

Sur les arbres rouge noir : partie IV

Rappel :

La structure de données d'arbre rouge et noir ne possède pas l'inconvénient des arbres AVL.

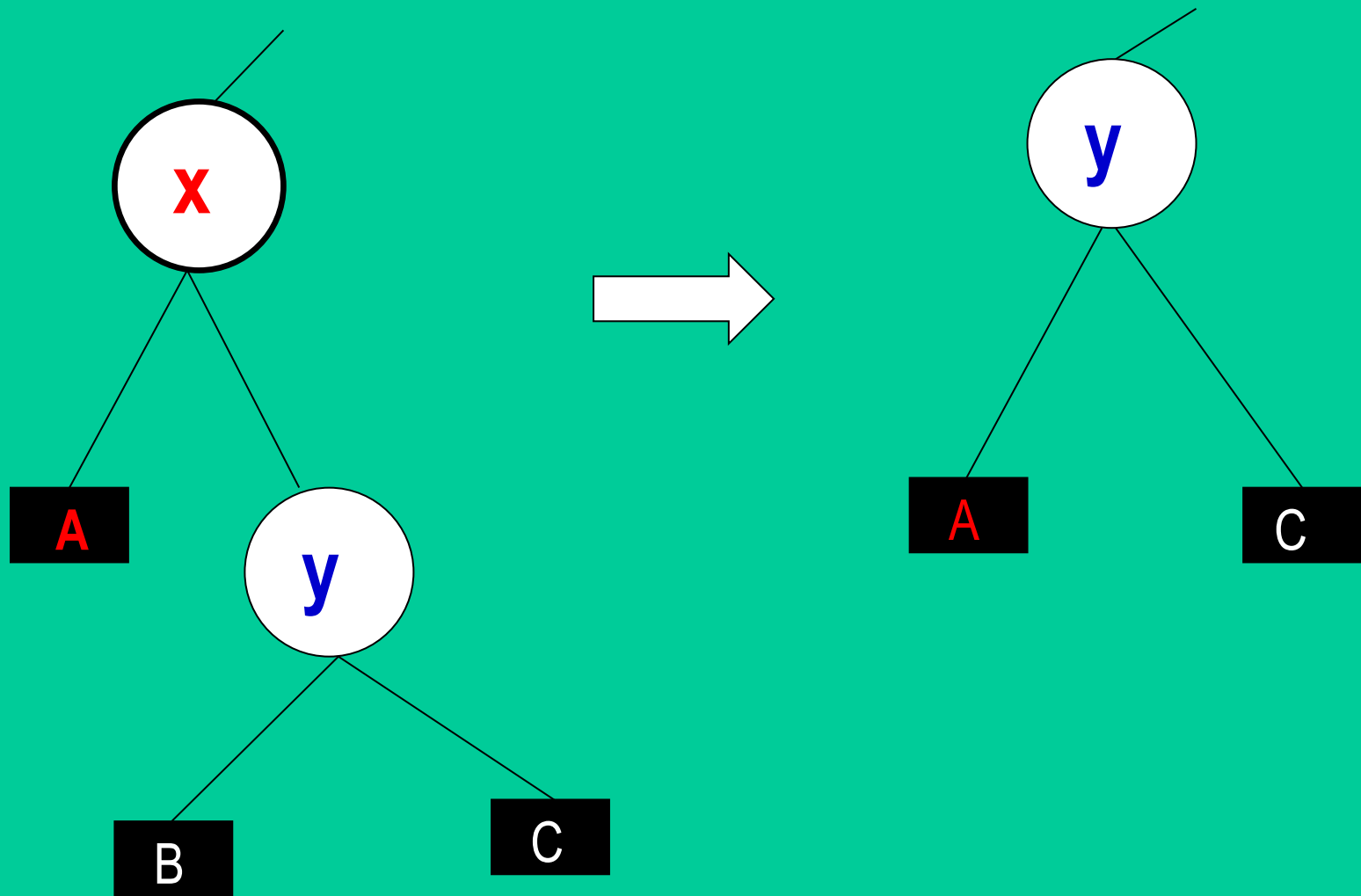
Elle ne nécessite qu'un **temps constant** après une mise à jour pour conserver sa propriété.

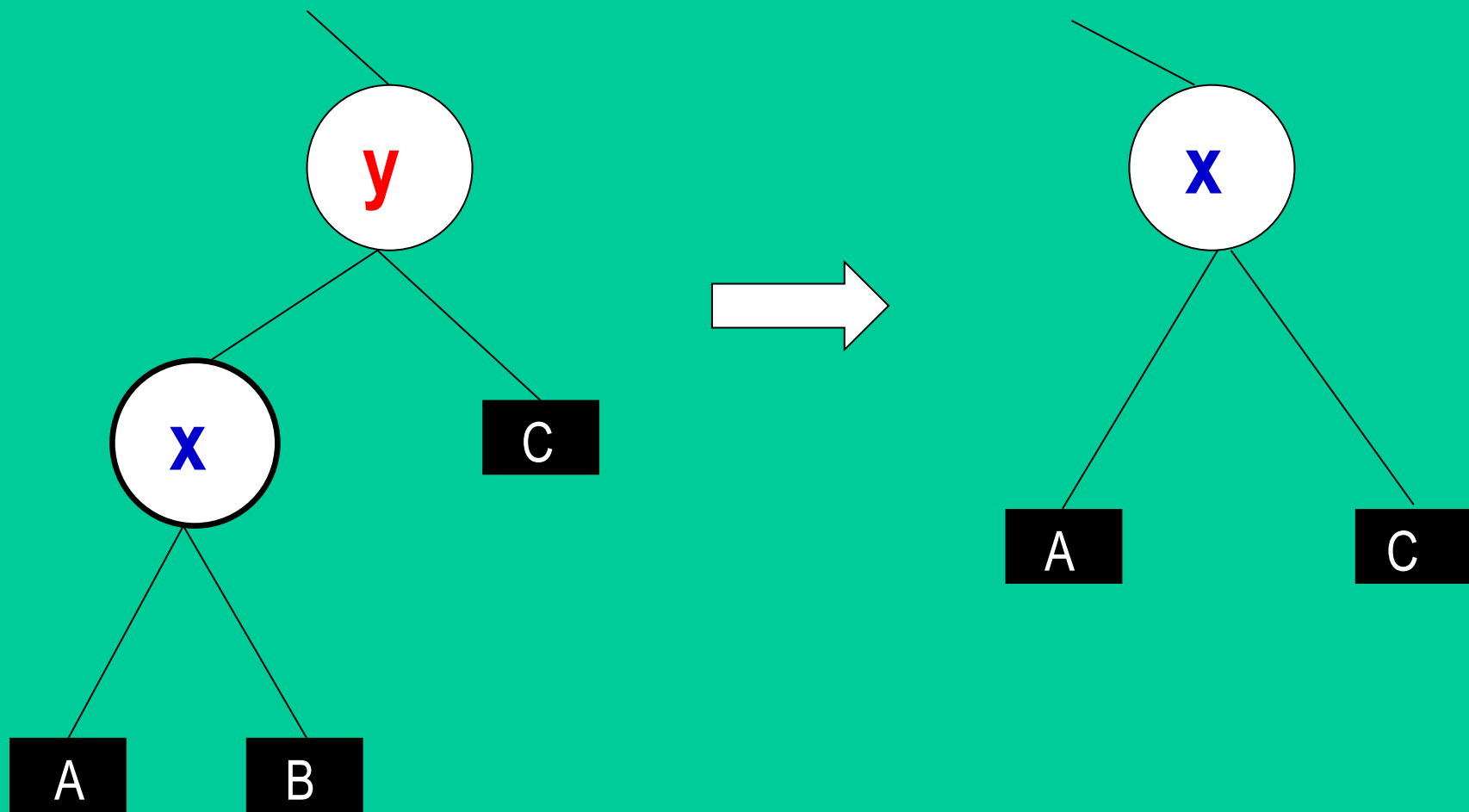
Principe:

La suppression dans un arbre rouge et noir distingue deux cas:

1-si le nœud à supprimer possède **zéro** ou **1** fils
alors :

- a) il **est supprimé**
- b) et son fils prend sa place.





