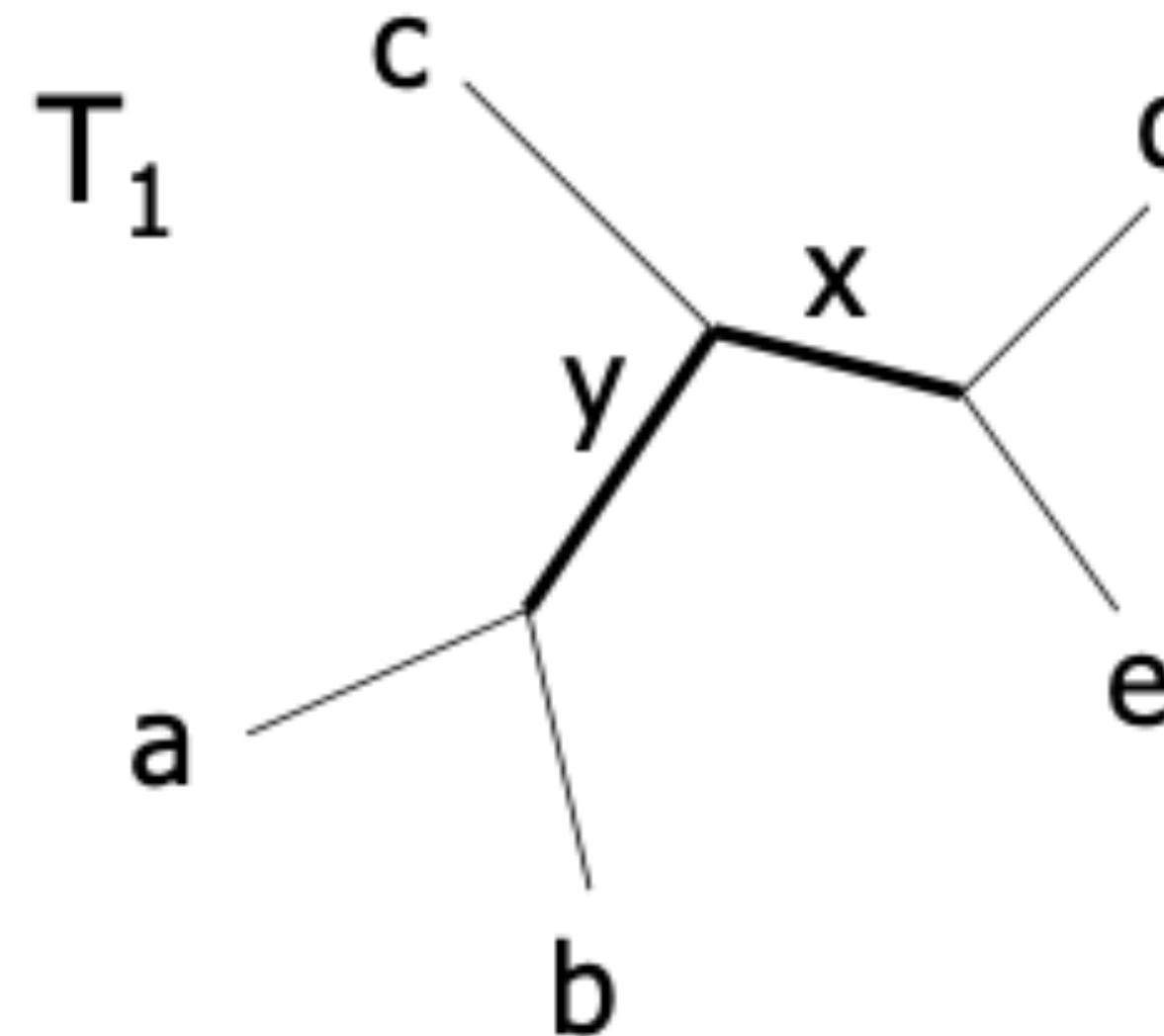
فاصله Robinson-Foulds

- تعداد تقسیم‌های نامساوی در هر دو تقسیم بر ۲



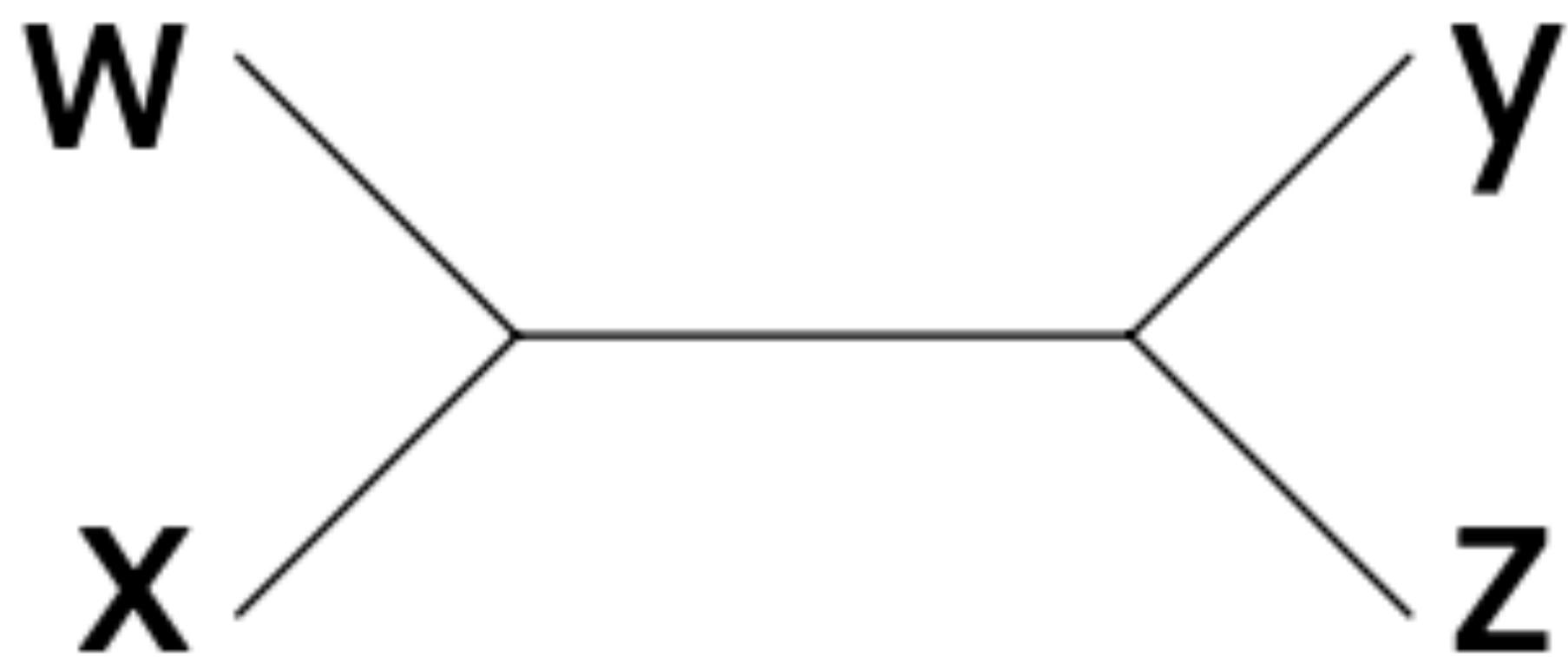
فاصله Robinson-Foulds

- تعداد تقسیم‌های نامساوی در هر دو تقسیم بر ۲

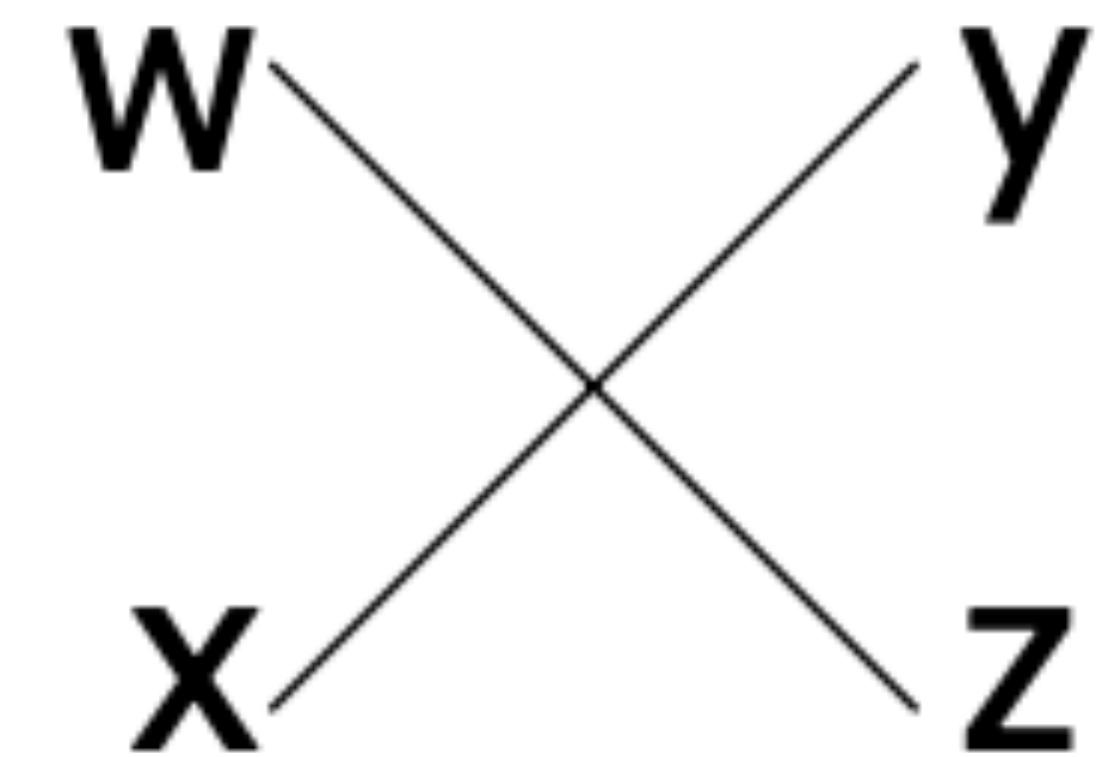


$$RF(T_1, T_2) = ?$$

فاصله چهارتایی‌ها

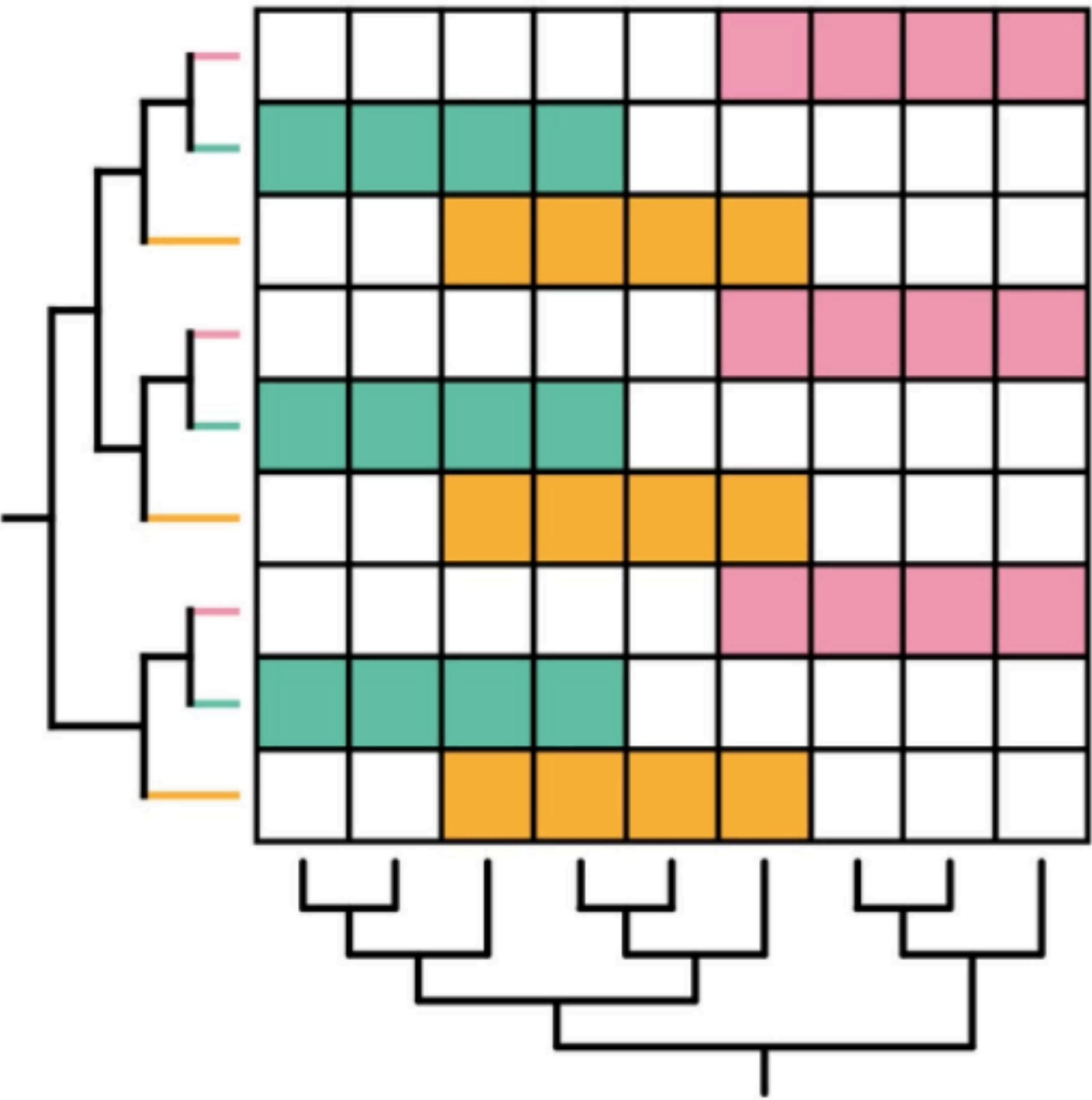


Butterfly quartet



Star quartet

درخت توانفچی



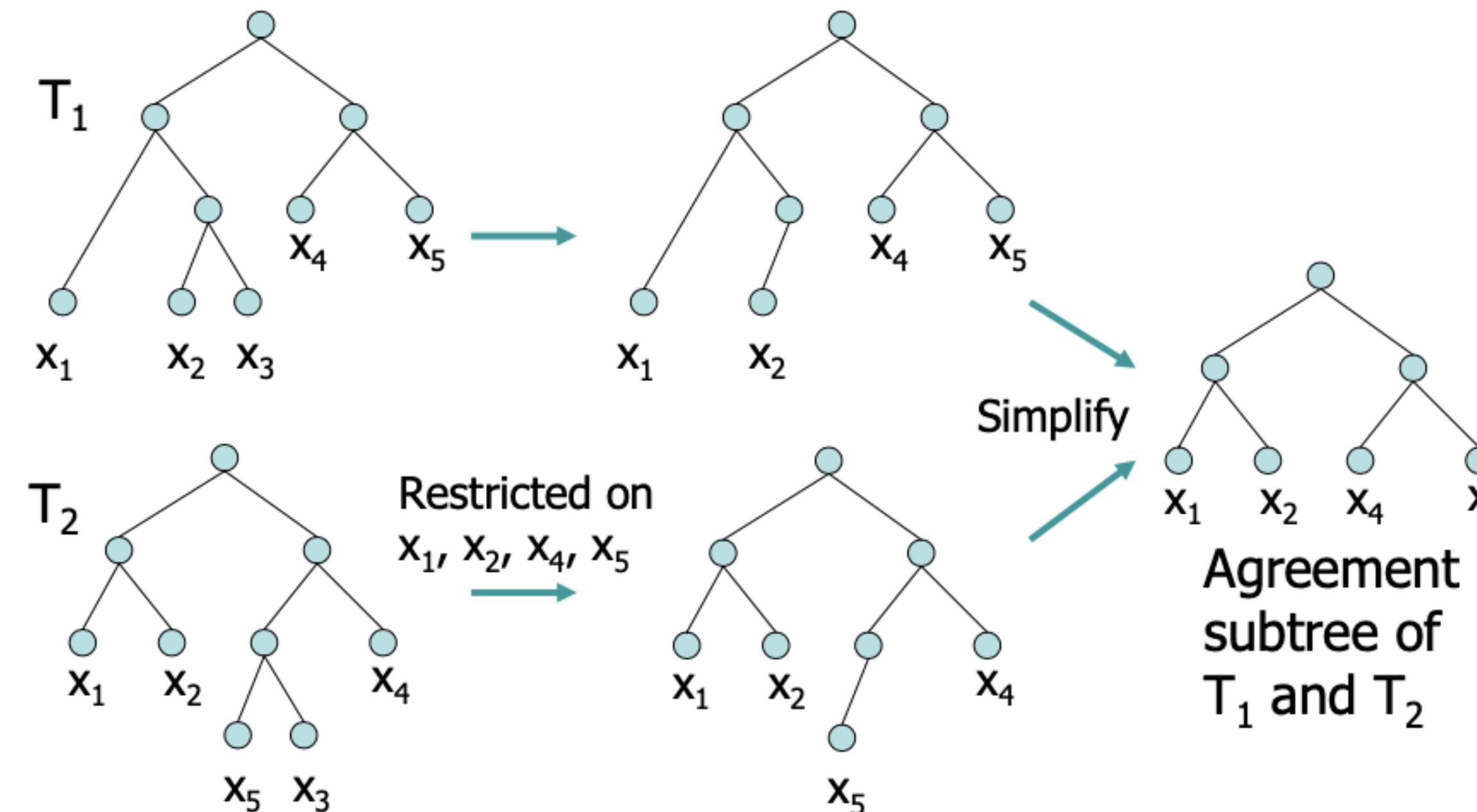
بزرگ‌ترین زیردرخت توافقی

● درخت توافقی: با حذف گونه‌ها، و ساده کردن درخت مساوی شوند.