2-si ce nœud possède 2 fils, alors :

a) il n'est **pas supprimé** ,

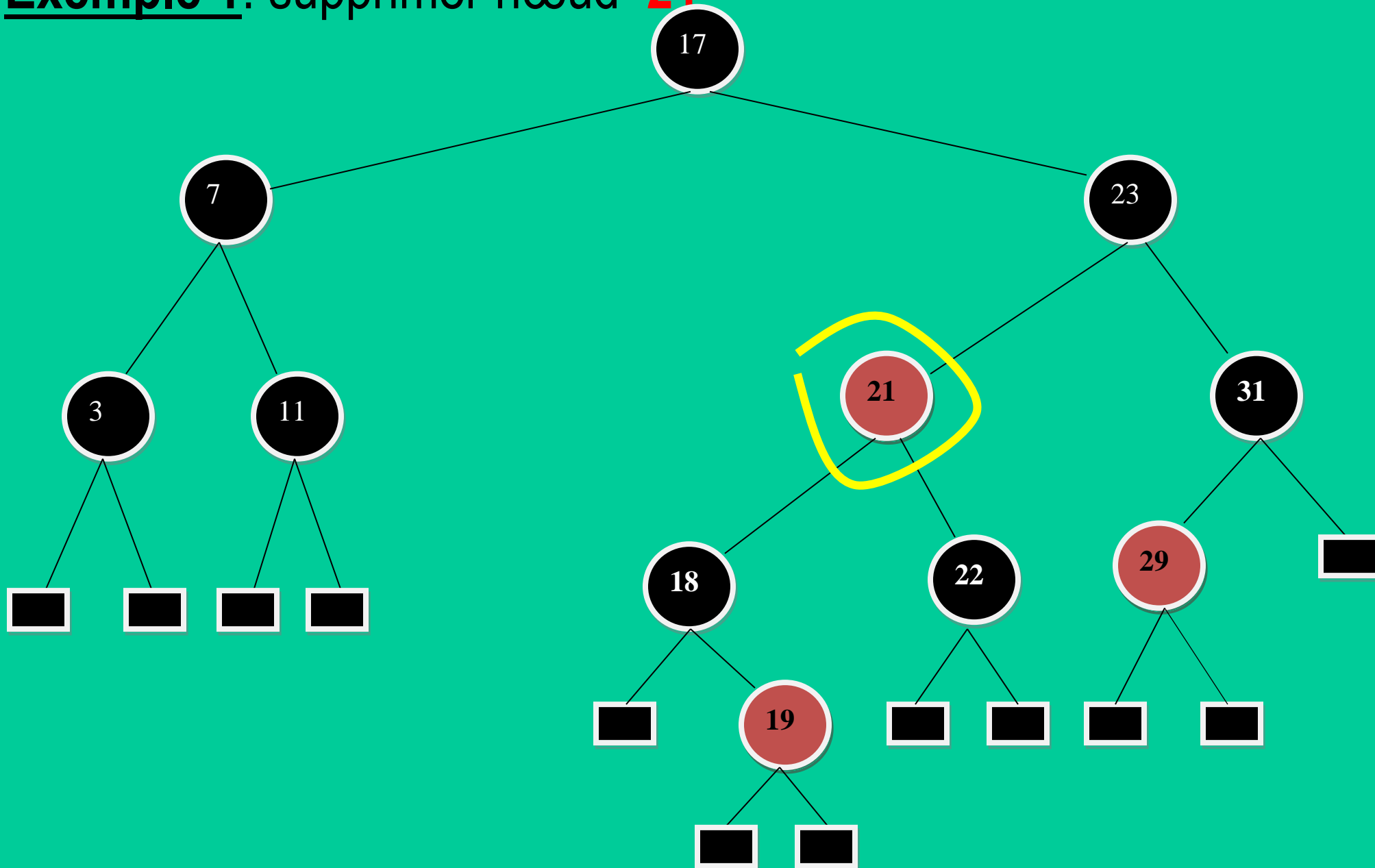
b) il portera la valeur précédente (ou suivante)
dans l'ordre,

c) le **nœud qui portait cette valeur est supprimé.**

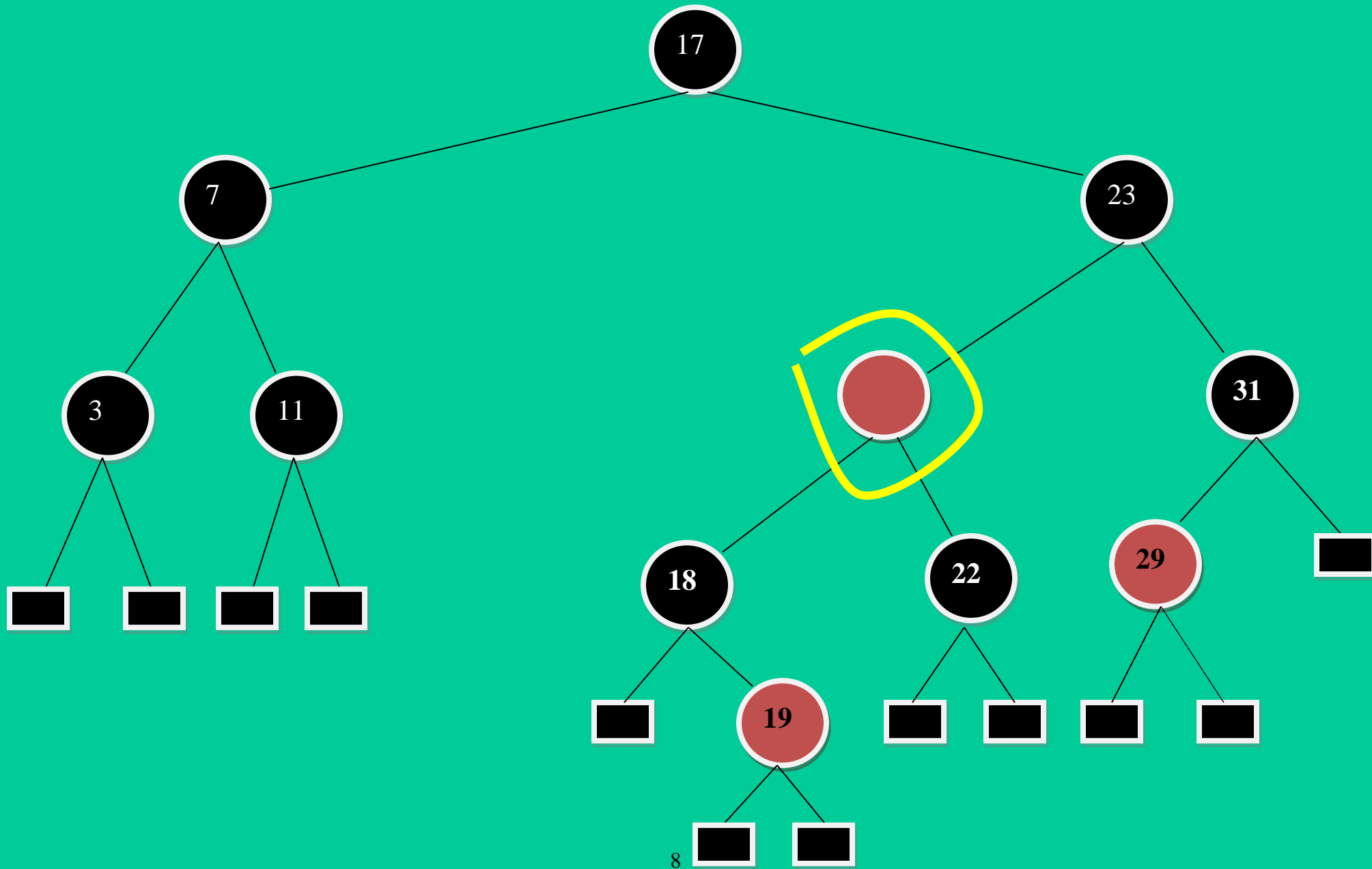
Hypothèse 1: le nœud à supprimer est **rouge**

- La propriété P5 reste vérifiée
- L'arbre demeure rouge noir.

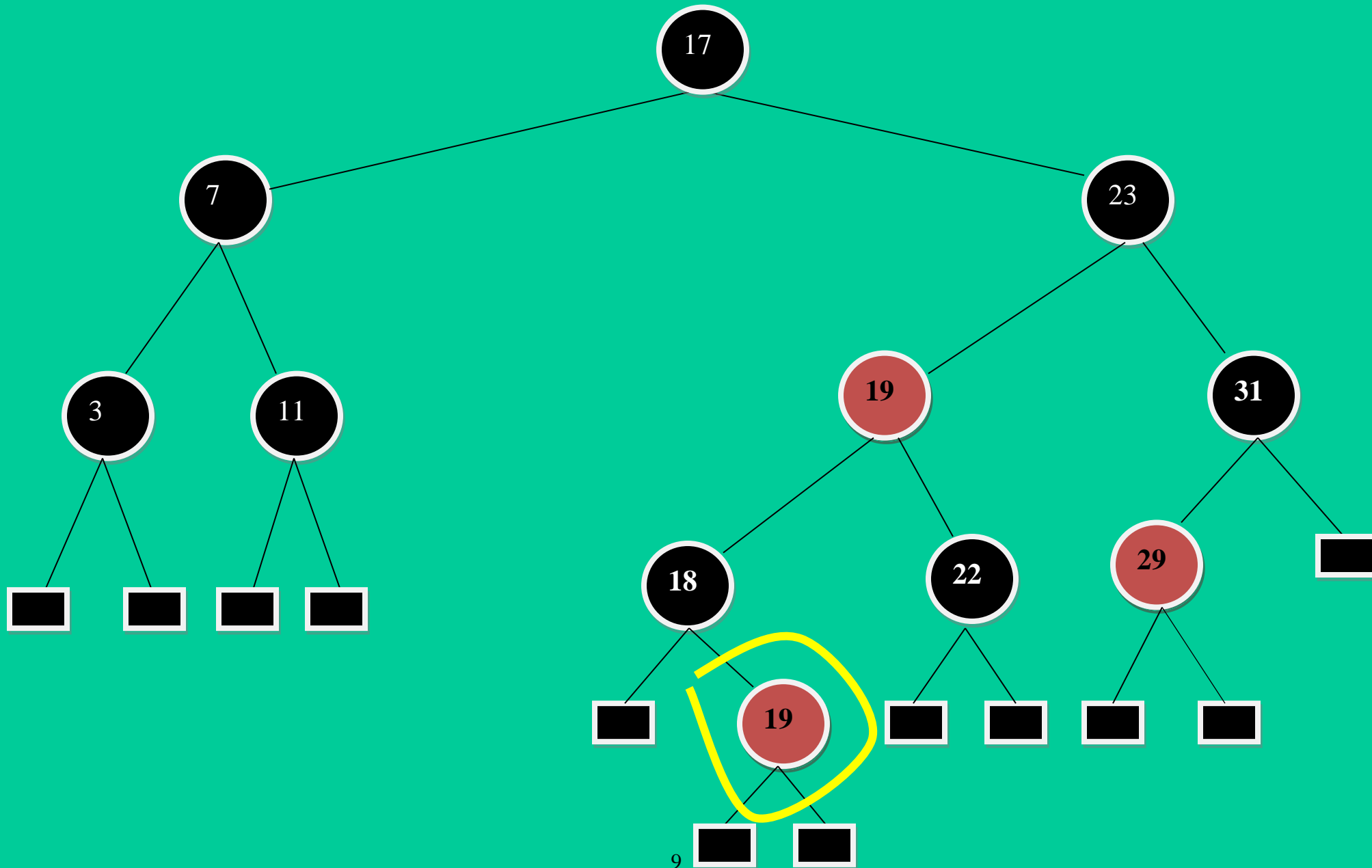
Exemple 1: supprimer nœud 21



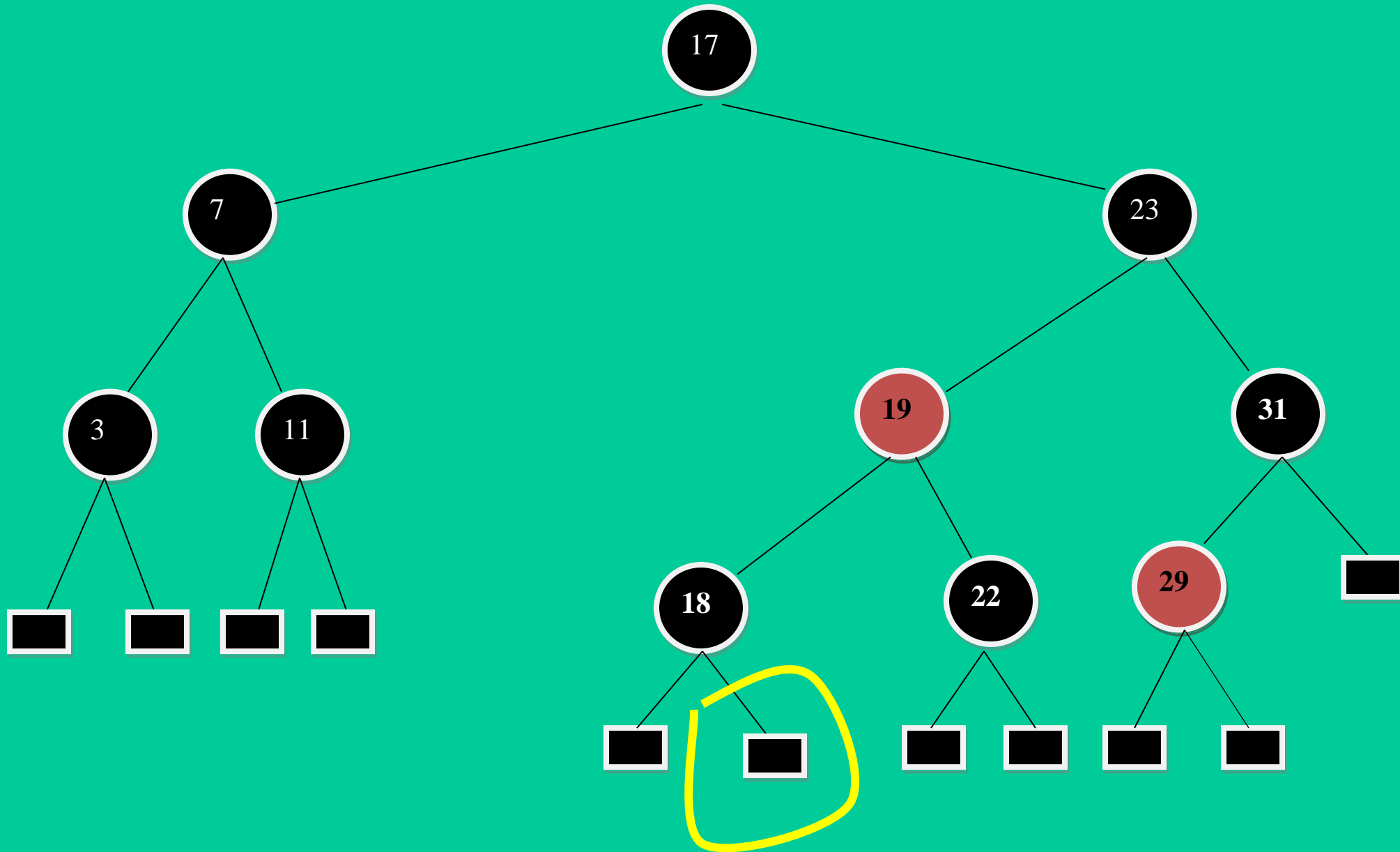
Acte 0: **21** ne sera pas supprimé !