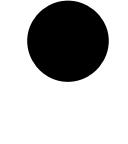


بزرگ‌ترین زیردرخت توافقی

- درخت توافقی: با حذف گونه‌ها، و ساده کردن درخت مساوی شوند.



الگوریتم بزرگ‌ترین زیردرخت توافقی

: بیشترین امتیاز از یکی کردن زیر درخت نوادگان v از T_1 و زیردرخت نوادگان u از T_2 

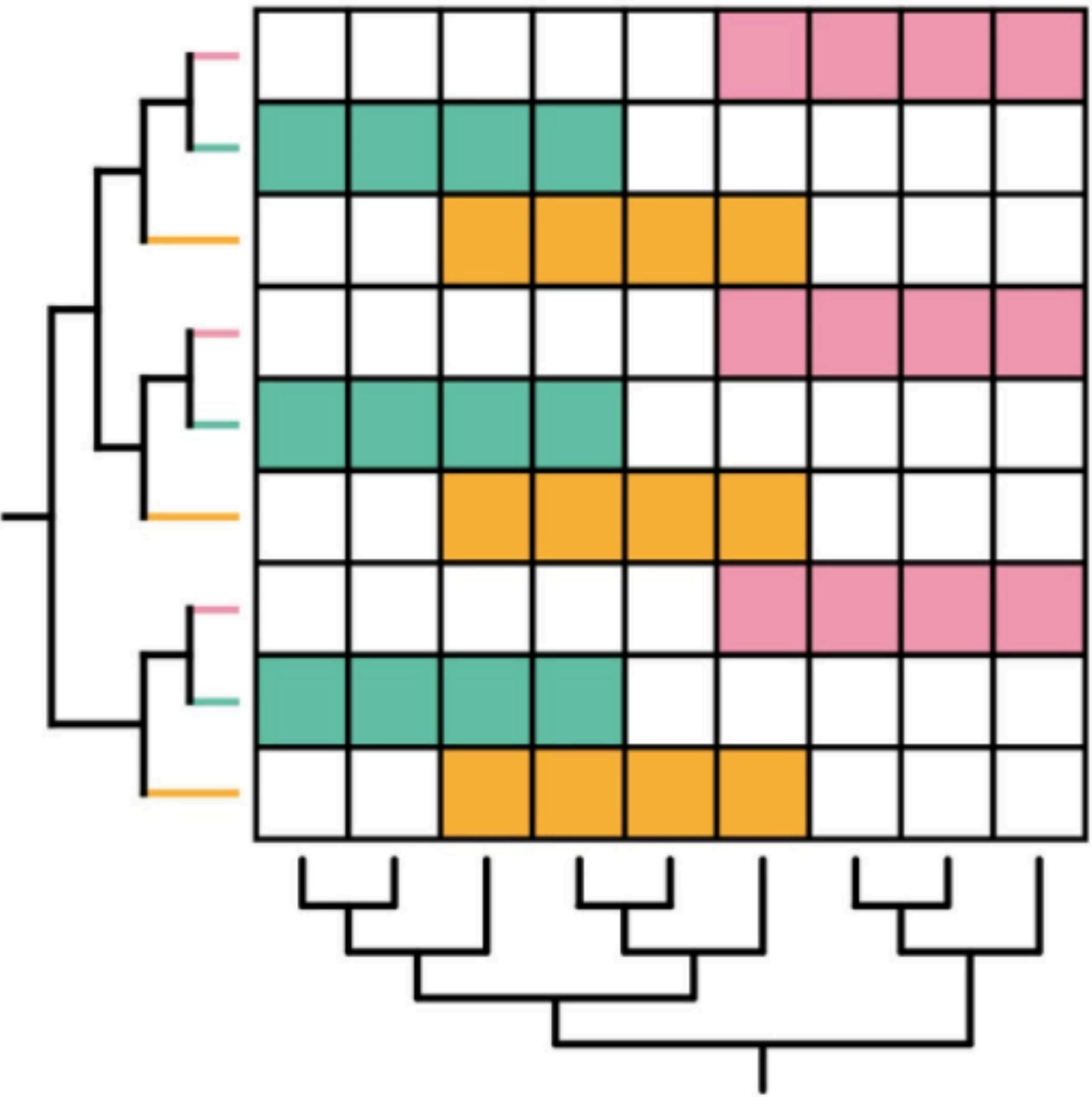
الگوریتم بزرگ‌ترین زیردرخت توافقی

: بیشترین امتیاز از یکی کردن زیر درخت نوادگان v از T1 و زیردرخت نوادگان u از T2 $A[v, u]$ ●

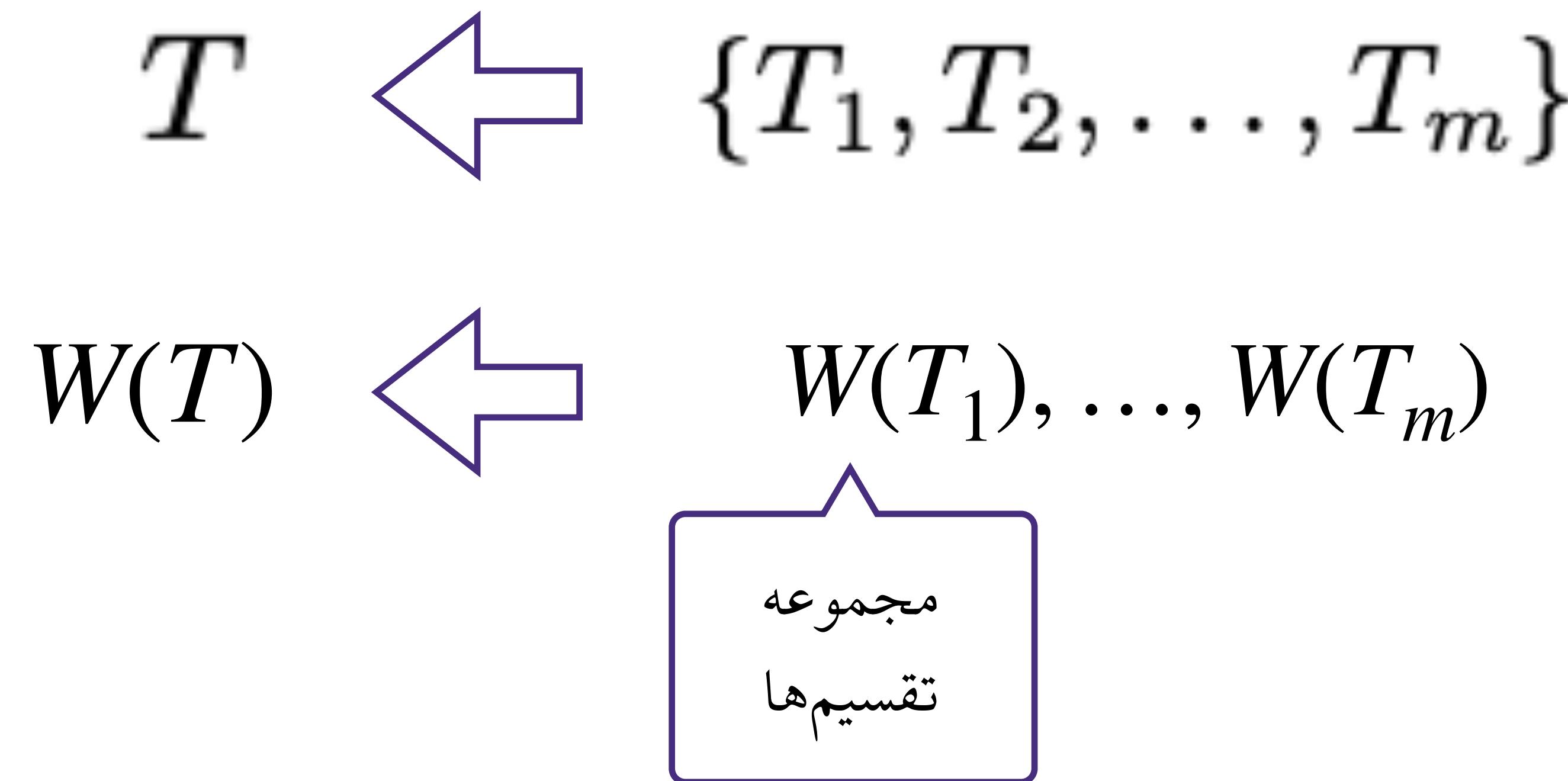
$$MAST(T_1^u, T_2^u) = \max \left\{ \begin{array}{l} MAST(T_1^{u_1}, T_2^{v_1}) + MAST(T_1^{u_2}, T_2^{v_2}) \\ MAST(T_1^{u_2}, T_2^{v_1}) + MAST(T_1^{u_1}, T_2^{v_2}) \\ MAST(T_1^{u_1}, T_2^v) \\ MAST(T_1^{u_2}, T_2^v) \\ MAST(T_1^u, T_2^{v_1}) \\ MAST(T_1^u, T_2^{v_2}) \end{array} \right.$$

درخت اجماعی

Consensus Tree



از روی تقسیم‌ها



درخت اجتماعی اکید

$$T \leftarrow \{T_1, T_2, \dots, T_m\}$$

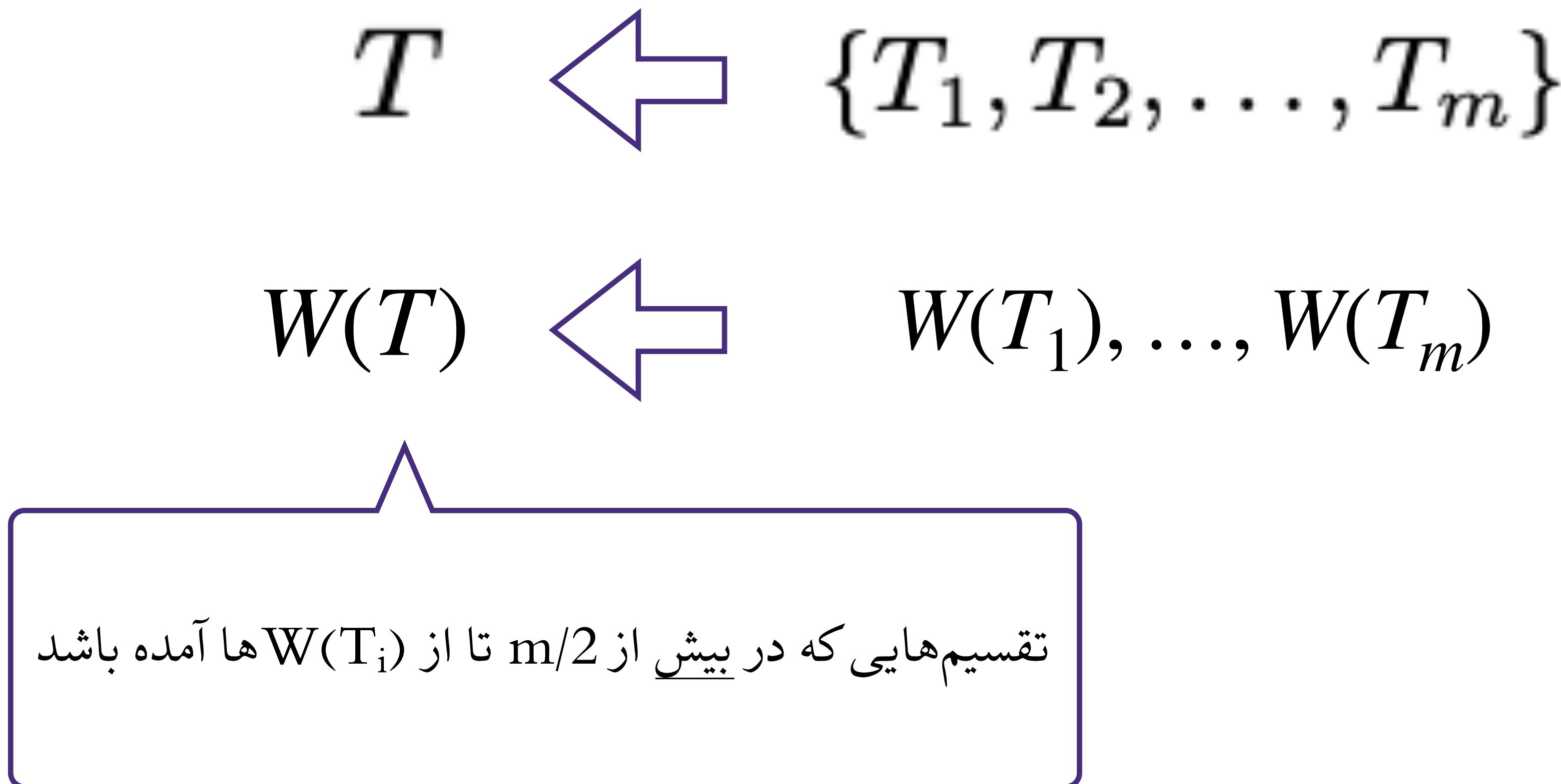
$$W(T) \leftarrow W(T_1), \dots, W(T_m)$$

$$W(T_1) \cap \dots \cap W(T_m) =$$

الگوریتم درخت اجتماعی اکید

- هر کدام از درخت‌ها را می‌توان تغییر داد که دقیقاً تقسیم‌های $= W(T_1) \cap \dots \cap W(T_m)$ را داشته باشند
-

درخت اجتماعی اکثریت



بې سوی الگوريتىم ...

بە سوی الگوريتم ...

- درخت‌های T_i ,
- ۱) ریشه روی گونه ۱
- ۲) منقبض که فقط $W(T)$ را داشته باشند
- $T'_i \leq$

بە سوی الگوريتم ...

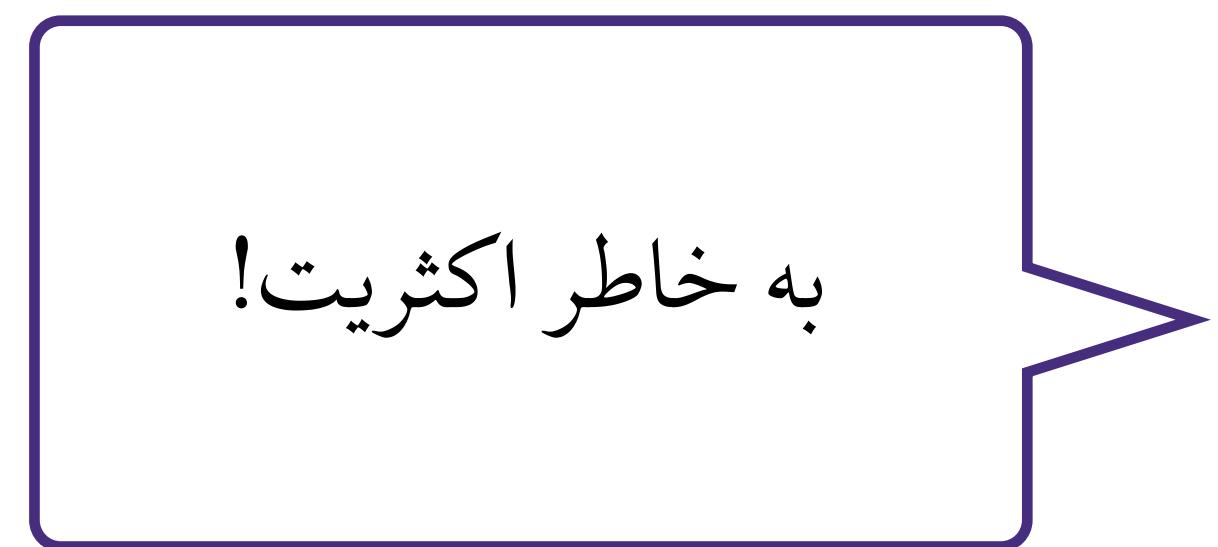
- درخت‌های T_i ,
- ۱) ریشه روی گونه ۱
- ۲) منقبض که فقط $W(T)$ را داشته باشند
- $T'_i \leq$
- بە ازای هر تقسیم c ، $B_c =$ آن طرف که شامل ۱ نیست.

بے سوی الگوریتم ...

- درخت‌های T_i ,
- ۱) ریشه روی گونه ۱
- ۲) منقبض که فقط $W(T)$ را داشته باشند
- $T'_i \leq$
- به ازای هر تقسیم C ، $B_C =$ آن طرف که شامل ۱ نیست.
- یک درخت با مجموعه تقسیم = اجتماع همه این B_C ‌ها؟!

بے سوی الگوریتم ...

- درخت‌های T_i
- ۱) ریشه روی گونه ۱
- ۲) منقبض که فقط $W(T)$ را داشته باشند
- $T'_i \leq$
- به ازای هر تقسیم c ، $B_c =$ آن طرف که شامل ۱ نیست.
- یک درخت با مجموعه تقسیم = اجتماع همه این B_c ‌ها؟!



● نکته: به ازای هر c و c' :
 $B_c \cap B_{c'} = \emptyset$ یا $B_{c'} \subseteq B_c$ یا $B_c \subseteq B_{c'}$

بے سوی الگوریتم ...

درخت‌های T_i

۱) ریشه روی گونه ۱

۲) منقبض که فقط $W(T)$ را داشته باشند

$T'_i \leq$

به ازای هر تقسیم c , $B_c =$ آن طرف که شامل ۱ نیست.

اصلًا امکان‌پذیر است؟

یک درخت با مجموعه تقسیم = اجتماع همه این B_c ‌ها؟!

نکته: به ازای هر c و c' :

به خاطر اکثریت!

$B_c \cap B_{c'} = \emptyset$ یا $B_{c'} \subseteq B_c$ یا $B_c \subseteq B_{c'}$

بە سوی الگوريتم ...

نکته: بە ازای هر c و c' :

$$B_c \cap B_{c'} = \emptyset \text{ یا } B_{c'} \subseteq B_c \text{ یا } B_c \subseteq B_{c'} \bullet$$

بە سوی الگوريتم ...

● نکته: بە ازای هر c و c' :

$$B_c \cap B_{c'} = \emptyset \text{ یا } B_{c'} \subseteq B_c \text{ یا } B_c \subseteq B_{c'} \bullet$$

● همه تقسیم‌های شامل یک گونه x :

بە سوی الگوریتم ...