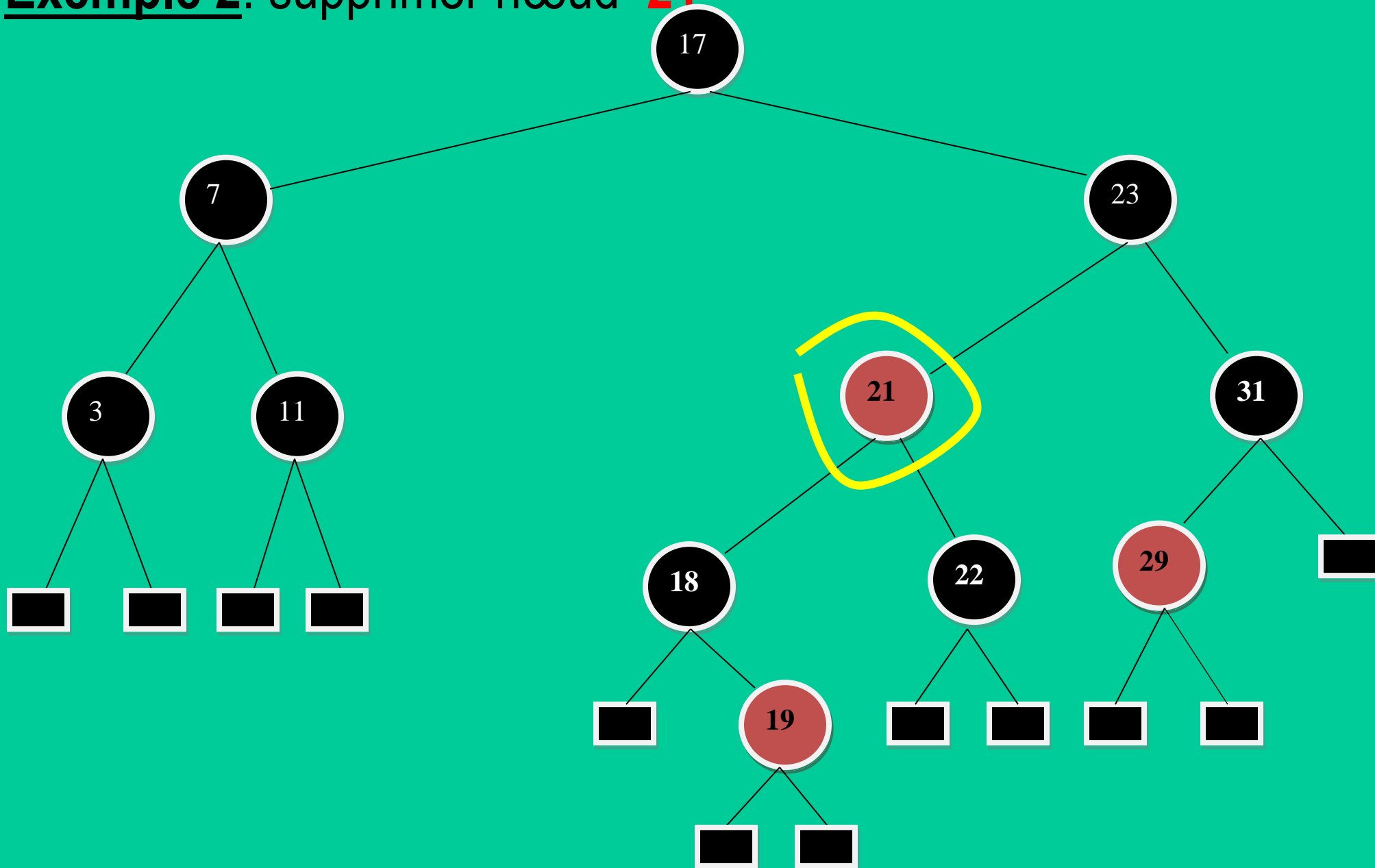
Acte 1: **21** est remplacé par **19** (Max à gauche)



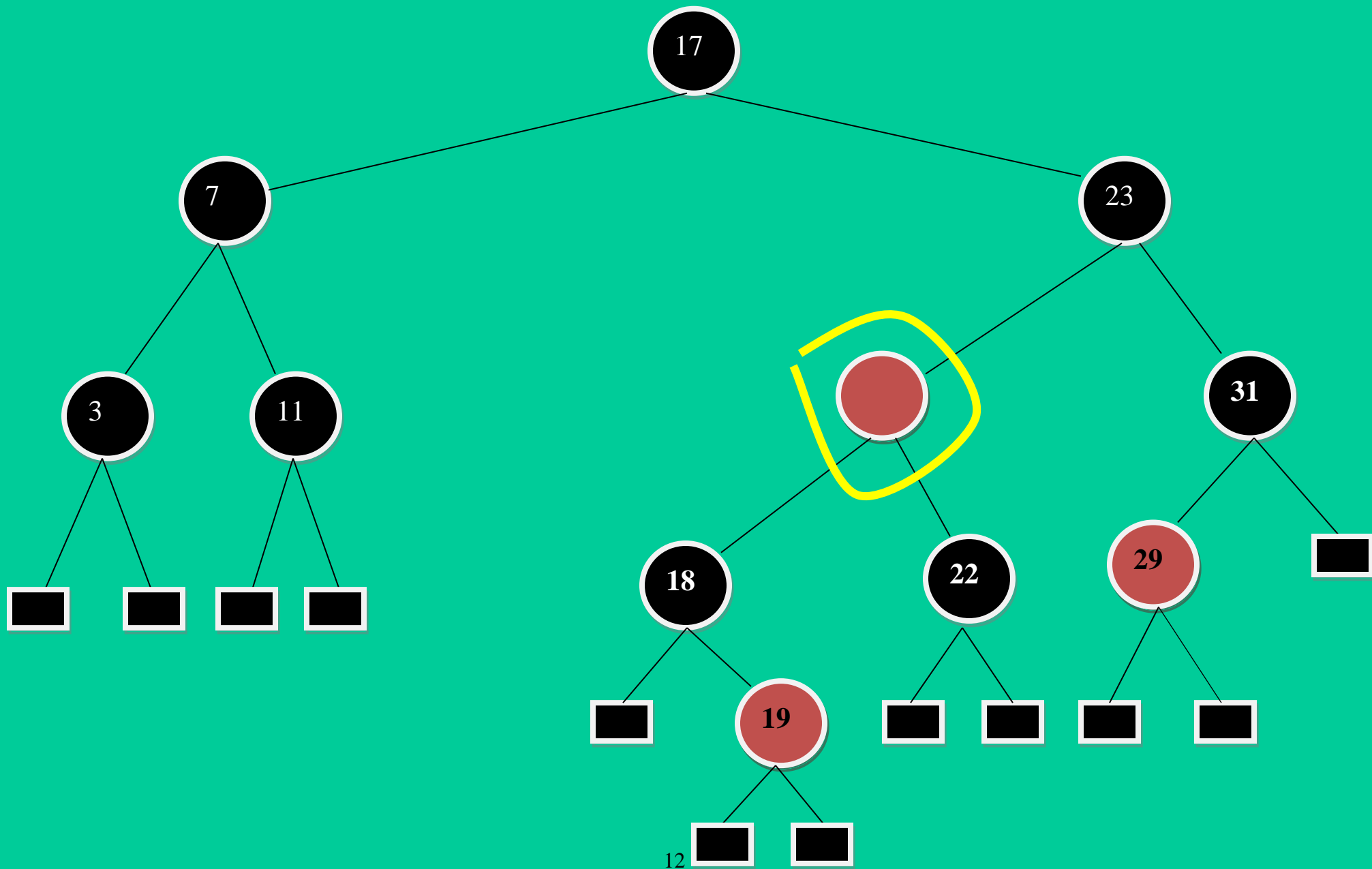
Acte 2 : le nœud **19** est supprimé



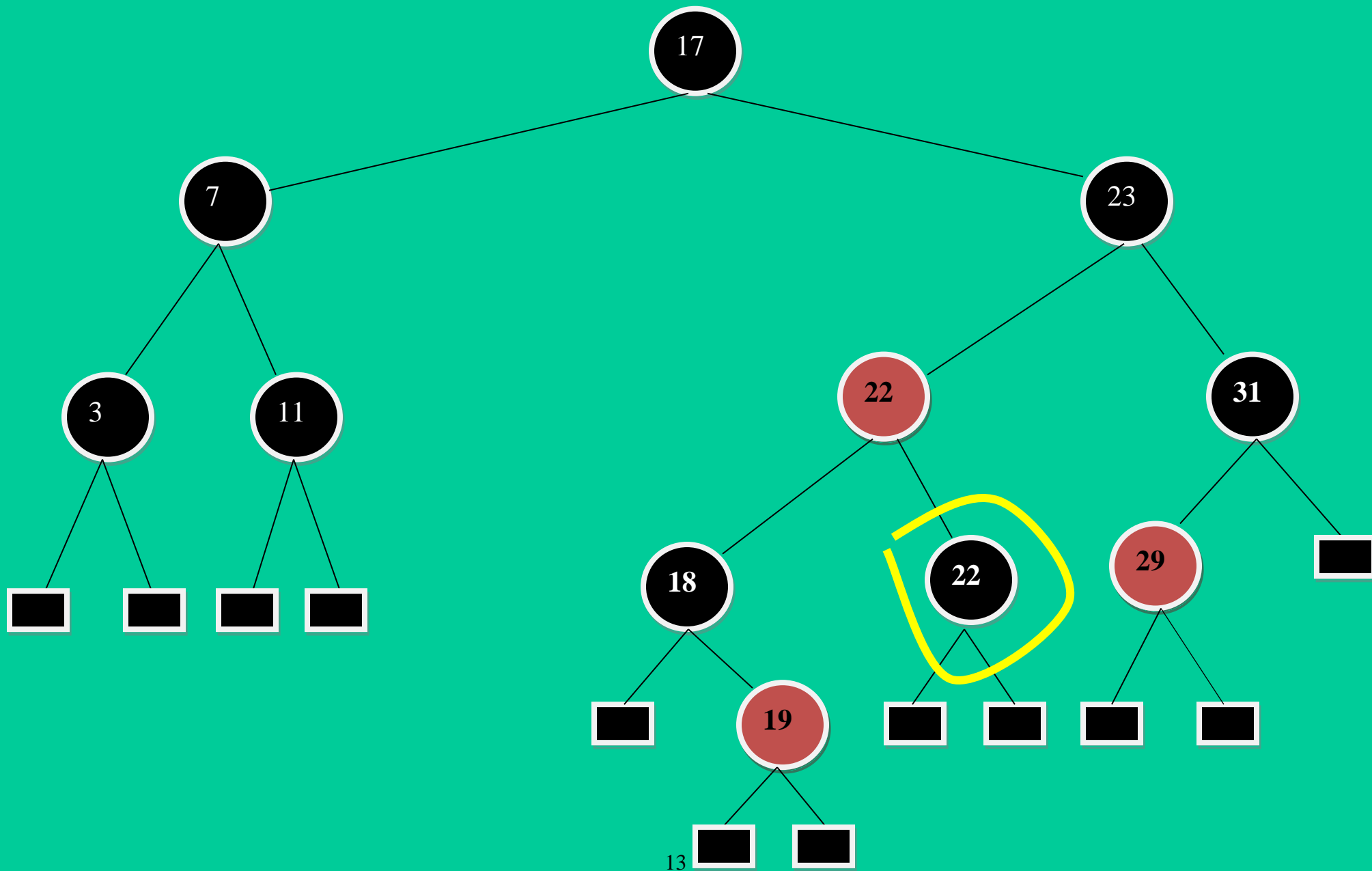
Exemple 2: supprimer nœud 21



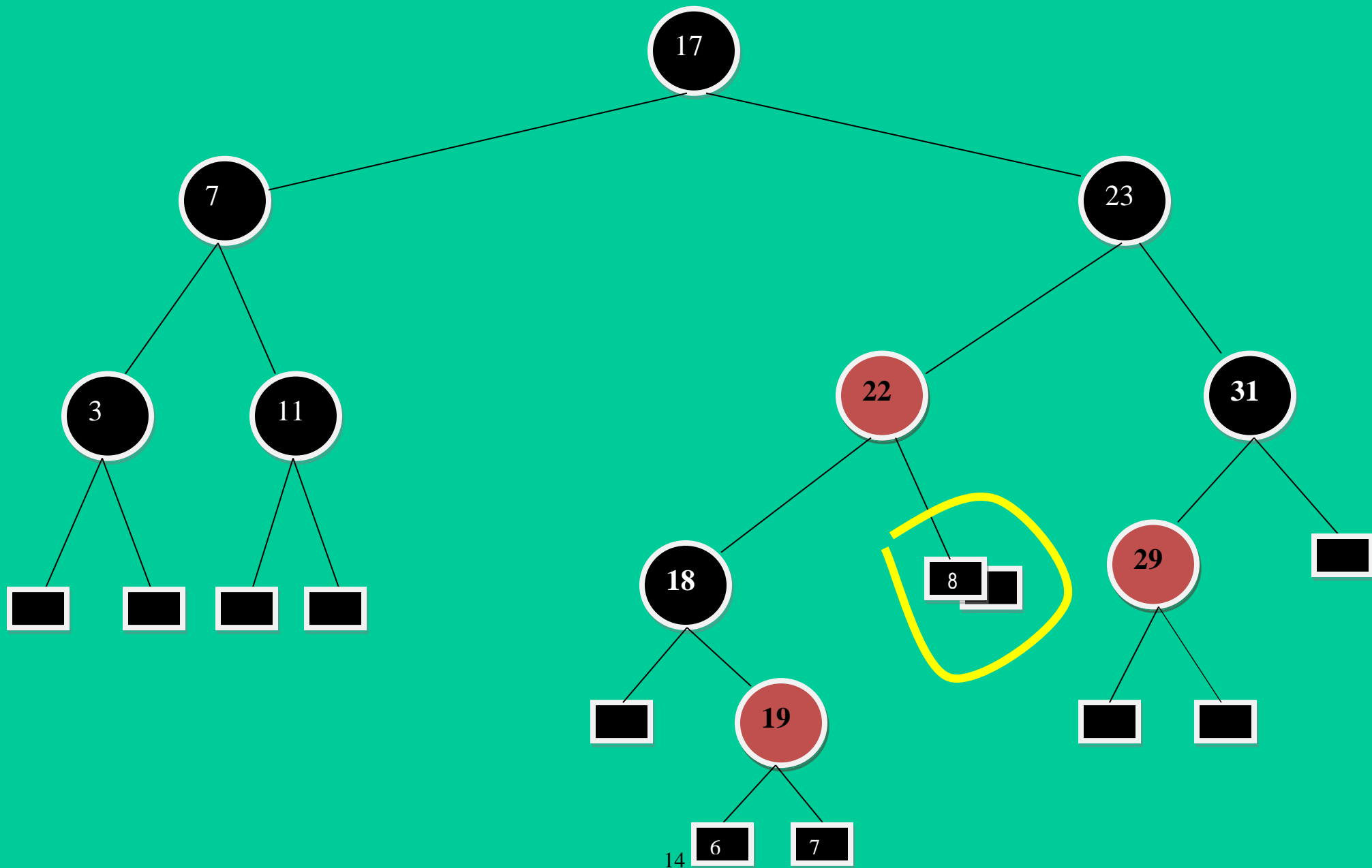
Acte 0: **21** ne sera pas supprimé !



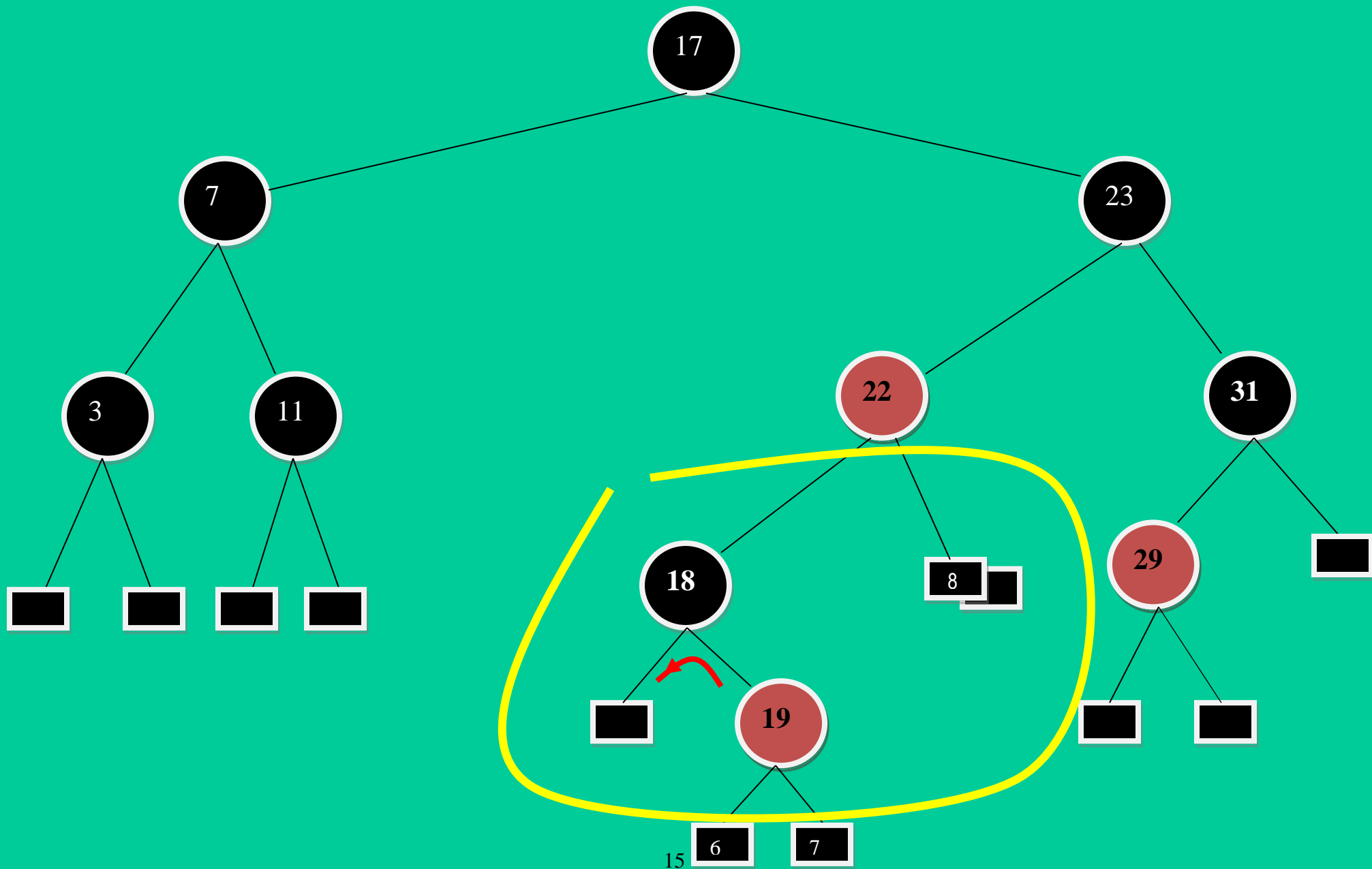
Acte 1: **21** est remplacé par **22** (Min à droite)