نکته: بە ازای هر c و c' :

$$B_c \cap B_{c'} = \emptyset \text{ یا } B_{c'} \subseteq B_c \text{ یا } B_c \subseteq B_{c'}$$

همه تقسیم‌های شامل یک گونه x :

$$B_{x_1} \subseteq B_{x_2} \subseteq \dots \subseteq B_{x_t}$$

بے سوی الگوریتم ...

نکته: به ازای هر c و c' :

$$B_c \cap B_{c'} = \emptyset \text{ یا } B_{c'} \subseteq B_c \text{ یا } B_c \subseteq B_{c'}$$

همه تقسیم‌های شامل یک گونه x :

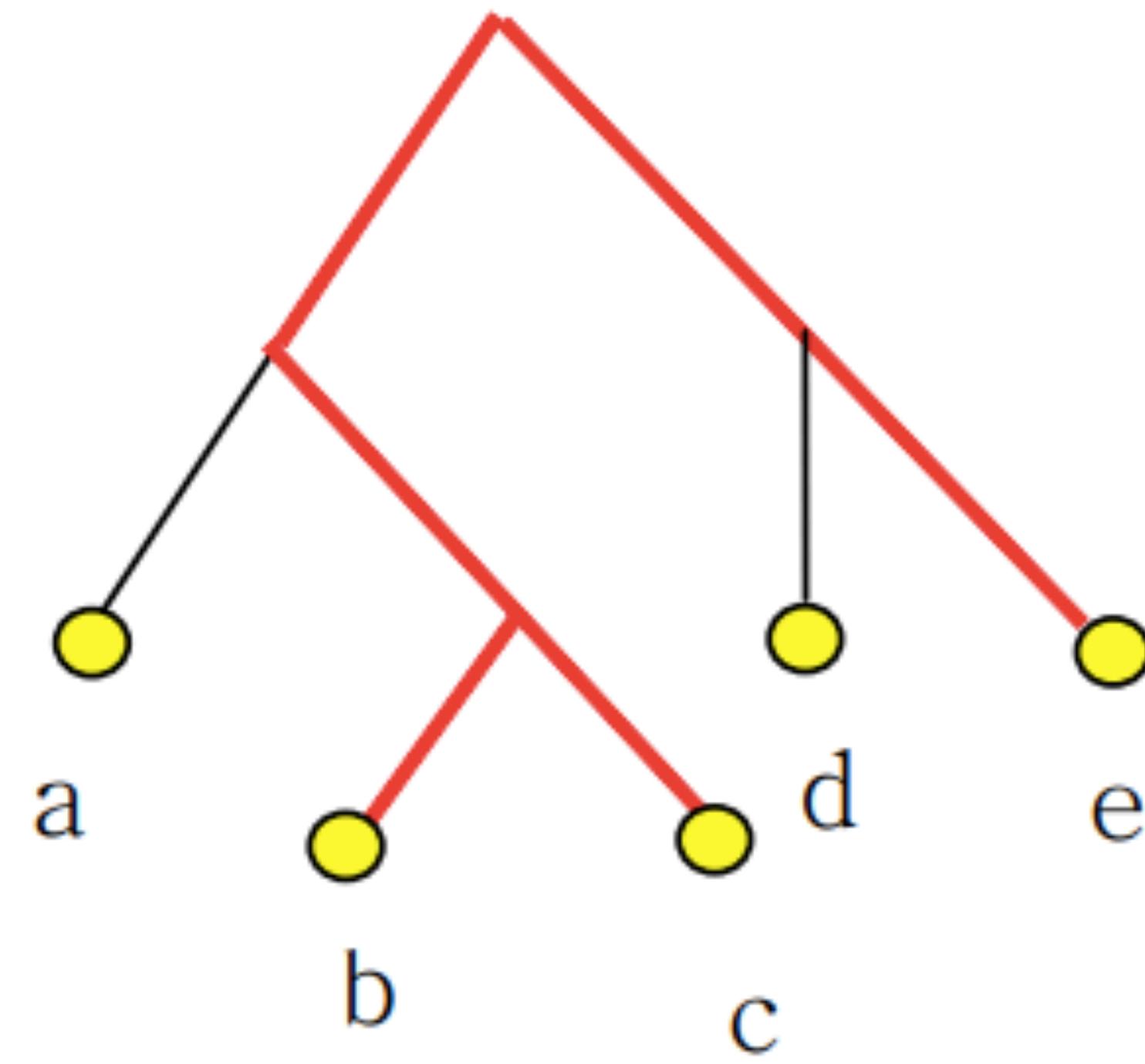
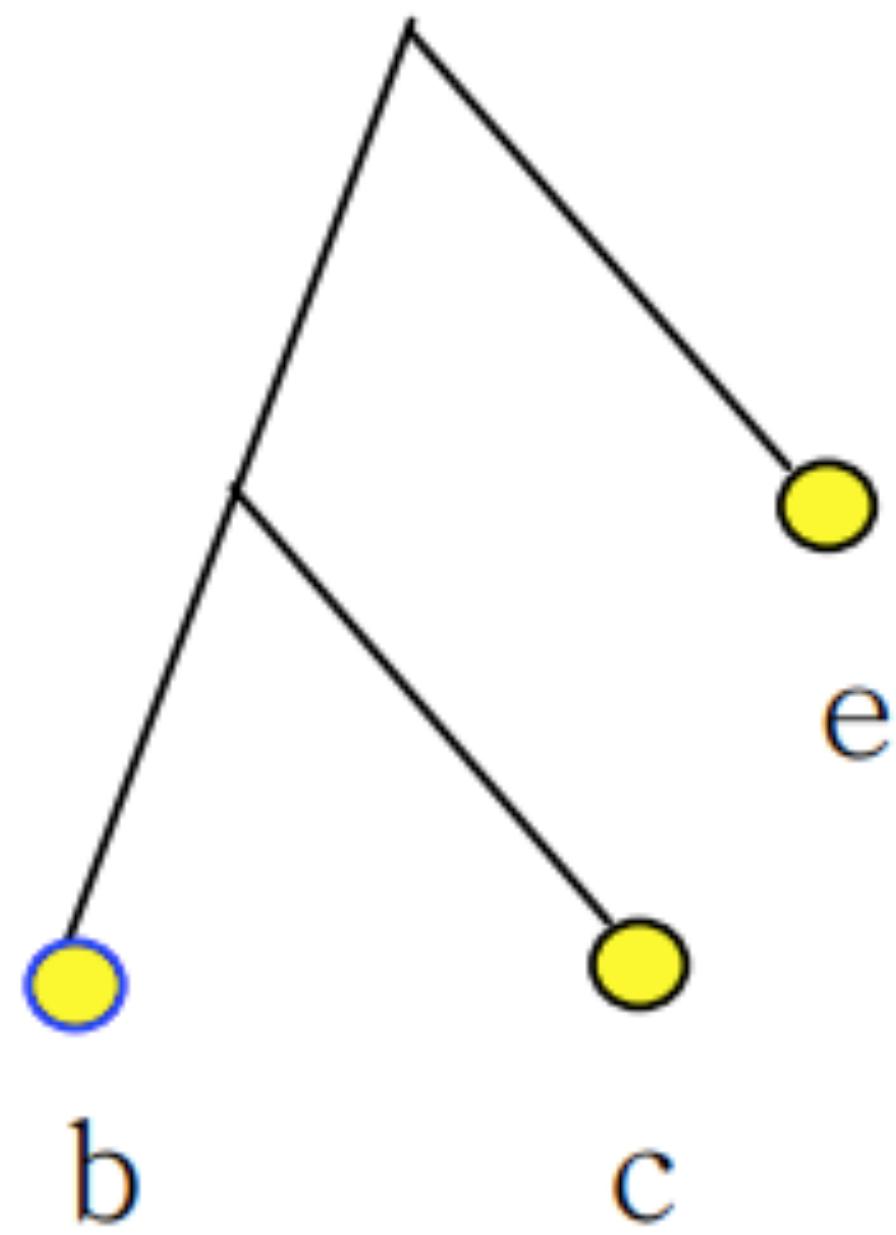
یک زنجیره از B ‌ها می‌توان ساخت که

حتما در یک درخت یال مربوط به B_{x_i} فرزند مستقیم $B_{x_{i+1}}$ است.

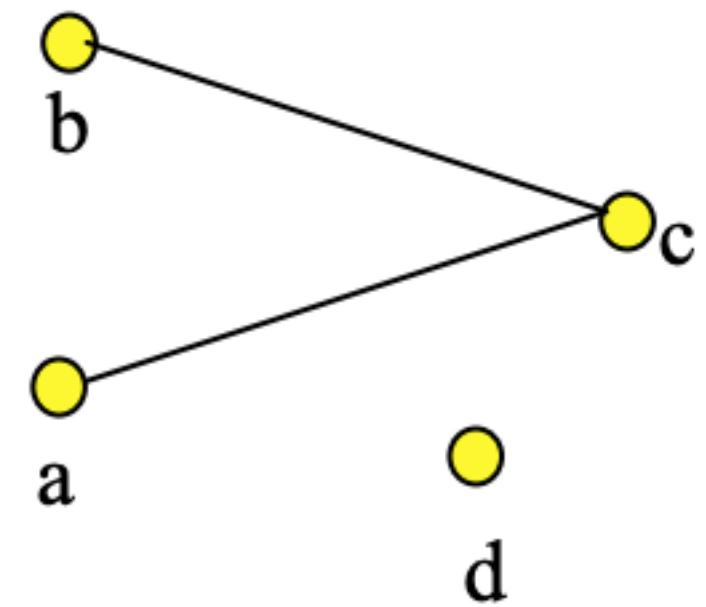
محاسبہ $P[c]$: پال بالاسریک پال c

```
5: for  $i = 1$  to  $m$  do
6:   for every edge  $c$  in  $T'_i$  do
7:     Let  $q$  be the parent edge of  $c$  in  $T'_i$ .
8:     if  $P[q] = \emptyset$  or  $|B_c| > |B_{P[q]}|$  then
9:        $P[q] = c$ .
10:      end if
11:    end for
12:  end for
```

سہ تا یی



بازسازی از روی سه تایی ها



$$bc \mid a, ac \mid d$$

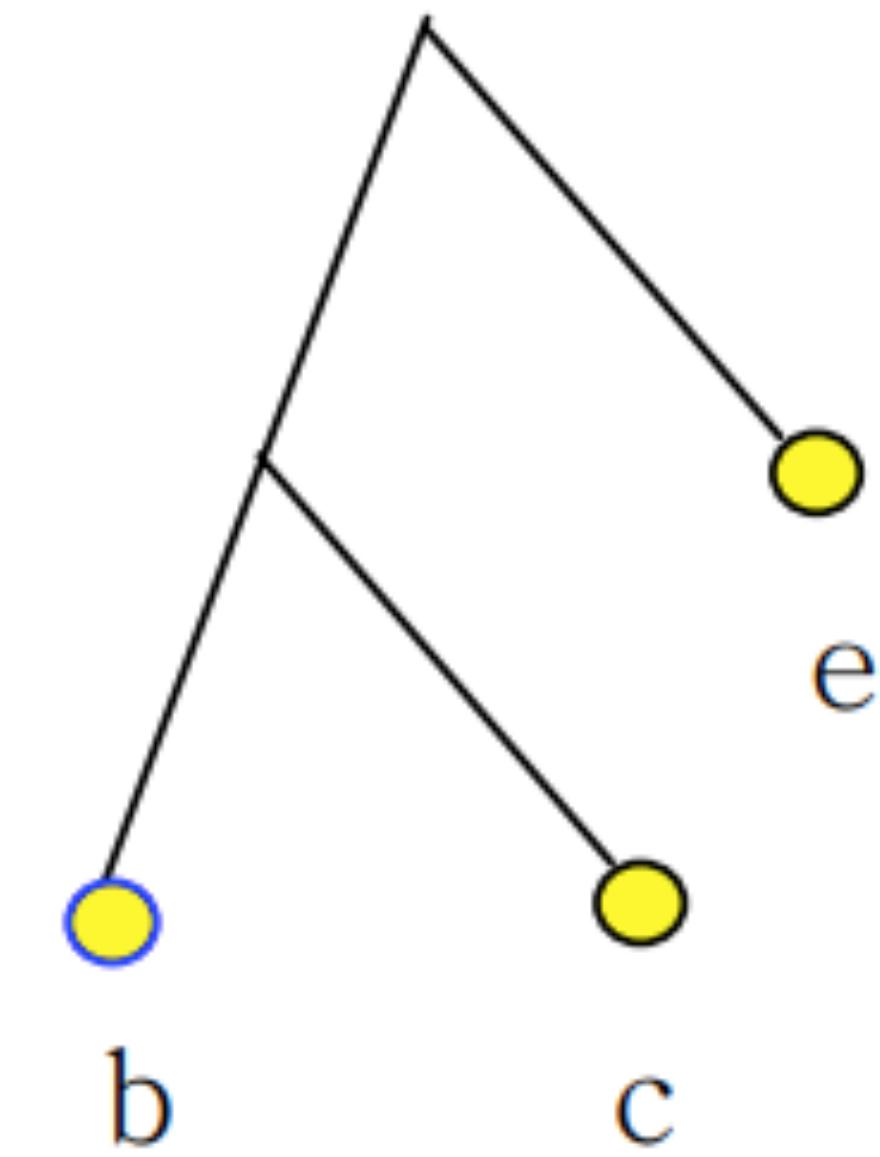
● گراف باقی مانده

● ورودی: سه تایی ها

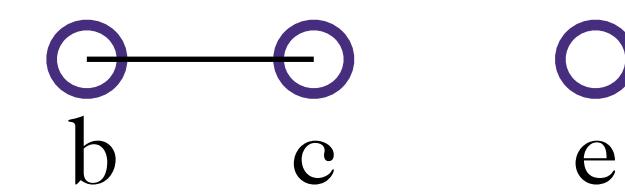
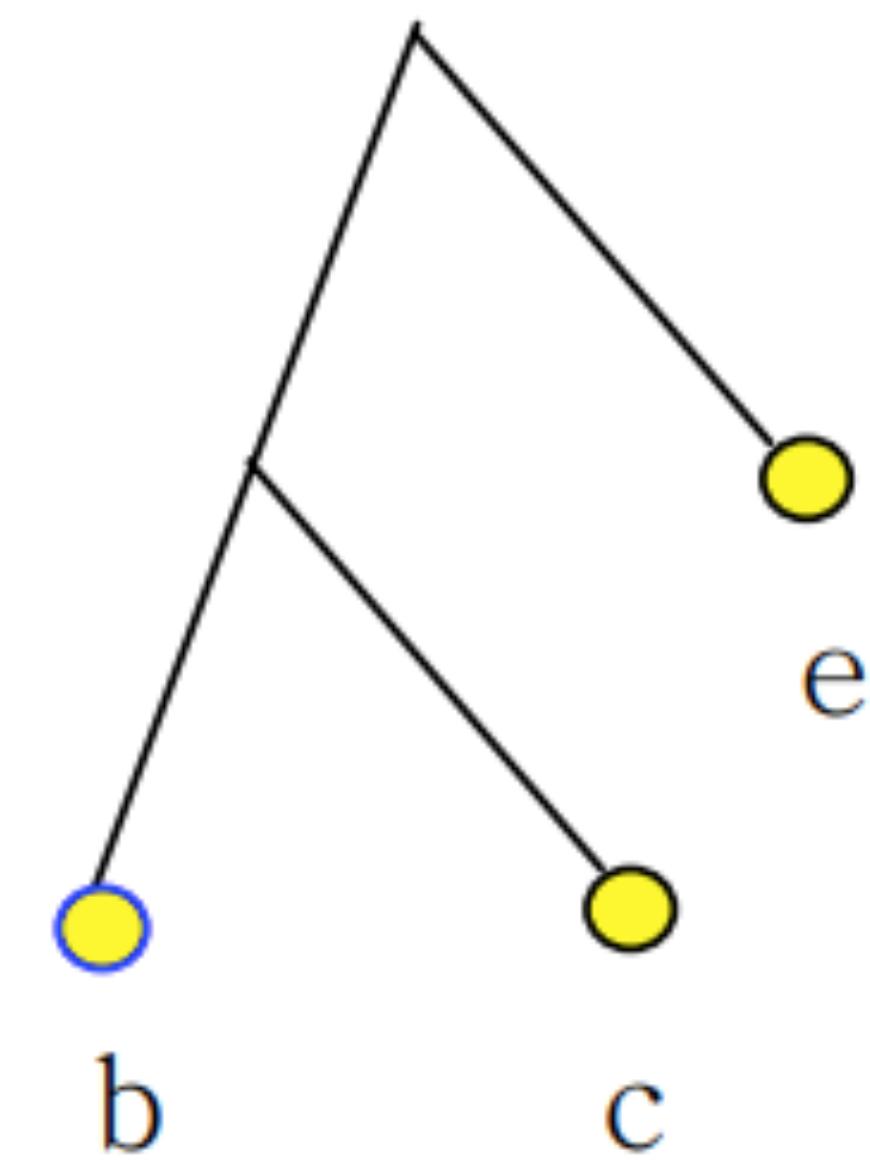
● يال بین a و b $\leq ab \mid c$

گراف باقیمانده

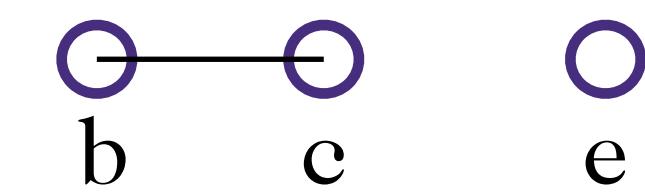
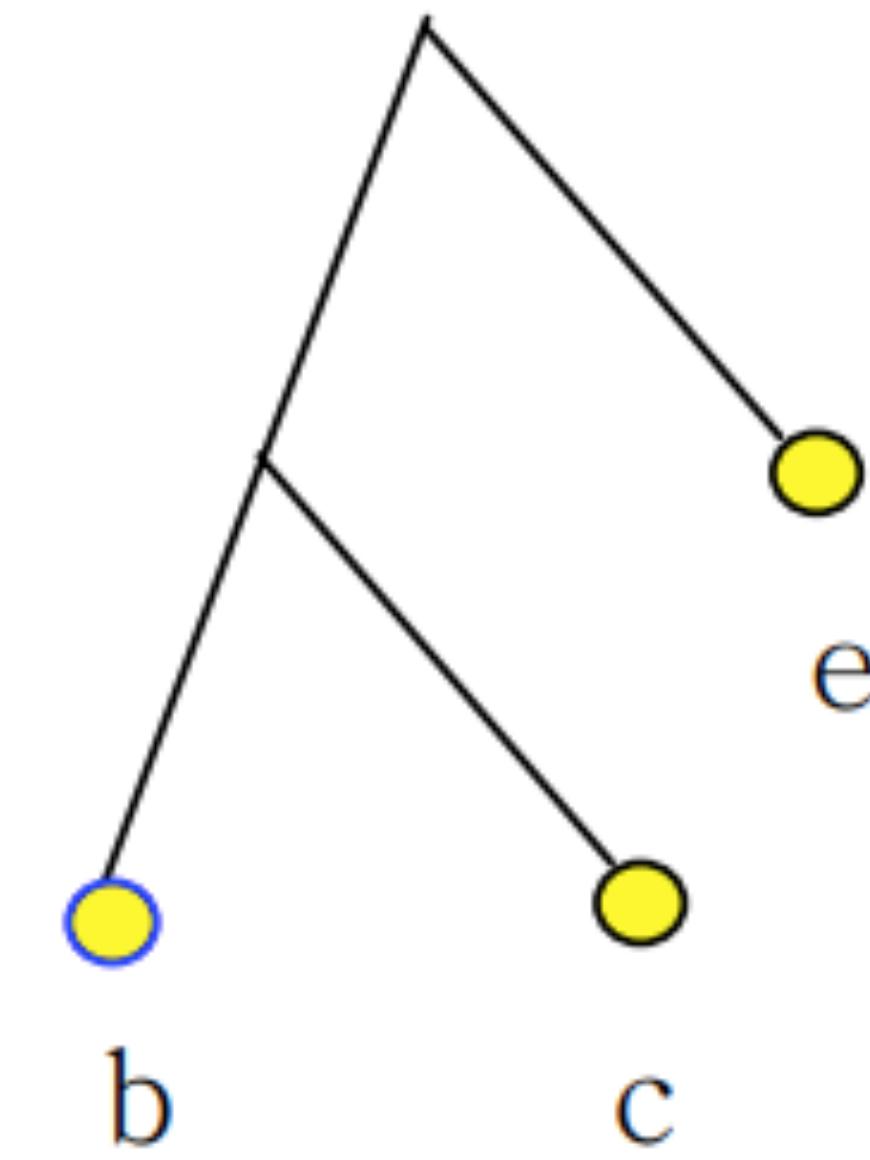
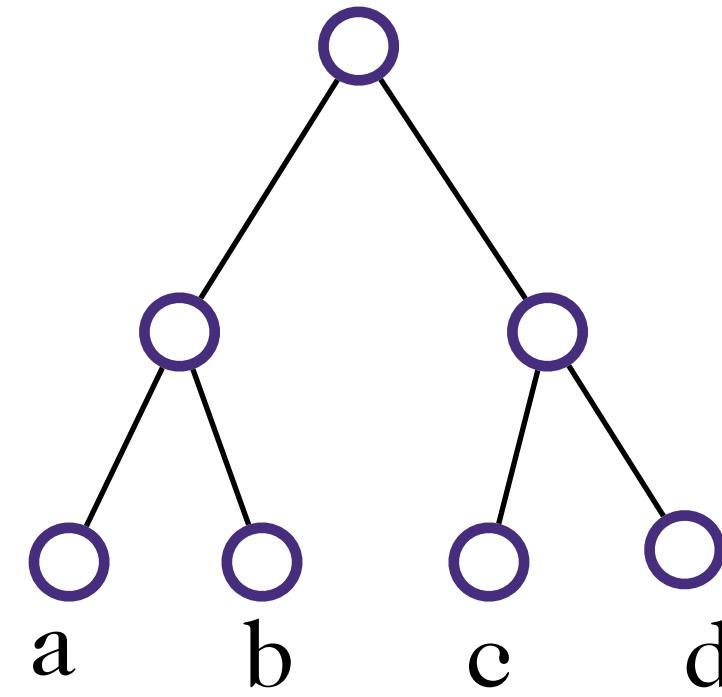
گراف باقیمانده



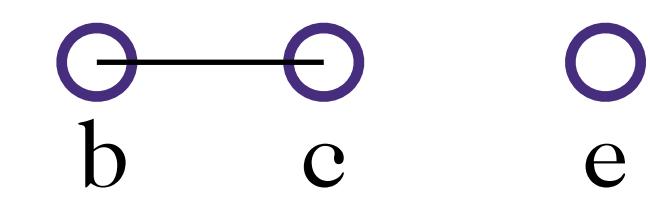
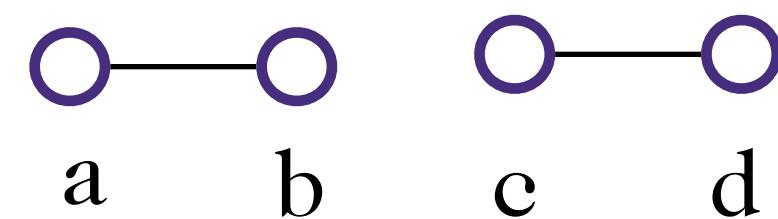
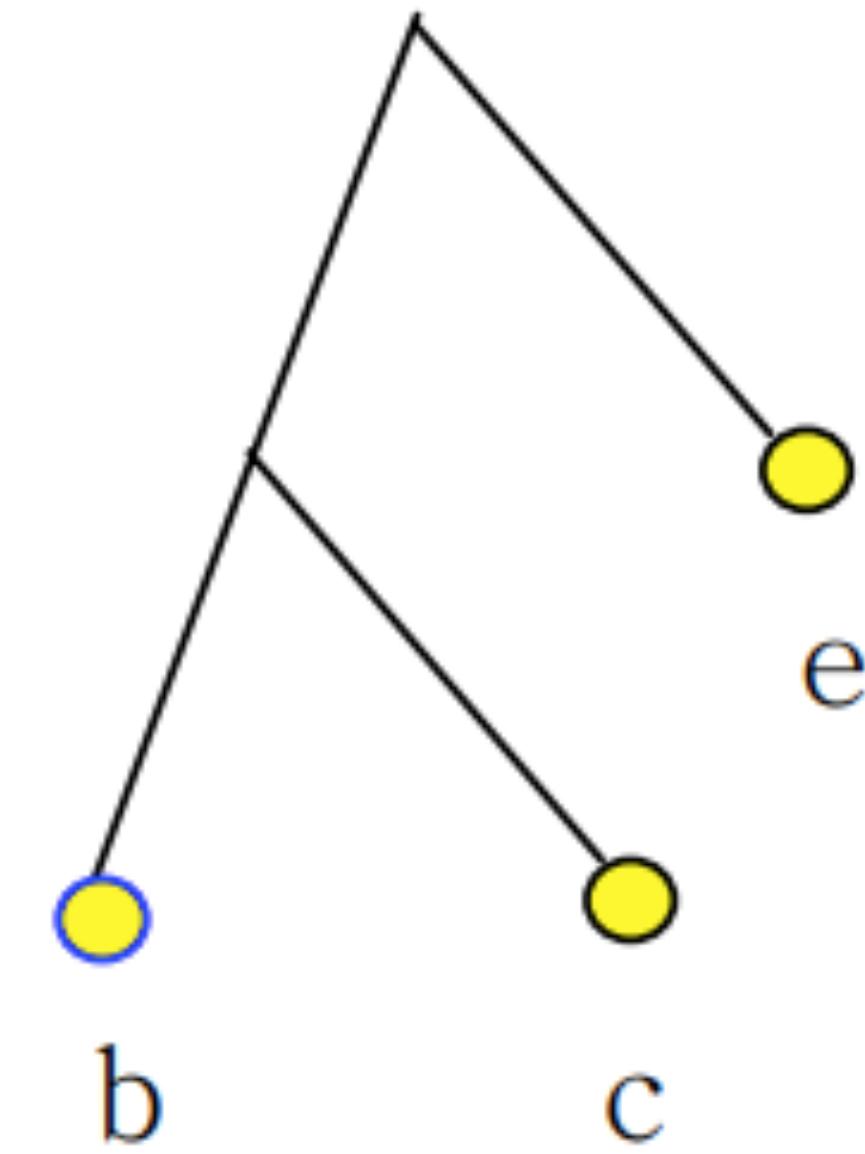
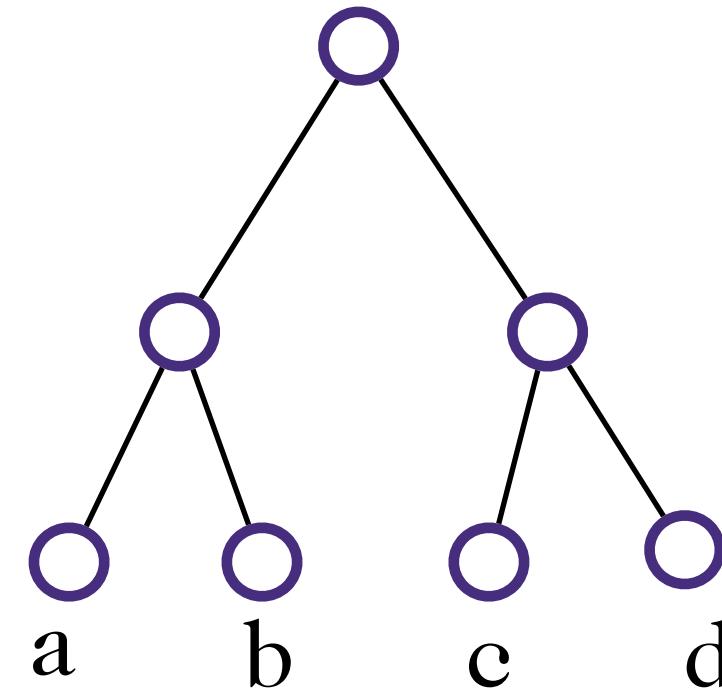
گراف باقیمانده



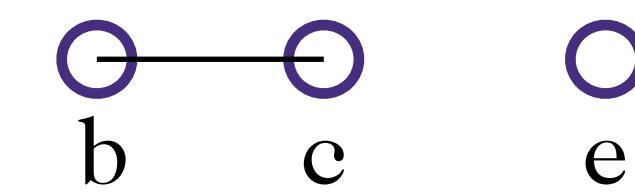
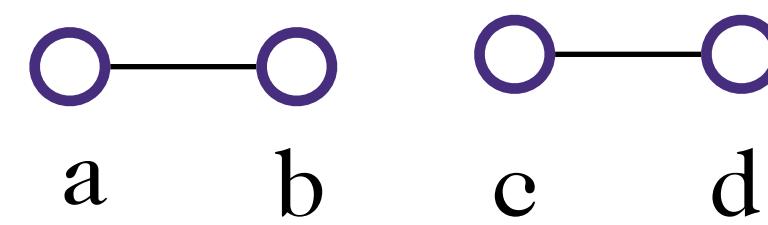
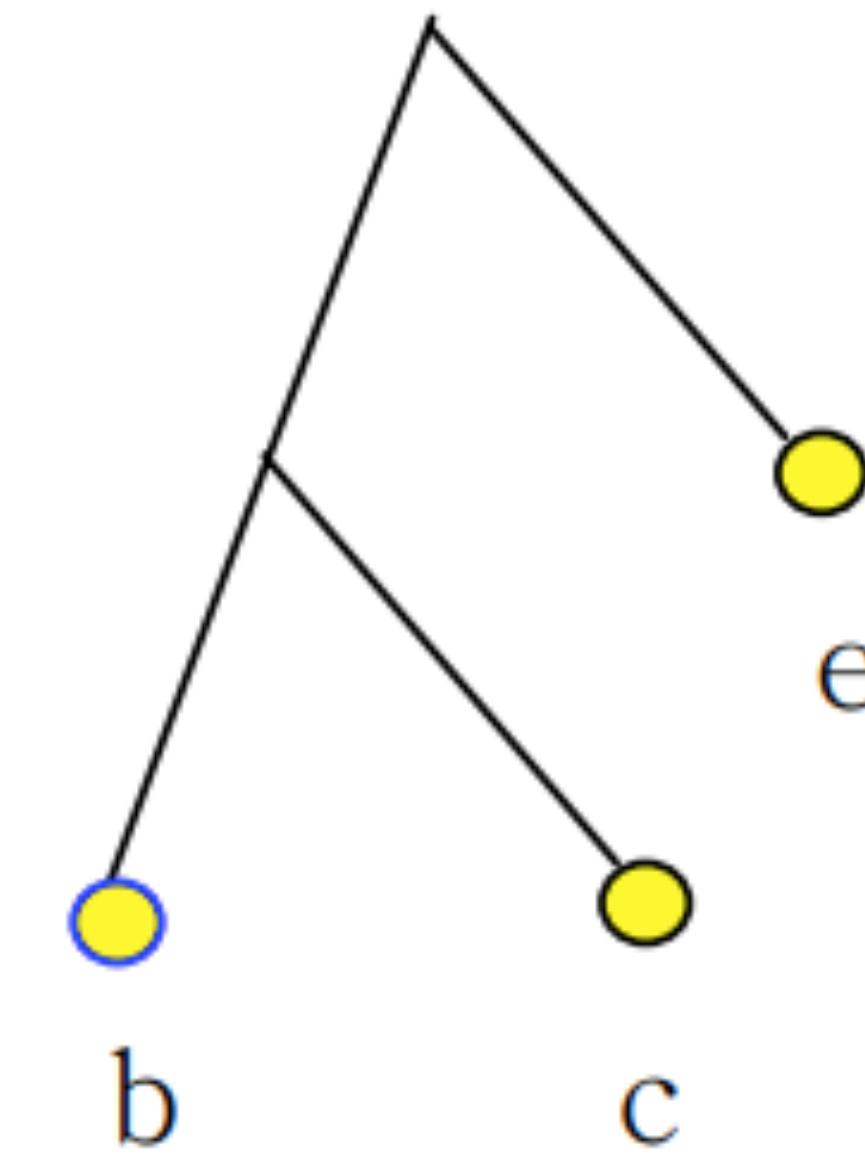
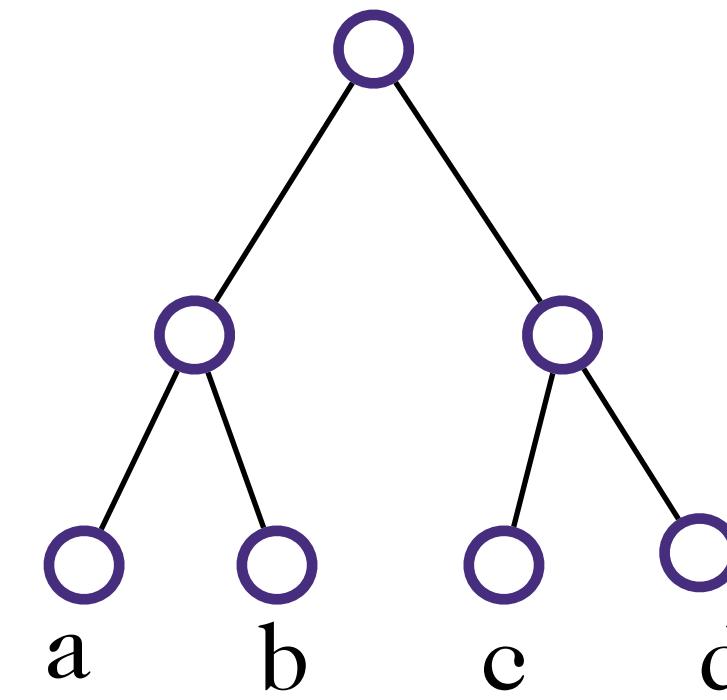
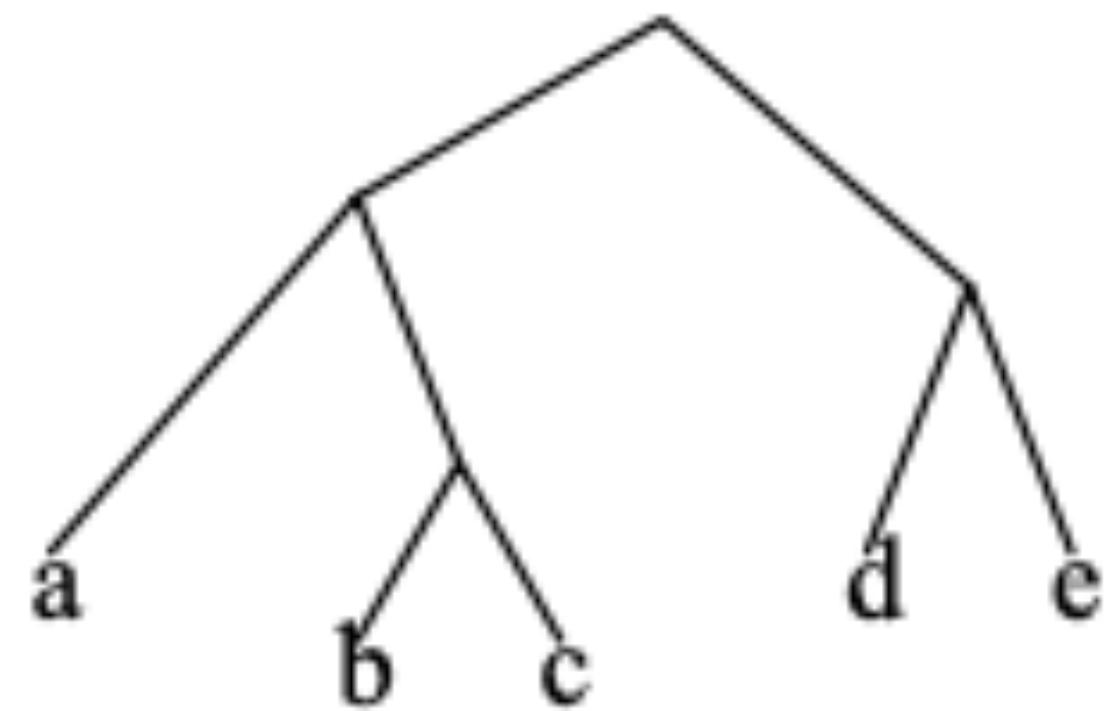
گراف باقیمانده