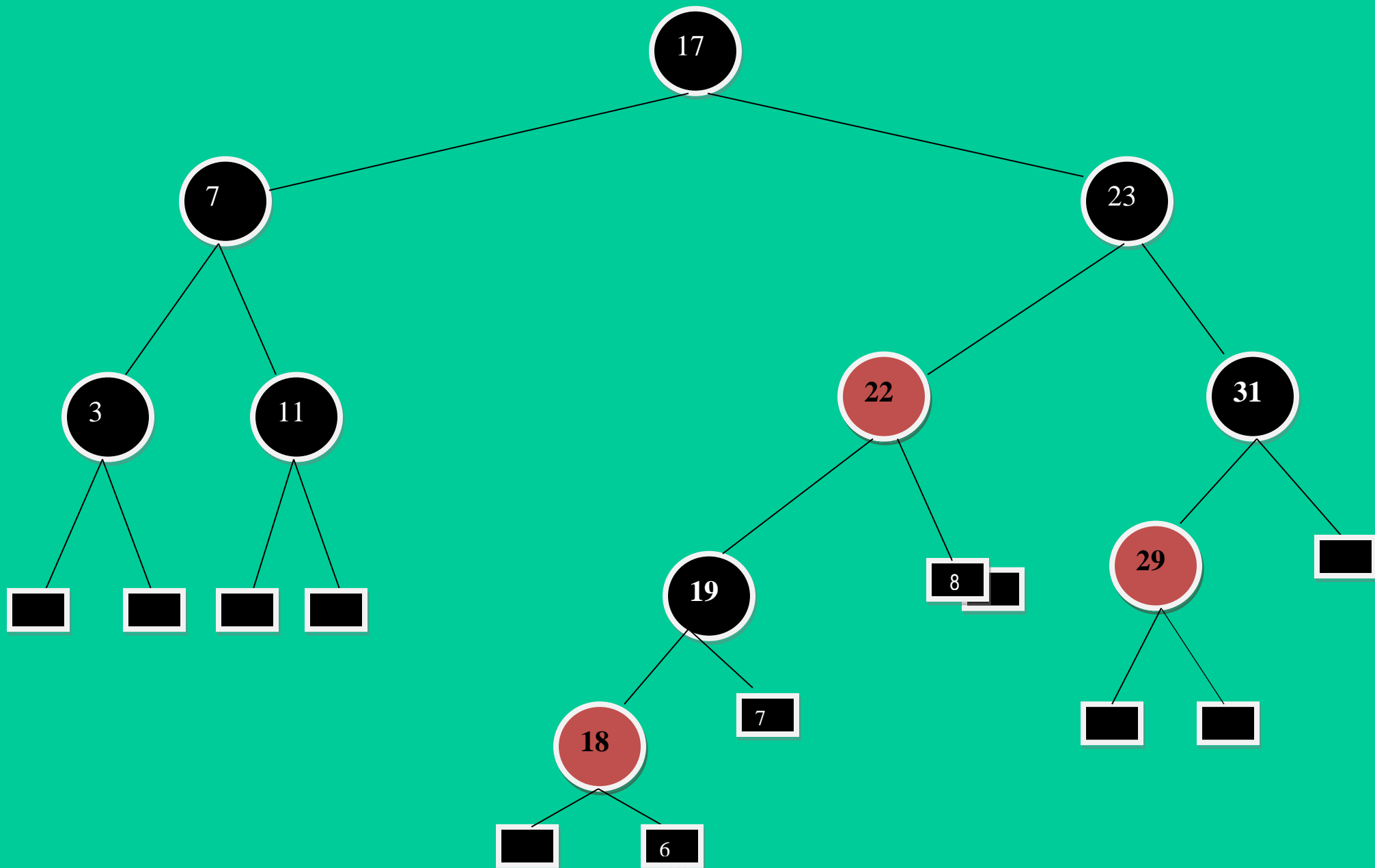
Acte 2: supprimer **22**, la sentinelle **8** le remplace en **noir**²



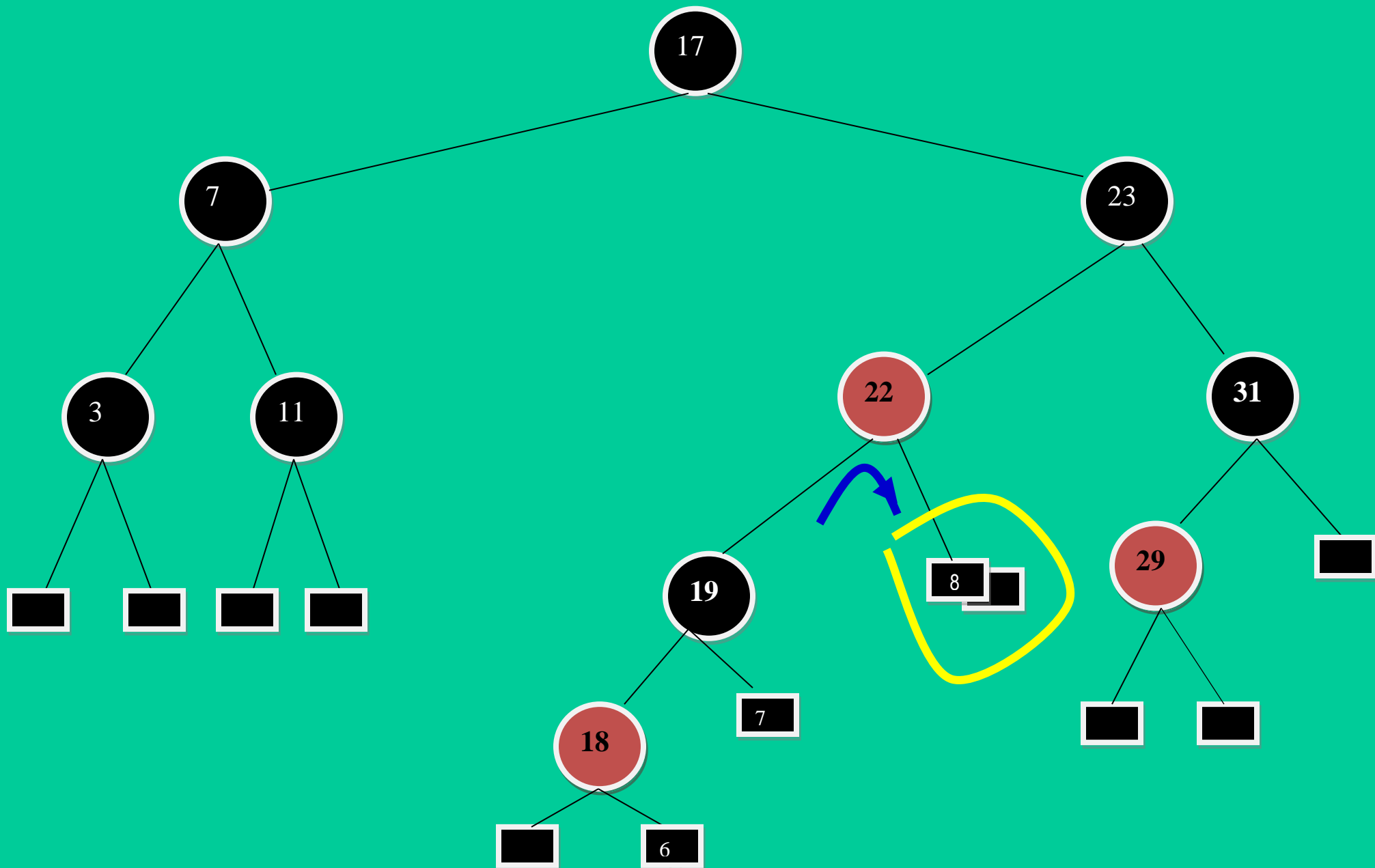
Acte 3: $rg(18)$ + changer couleur 18 et 19