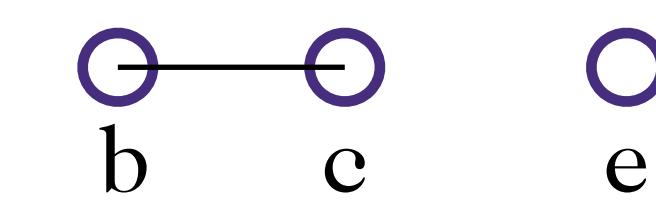
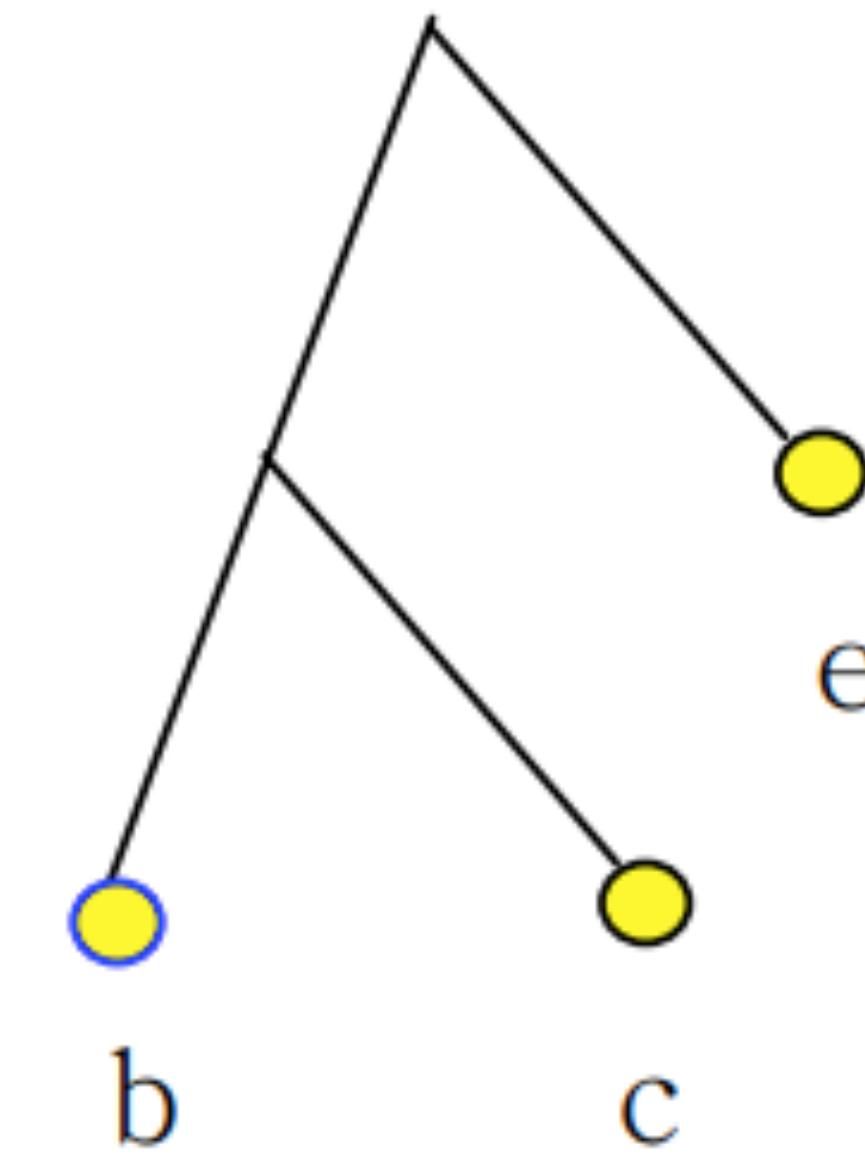
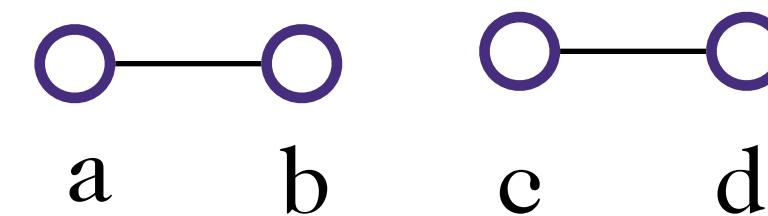
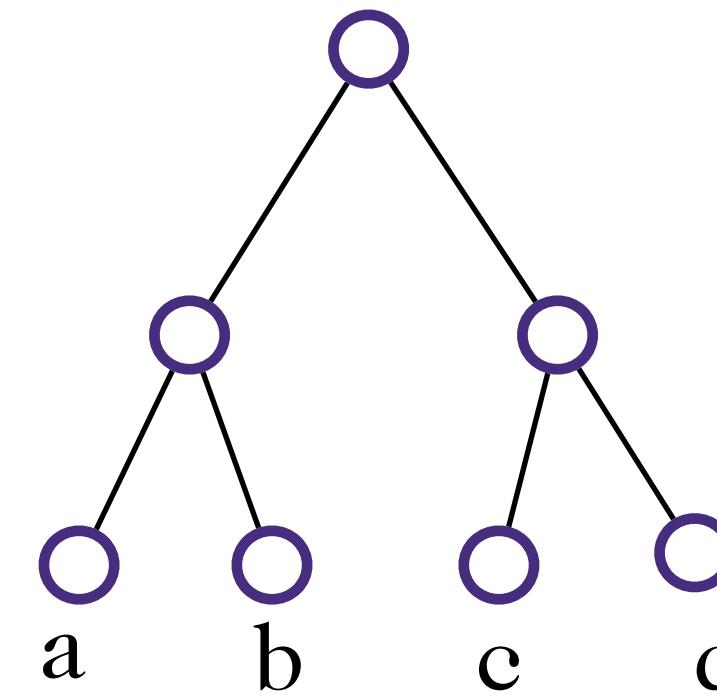
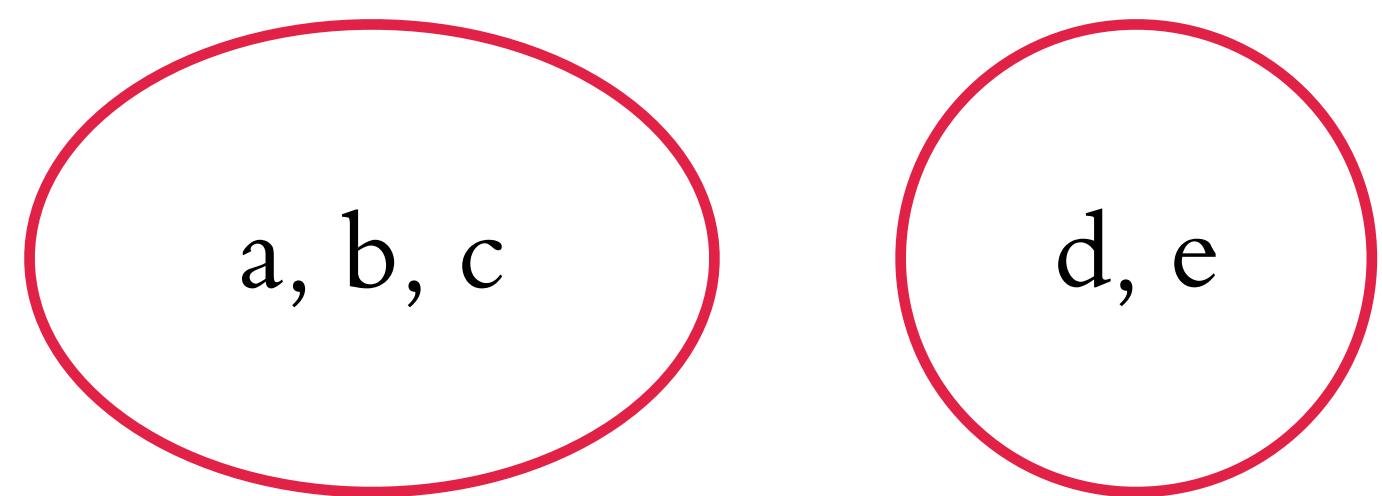
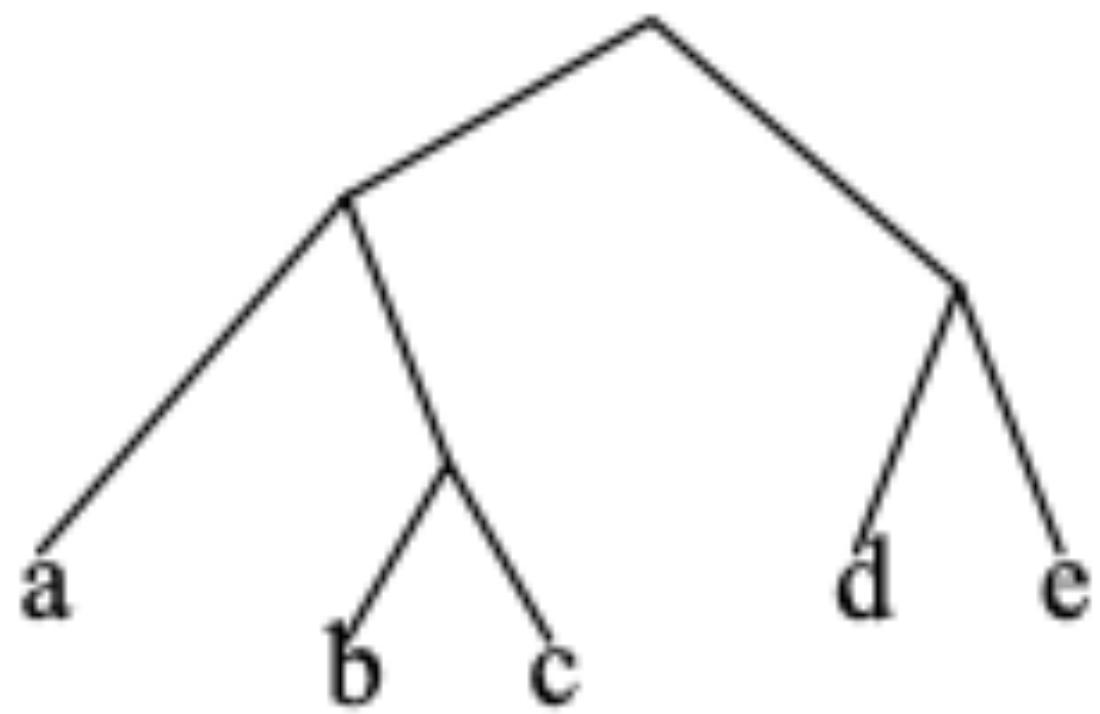
گراف باقیمانده



گراف باقیمانده



گراف باقیمانده



الگوريتم

الگوریتم

● گراف باقیمانده را بساز

الگوریتم

- گراف باقیمانده را بساز
- اگر ناهمبند بود،

الگوریتم

- گراف باقیمانده را بساز
- اگر ناهمبند بود،
- برای هر زیرمجموعه درخت را بساز (بازگشتی)

الگوریتم

- گراف باقیمانده را بساز
- اگر ناهمبند بود،
- برای هر زیرمجموعه درخت را بساز (بازگشتی)
- یک راس ریشه جدید اضافه کن و همه درخت‌ها را زیردرخت آن بگذار

الگوریتم

- گراف باقیمانده را بساز
- اگر ناهمبند بود،
- برای هر زیرمجموعه درخت را بساز (بازگشتی)
- یک راس ریشه جدید اضافه کن و همه درخت‌ها را زیردرخت آن بگذار
- این درخت را برگردان

الگوریتم

- گراف باقیمانده را بساز
- اگر ناهمبند بود،
- برای هر زیرمجموعه درخت را بساز (بازگشتی)
- یک راس ریشه جدید اضافه کن و همه درخت‌ها را زیردرخت آن بگذار
- این درخت را برگردان
- اگر نه،

الگوریتم

- گراف باقیمانده را بساز
- اگر ناهمبند بود،
- برای هر زیرمجموعه درخت را بساز (بازگشتی)
- یک راس ریشه جدید اضافه کن و همه درخت‌ها را زیردرخت آن بگذار
- این درخت را برگردان
- اگر نه،
- باید حتماً تک راس باقیمانده باشد، و گرنه نمی‌توان درخت ساخت

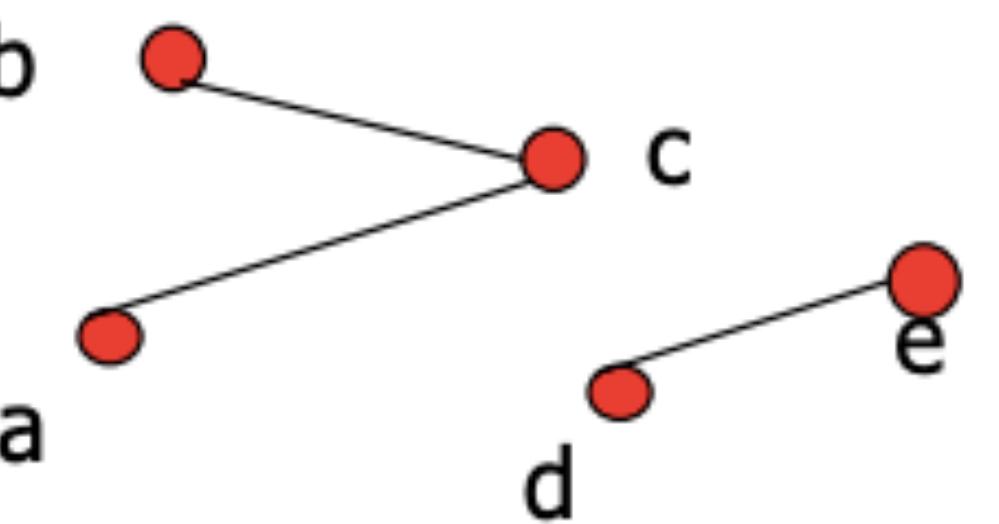
الگوریتم

- گراف باقیمانده را بساز
- اگر ناهمبند بود،
- برای هر زیرمجموعه درخت را بساز (بازگشتی)
- یک راس ریشه جدید اضافه کن و همه درخت‌ها را زیردرخت آن بگذار
- این درخت را برگردان
- اگر نه،
- باید حتماً تک راس باقیمانده باشد، و گرنه نمی‌توان درخت ساخت
- آن تک راس را به عنوان درخت بازسازی شده برگردان

مثال

$ab \mid c, ac \mid d, de \mid b$

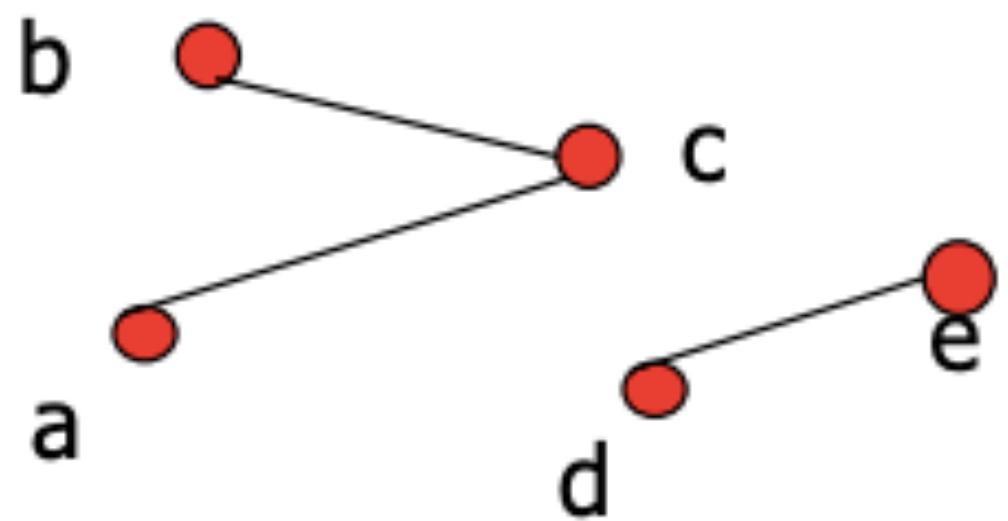
مثال



$G(\{a,b,c,d,e\})$

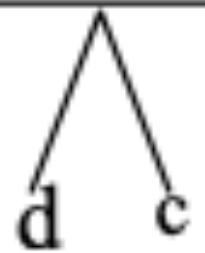
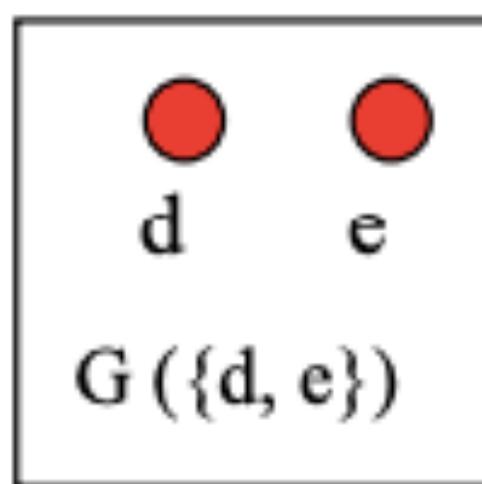
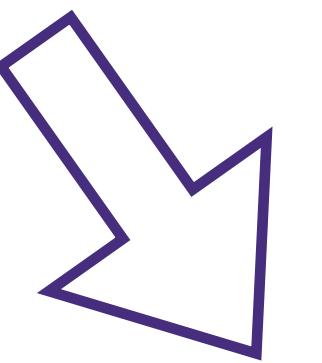
$ab \perp c, ac \perp d, de \perp b$

مثال

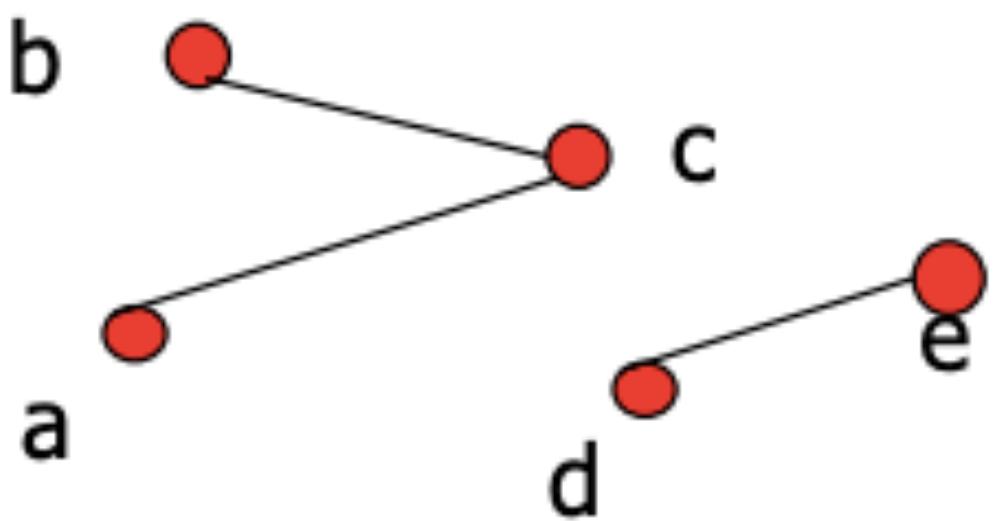


$G(\{a,b,c,d,e\})$

$ab \sqsubset c, ac \sqsubset d, de \sqsubset b$

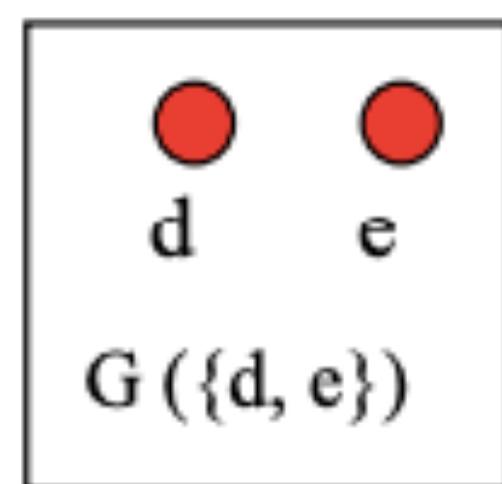
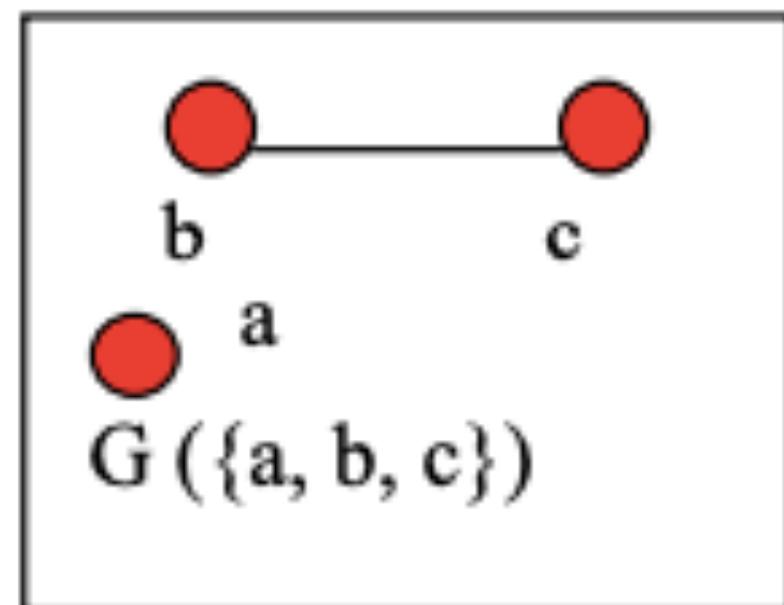
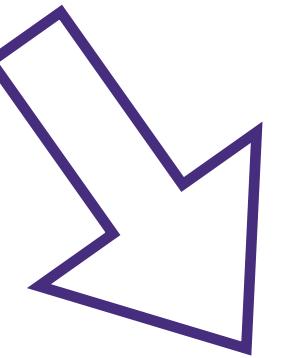
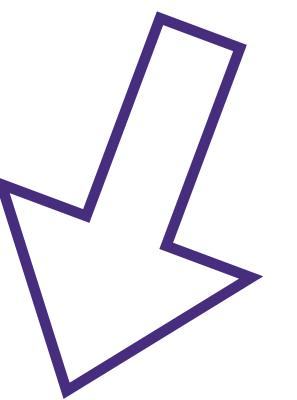


مثال

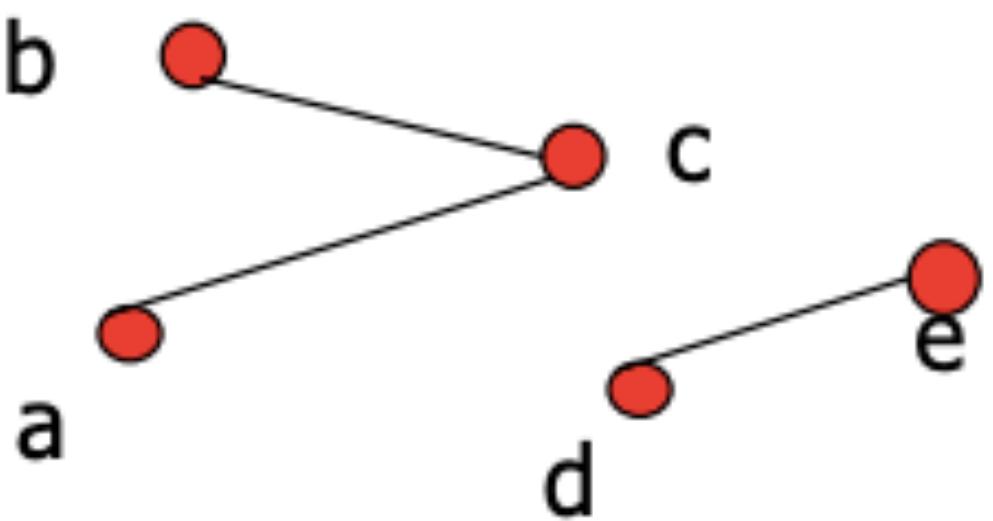


$ab \sqsubset c, ac \sqsubset d, de \sqsubset b$

$G(\{a,b,c,d,e\})$

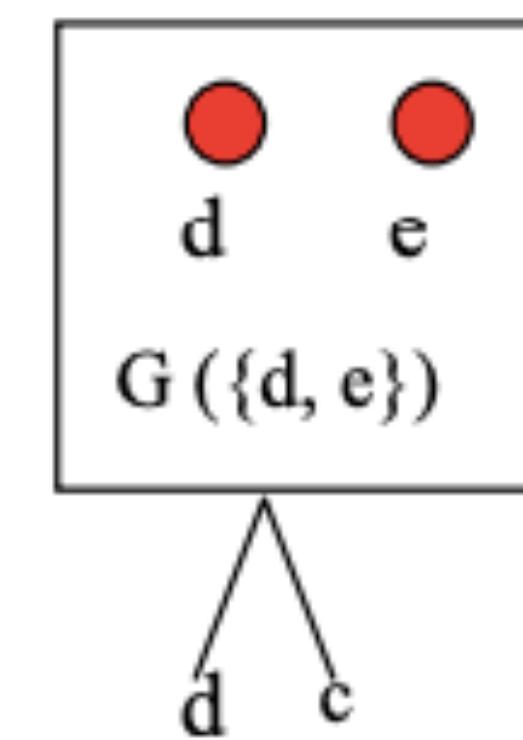
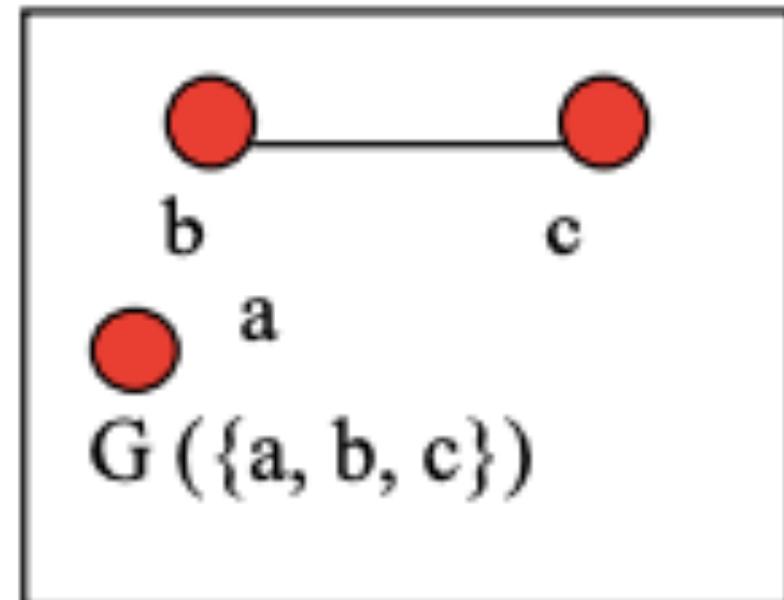
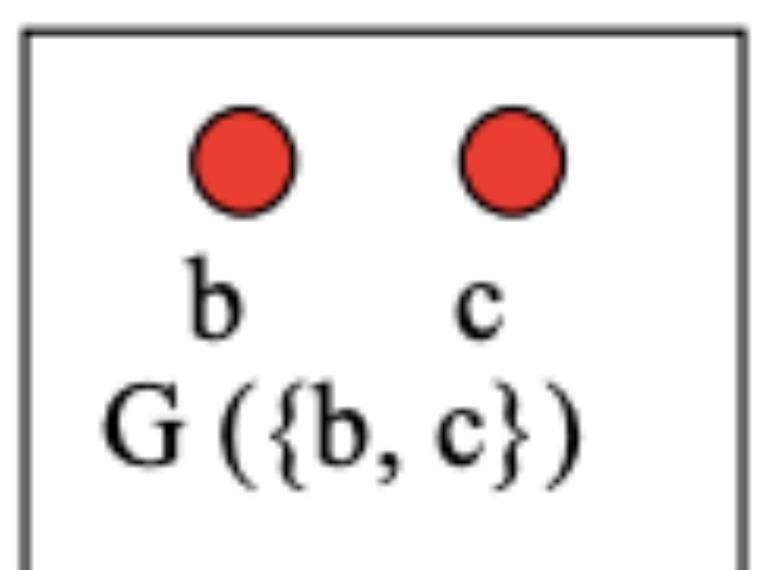


مثال

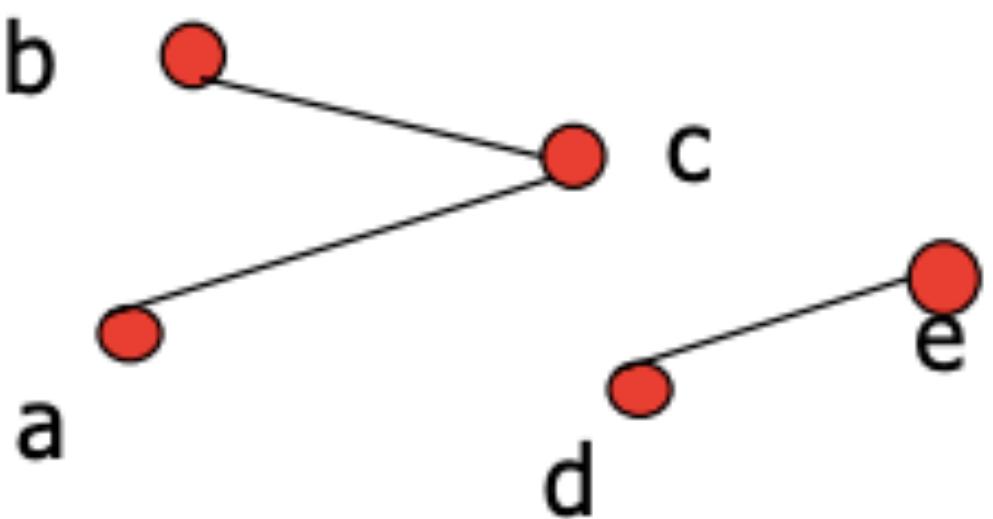


$ab \sqsubset c, ac \sqsubset d, de \sqsubset b$

$G(\{a,b,c,d,e\})$

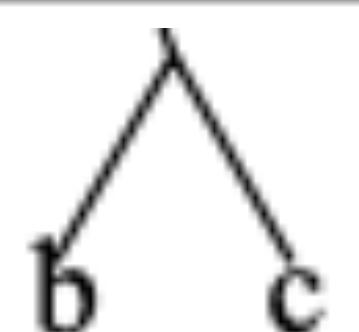
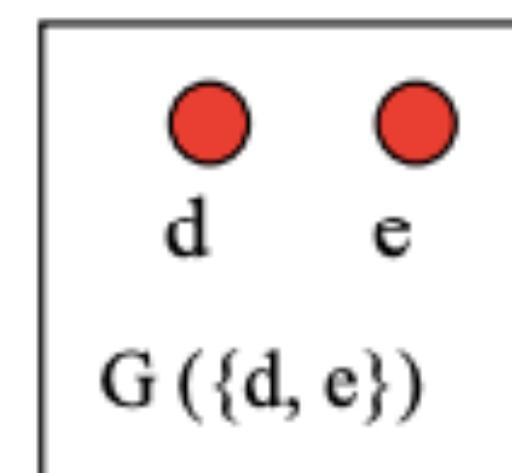
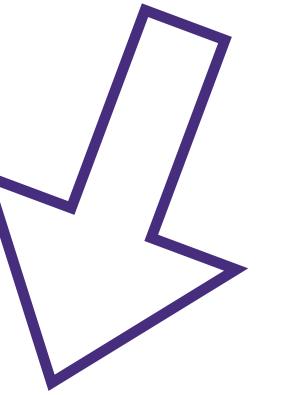
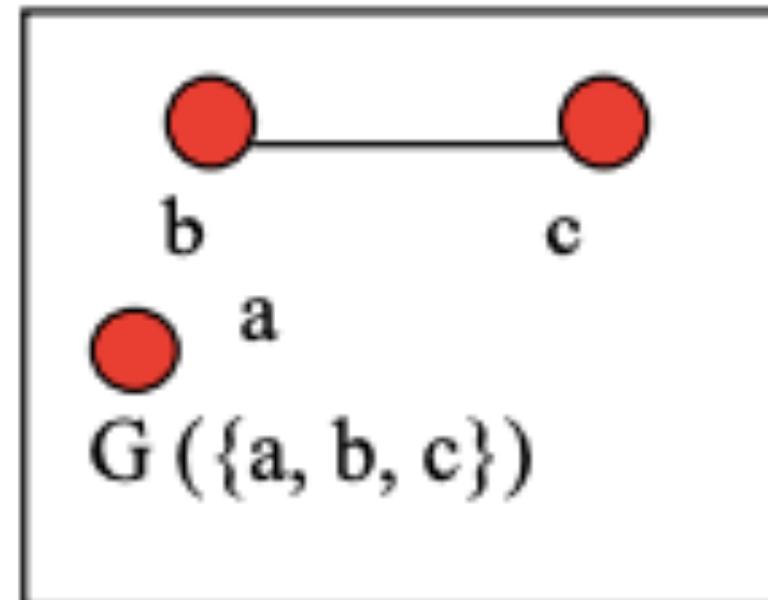
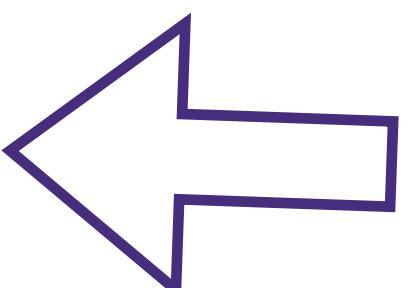
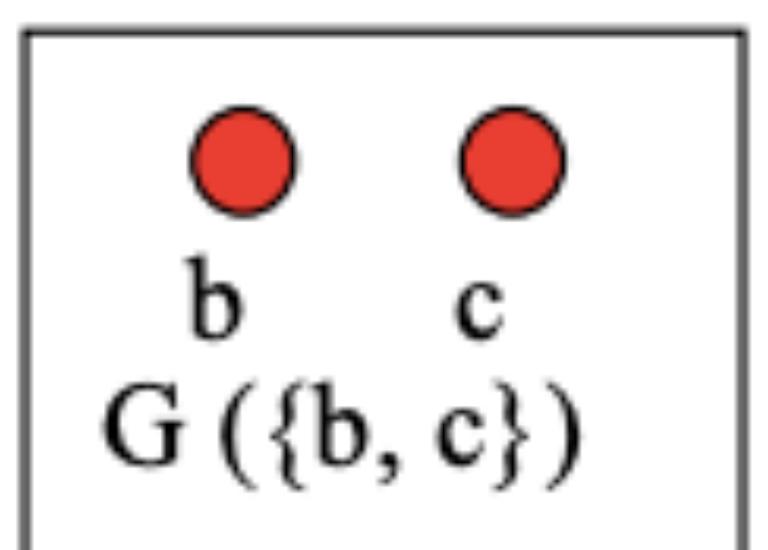