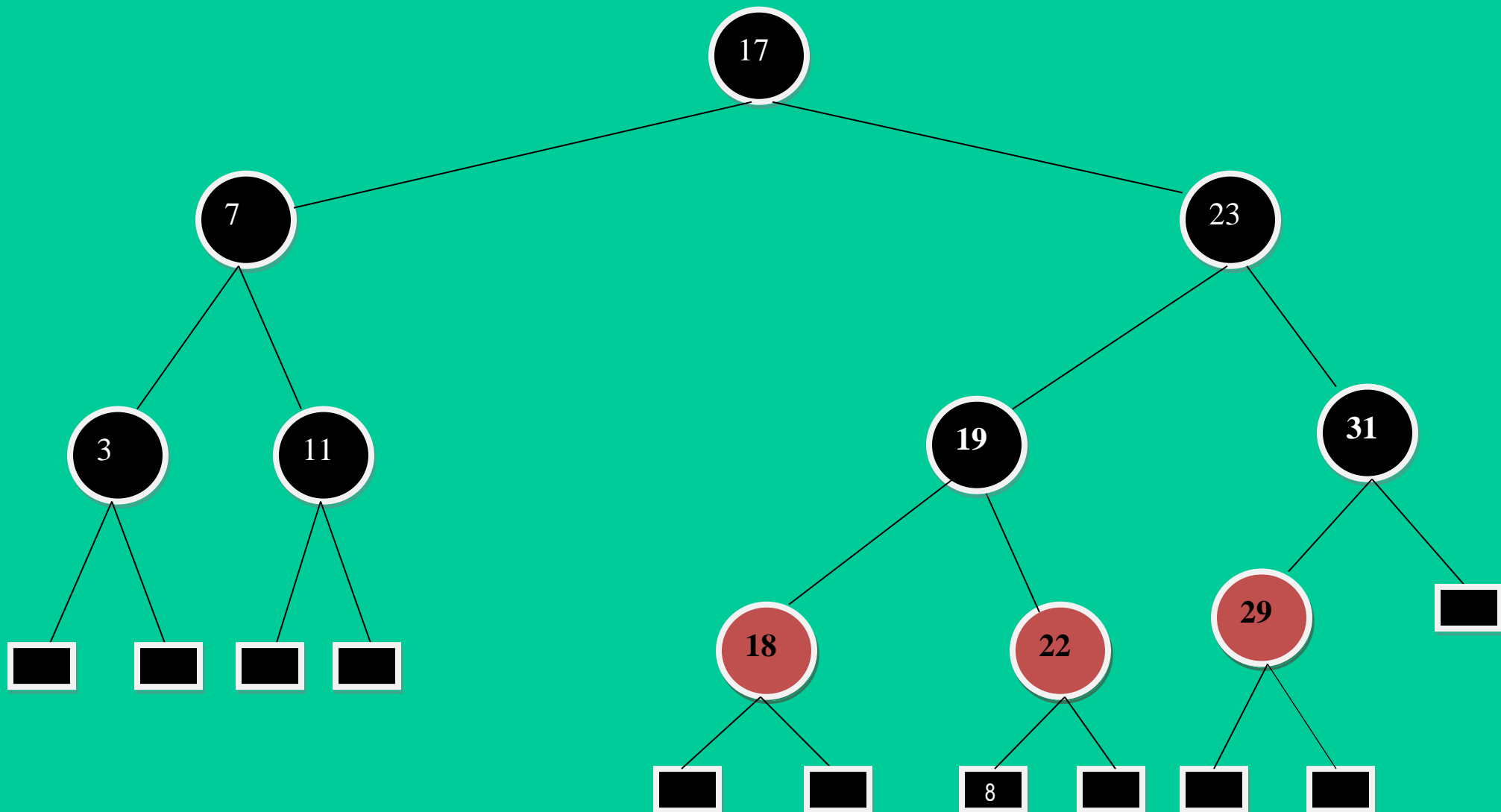
Après acte 3: $rg(18)$ + changer couleur 18 et 19



Acte 4: rd(**22**)



Après acte 4: la sentinelle 8 redevient **noir**



Hypothèse 2: le nœud à supprimer est noir

La suppression du nœud va **diminuer de 1** la hauteur noire de ses ascendants.

La propriété P5 est recouvrée en rectifiant les couleurs comme suit :

« Le nœud qui **remplace** le nœud noir à supprimer doit porter la couleur **noire⁺**. »

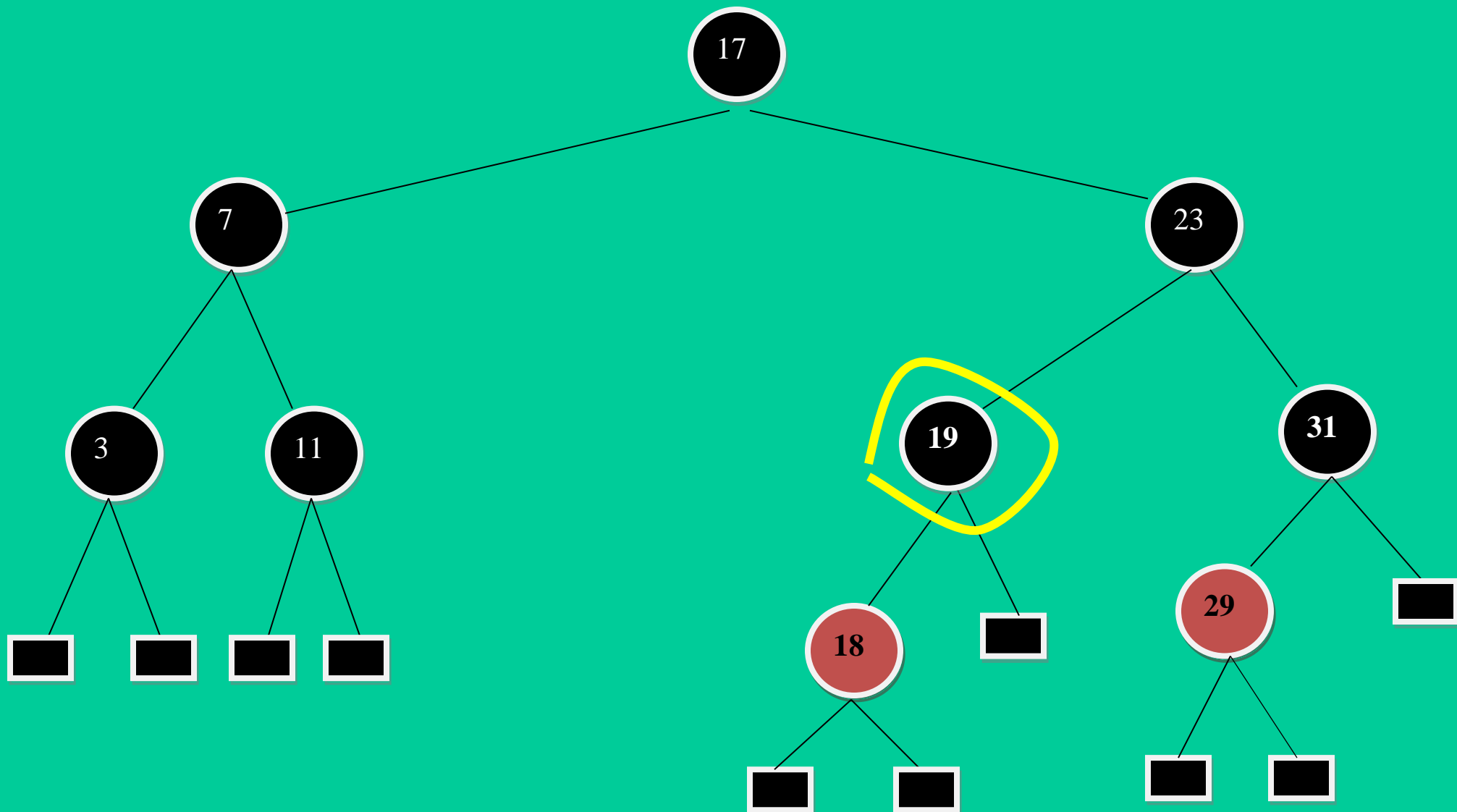
Cela signifie qu'il devient :