مثال

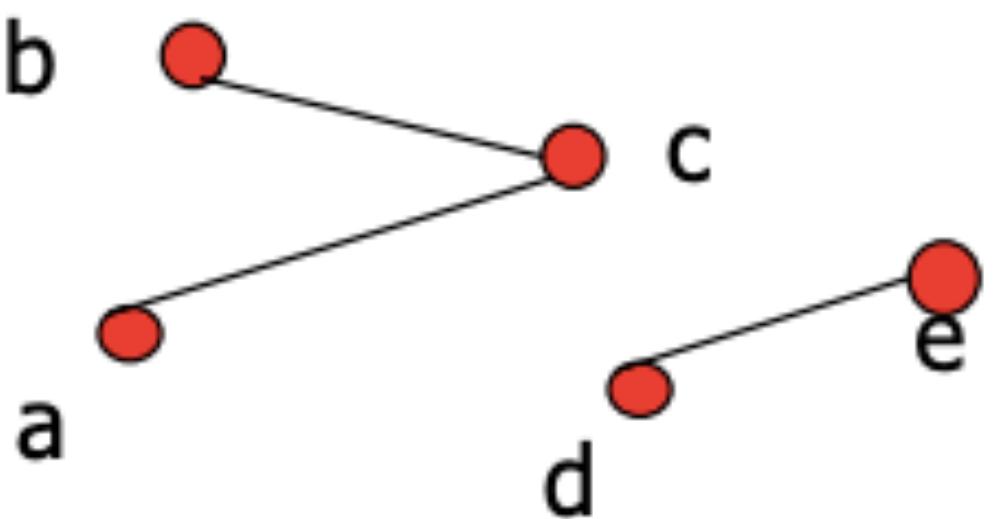


$ab \sqsubset c, ac \sqsubset d, de \sqsubset b$

$G(\{a,b,c,d,e\})$

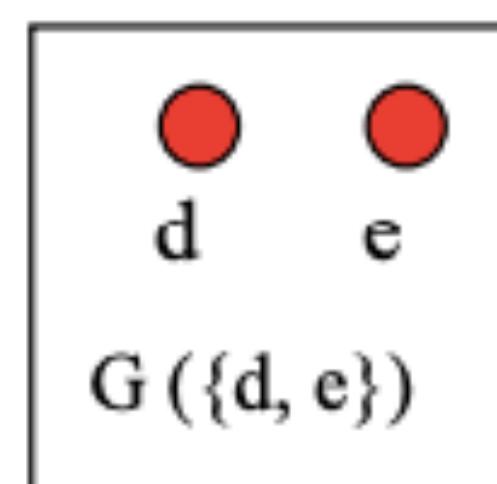
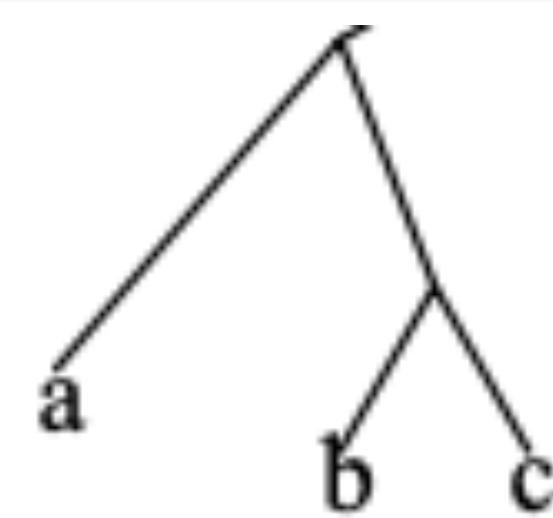
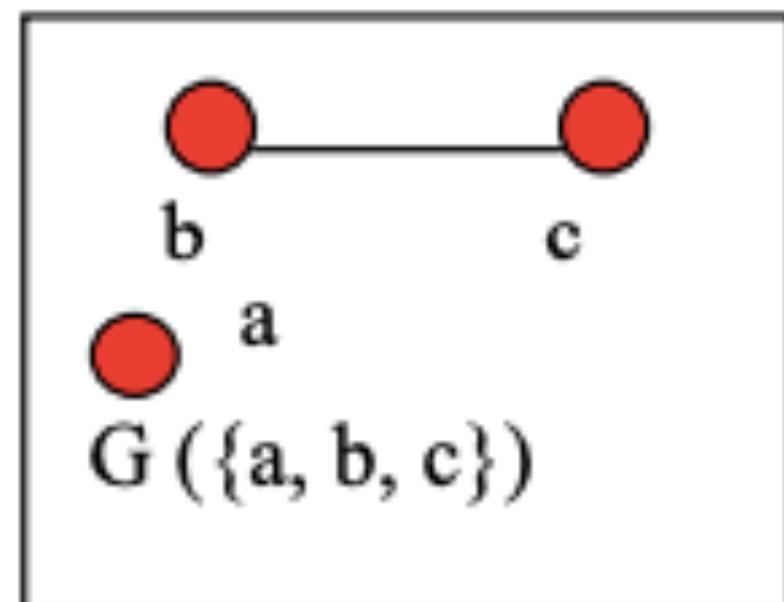
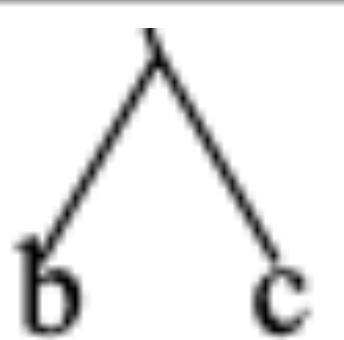
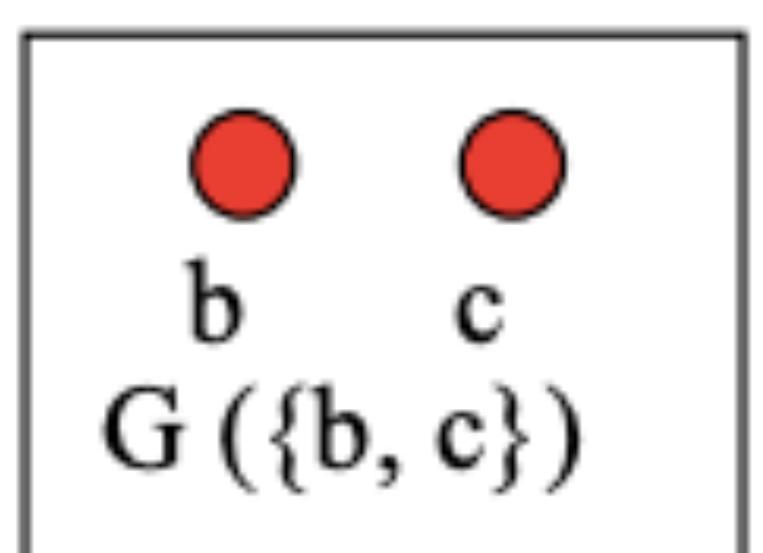


مثال

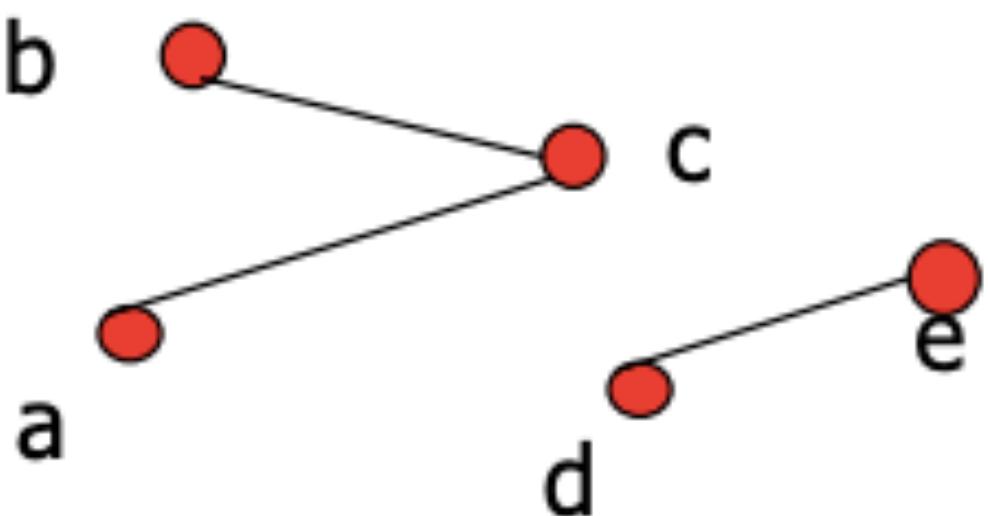
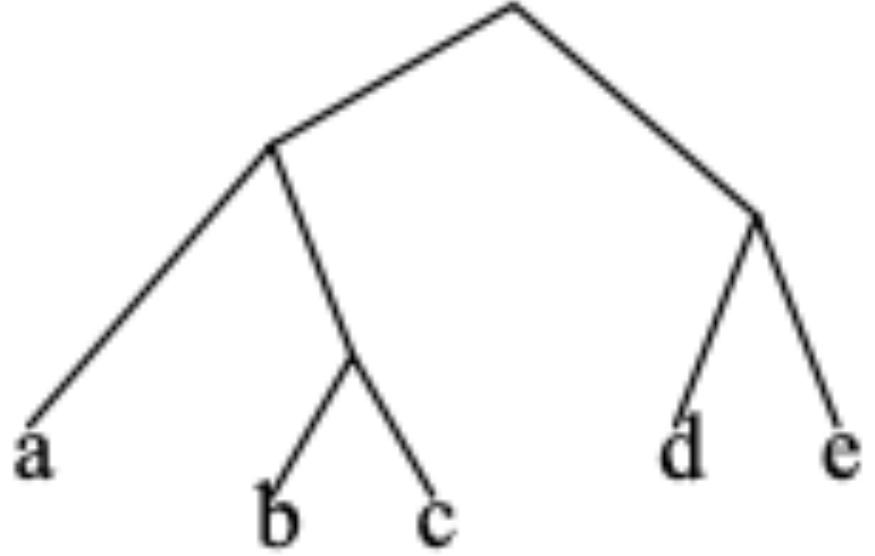


$G(\{a,b,c,d,e\})$

$ab \sqsubset c, ac \sqsubset d, de \sqsubset b$

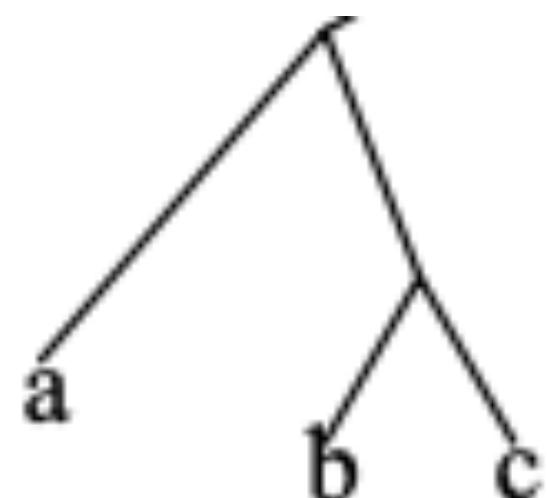
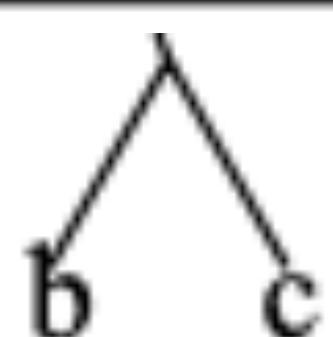
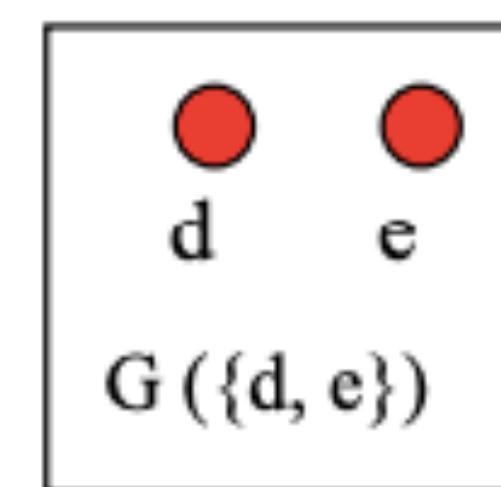
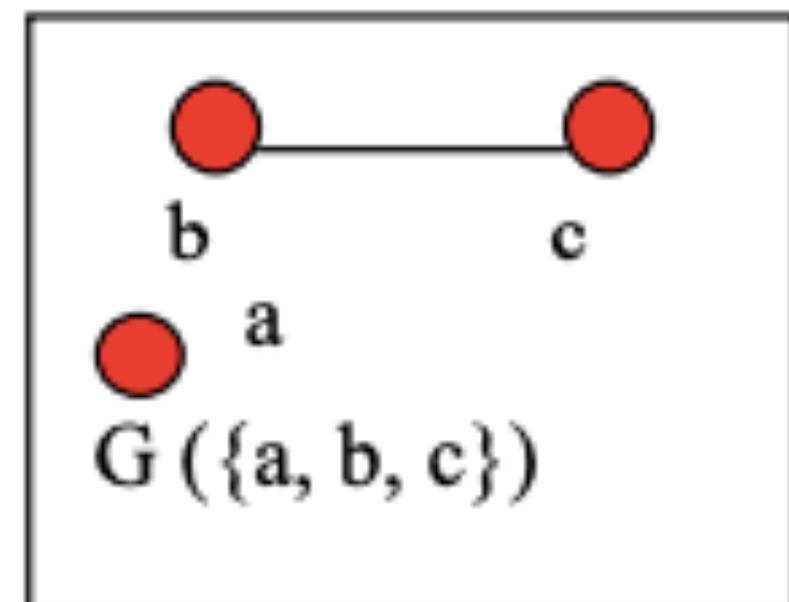
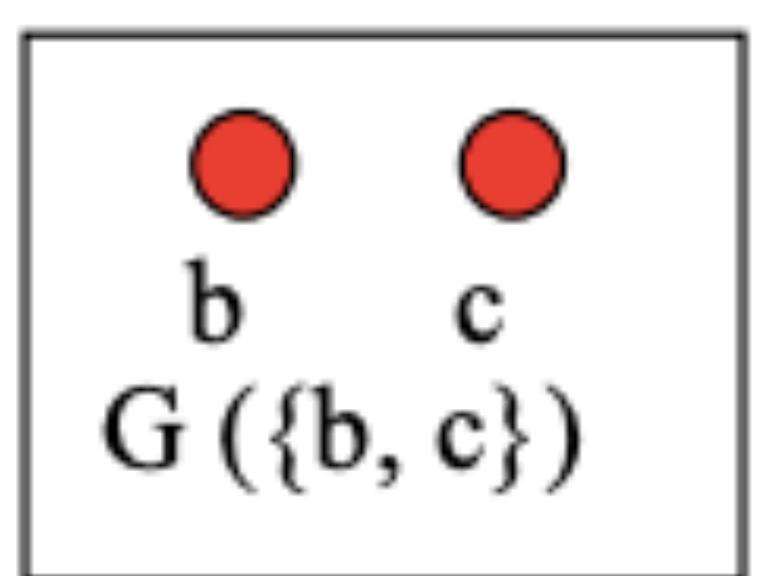


مثال

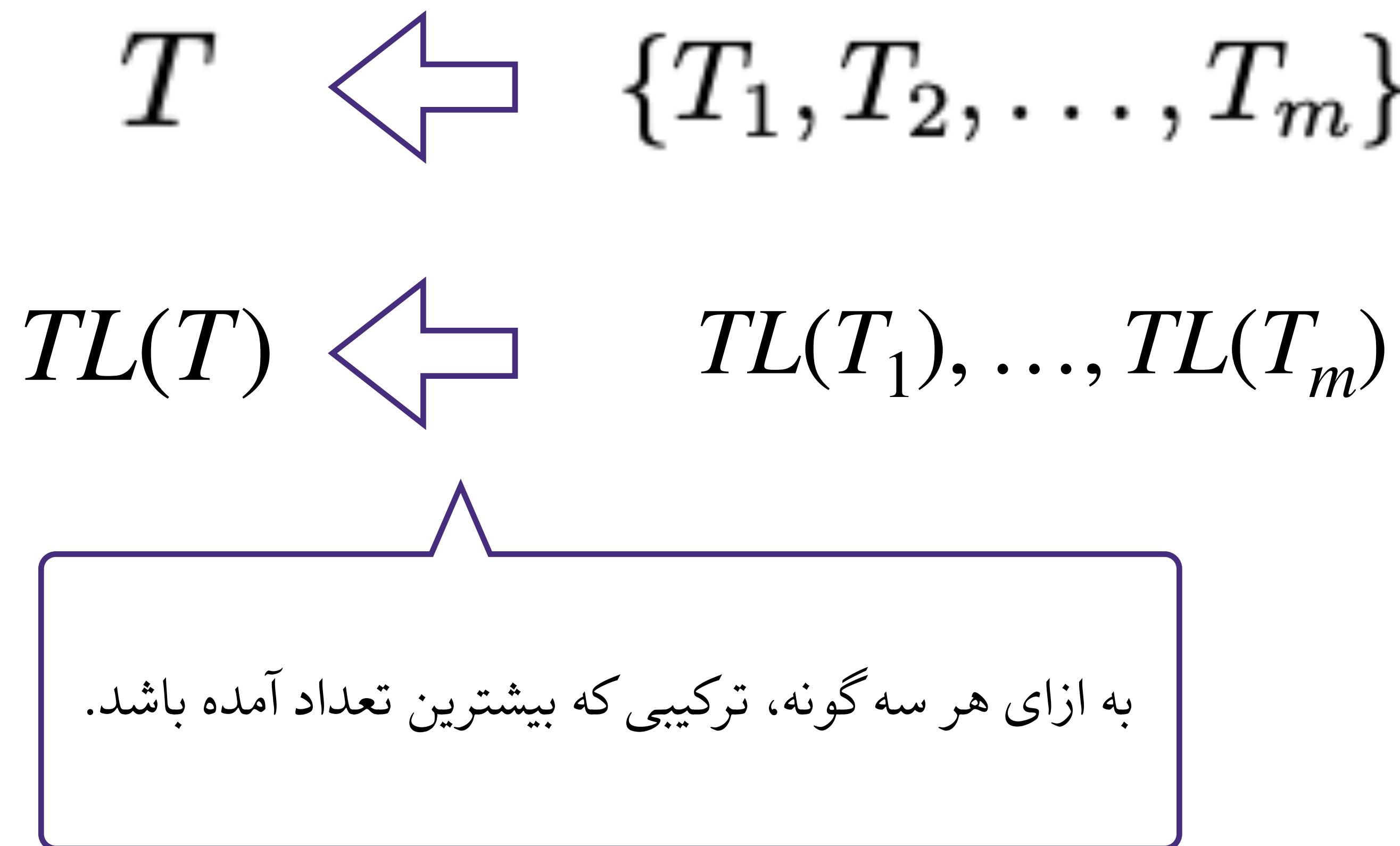


$G(\{a, b, c, d, e\})$

$ab \sqsubset c, ac \sqsubset d, de \sqsubset b$



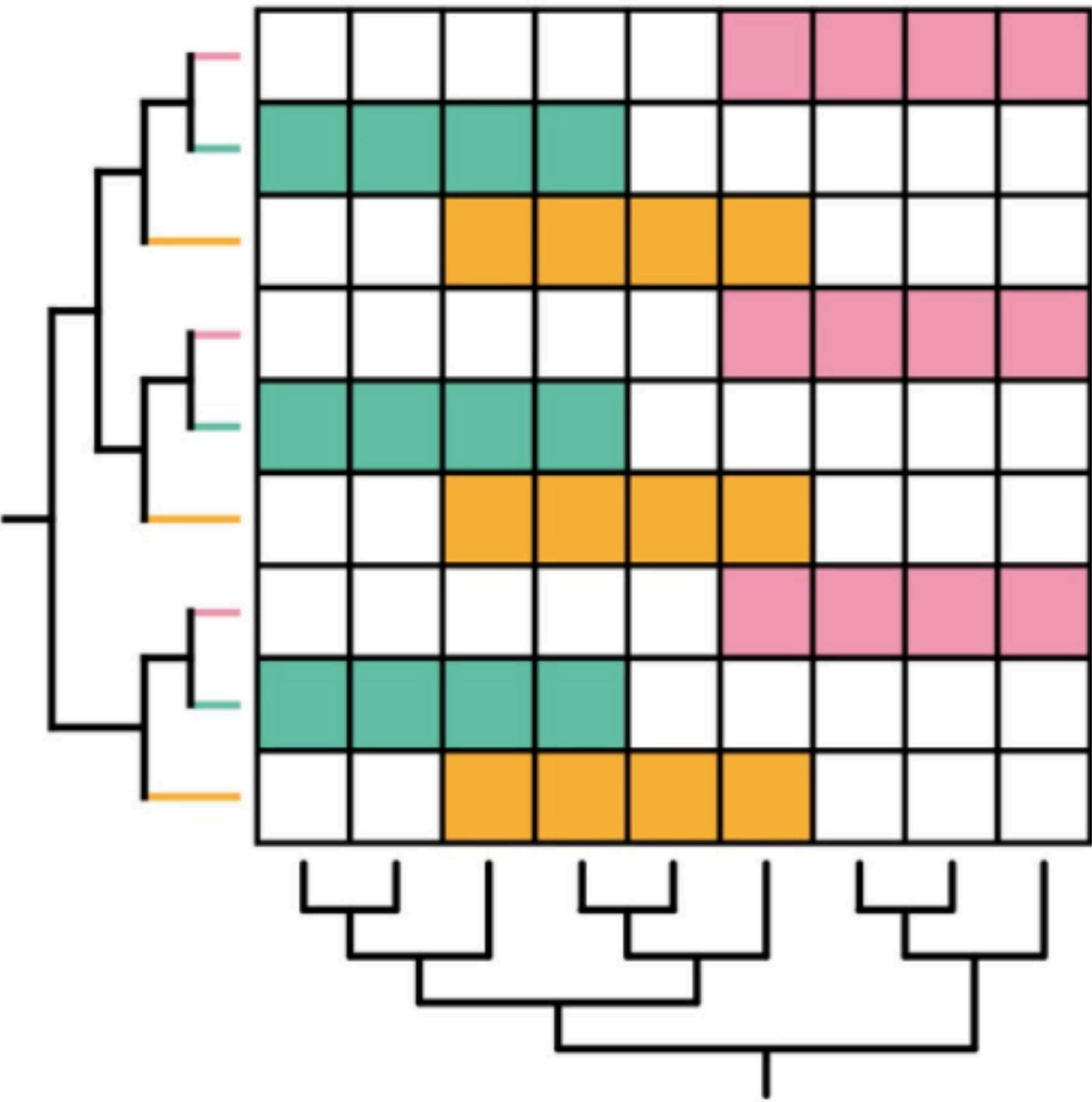
درخت R^*



الگوریتم

● با الگوریتم بازسازی سه تایی ها

چطوری یک درخت را
بنویسیم؟



قابل Newick

(, , (,));	<i>no nodes are named</i>
(A,B,(C,D));	<i>leaf nodes are named</i>
(A,B,(C,D)E)F;	<i>all nodes are named</i>
(:0.1,:0.2,(:0.3,:0.4):0.5);	<i>all but root node have a distance to parent</i>
(:0.1,:0.2,(:0.3,:0.4):0.5):0.0;	<i>all have a distance to parent</i>
(A:0.1,B:0.2,(C:0.3,D:0.4):0.5);	<i>distances and leaf names (popular)</i>
(A:0.1,B:0.2,(C:0.3,D:0.4)E:0.5)F;	<i>distances and all names</i>
((B:0.2,(C:0.3,D:0.4)E:0.5)F:0.1)A;	<i>a tree rooted on a leaf node (rare)</i>

NEXUS

```
#NEXUS
Begin TAXA;
    Dimensions ntax=4;
    TaxLabels SpaceDog SpaceCat SpaceOrc SpaceElf
End;

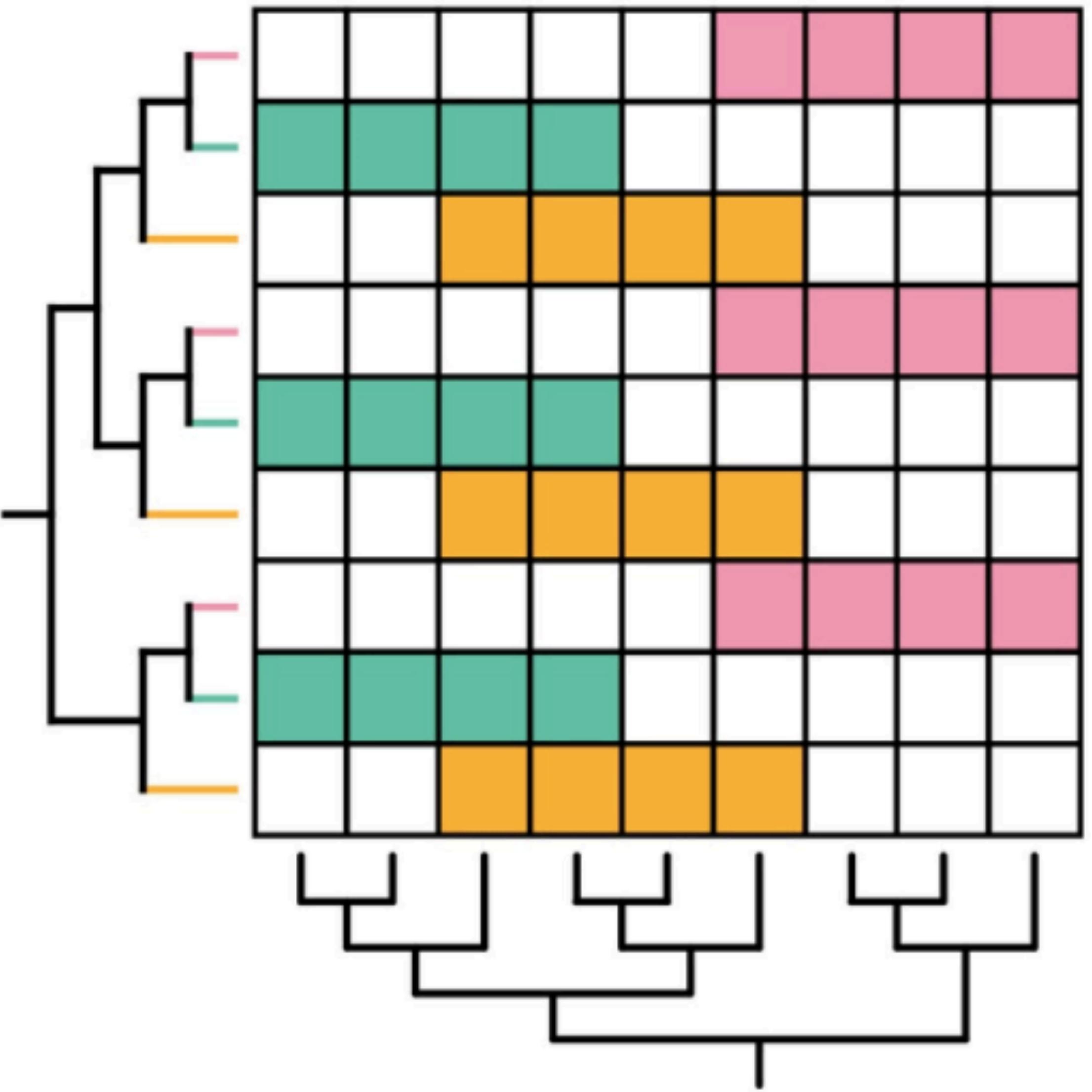
Begin data;
    Dimensions nchar=15;
    Format datatype=dna missing=? gap=- matchchar=.;
    Matrix
        [ When a position is a "matchchar", it means that it is the same as the first entry at the same position. ]
        SpaceDog    atgctagctagctcg
        SpaceCat    .......??....-.a.
        SpaceOrc    ...t.....-.g. [ same as atgttagctag-tgg ]
        SpaceElf    ...t.....-.a.

    ;
End;

BEGIN TREES;
    Tree tree1 = (((SpaceDog,SpaceCat),SpaceOrc,SpaceElf));
END;
```

اپردرخت

Supertree

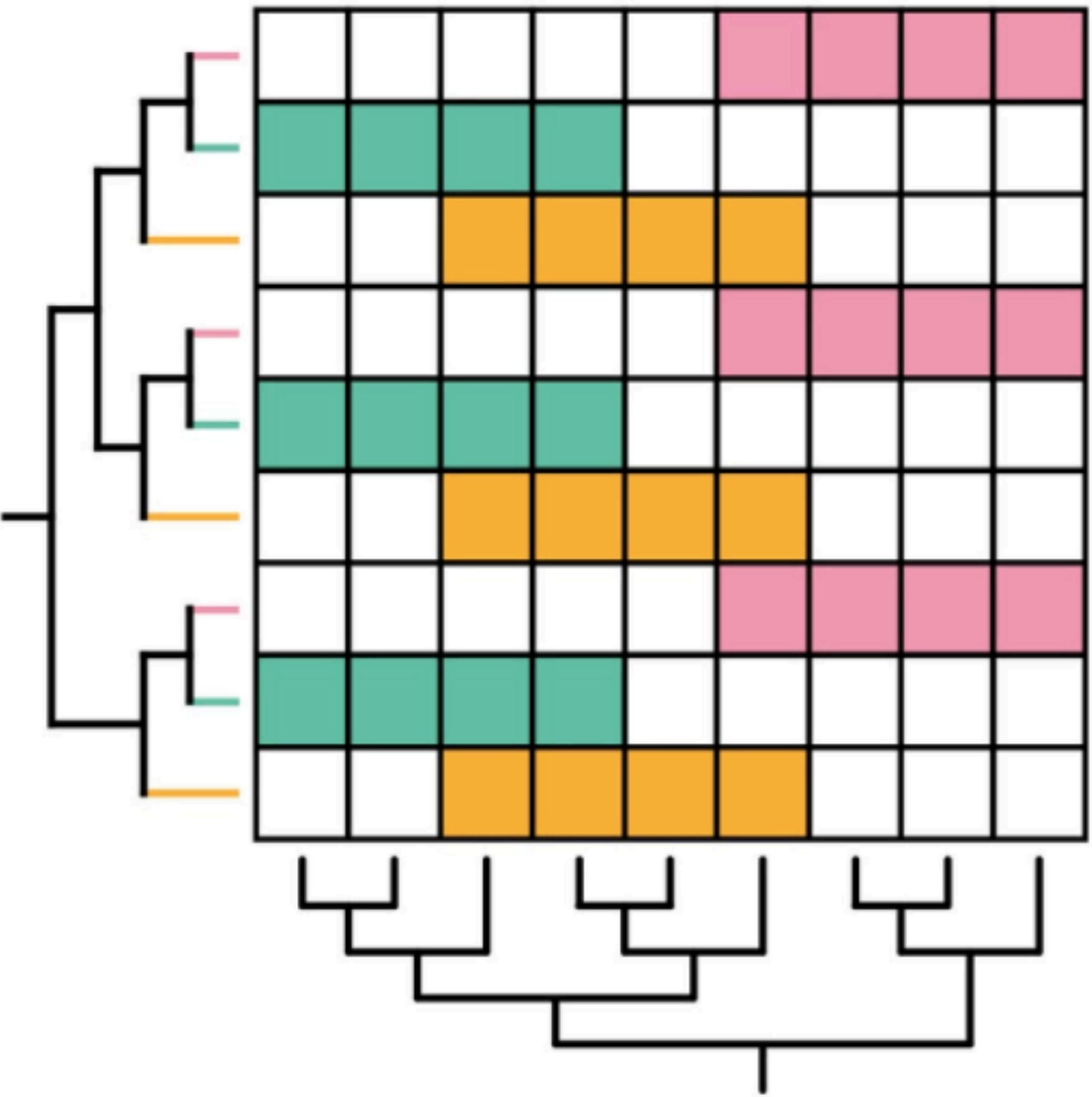


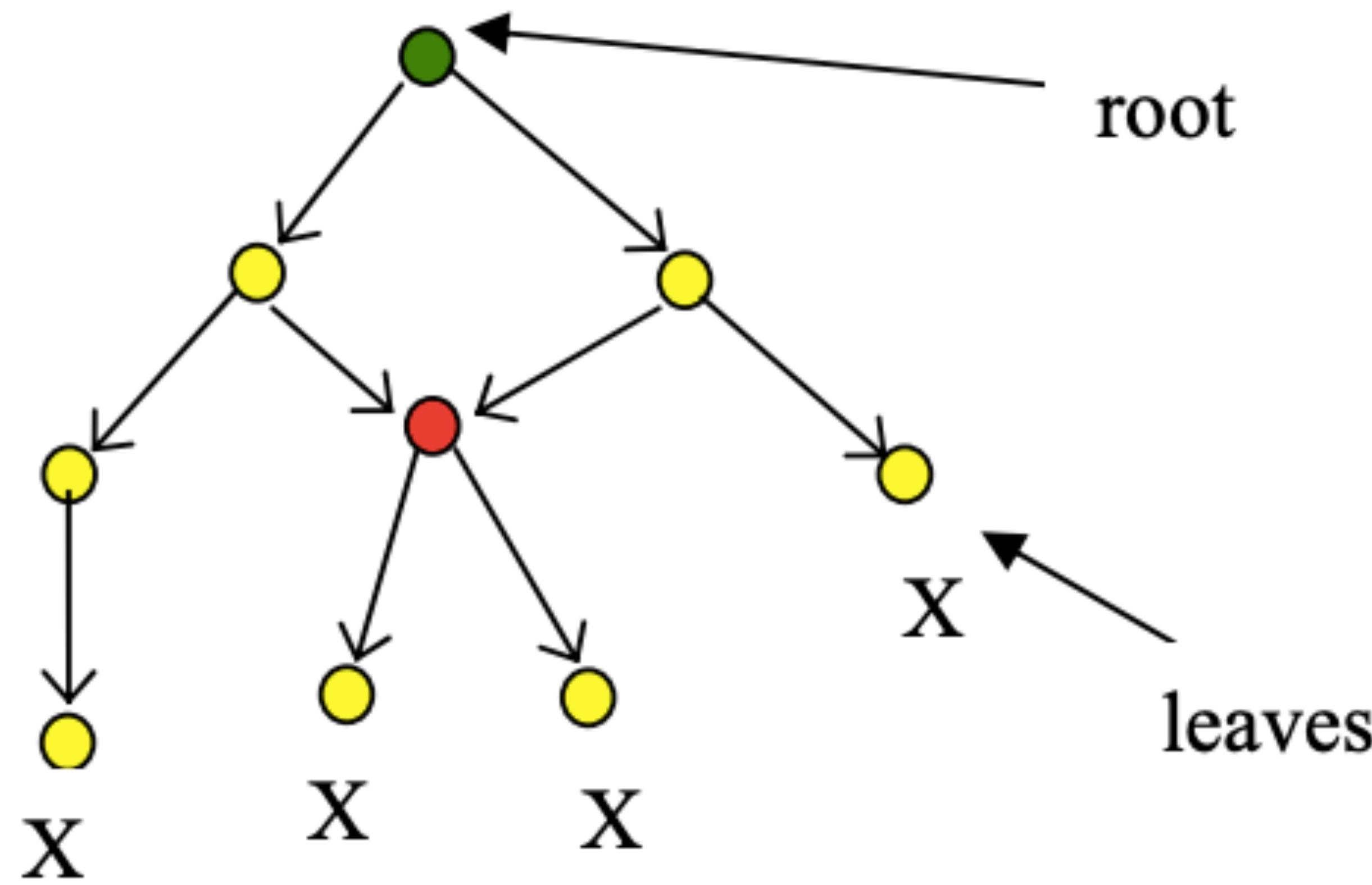
درخت‌هایی روی برخی از گونه‌ها

● چگونه درخت کامل را بسازیم؟

شبکه تبارزایی

Supertree





سؤال؟