- **noir**, s'il était **rouge**,
- **noir²** : «doublement noir», s'il était déjà **noir**

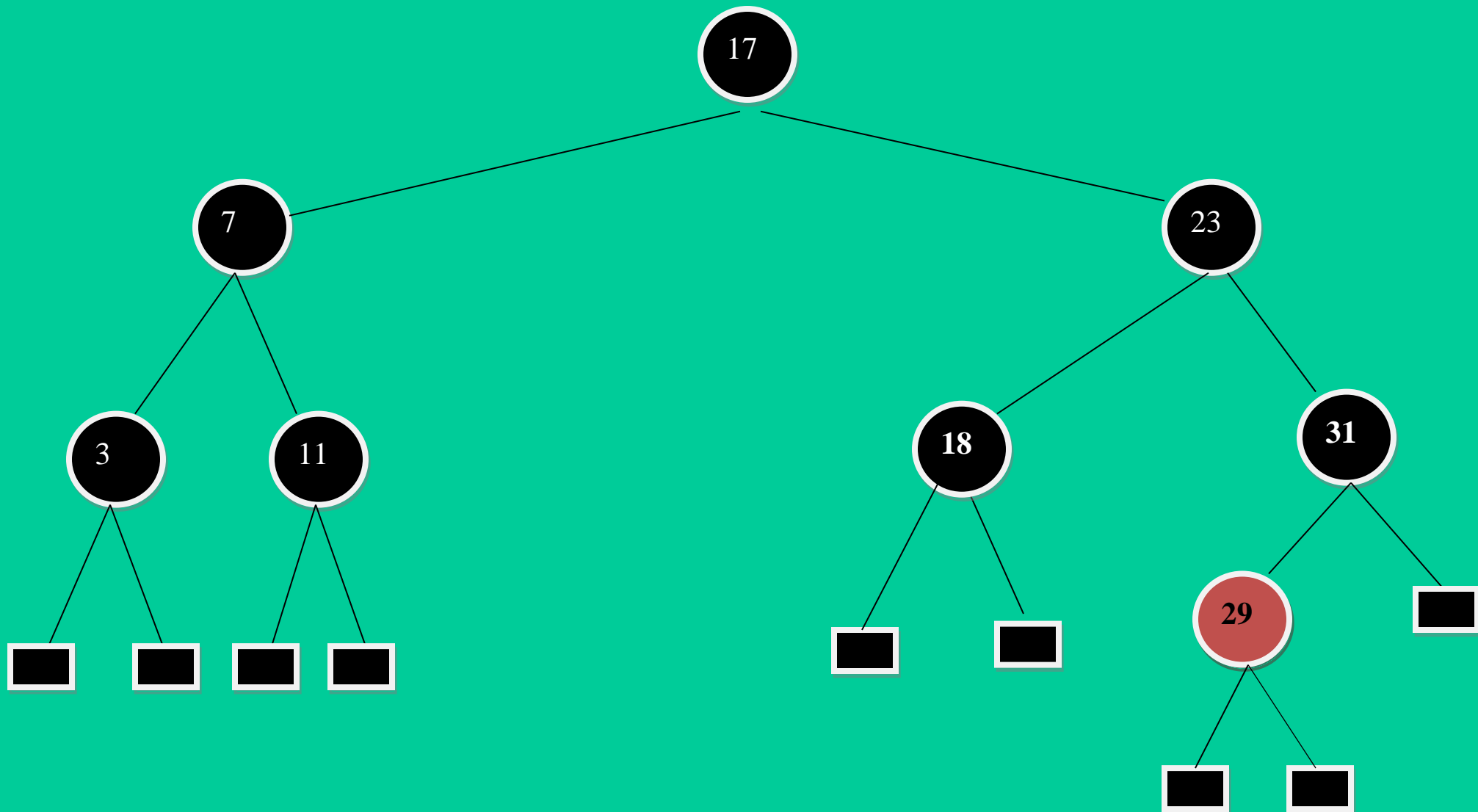
La propriété P5 reste ainsi vérifiée.

Mais il y a éventuellement un nœud qui est noir².

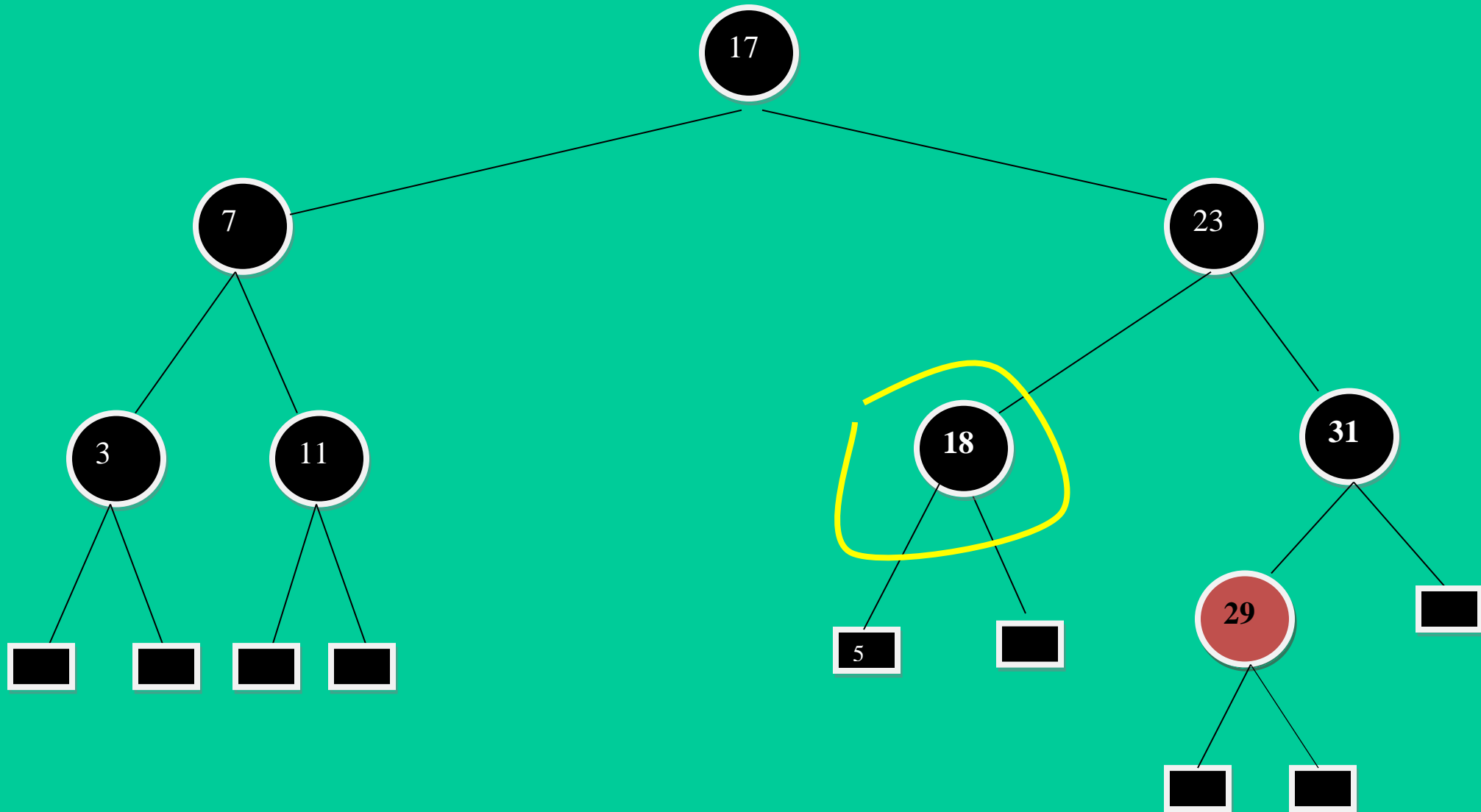
Exemple 1: supprimer 19



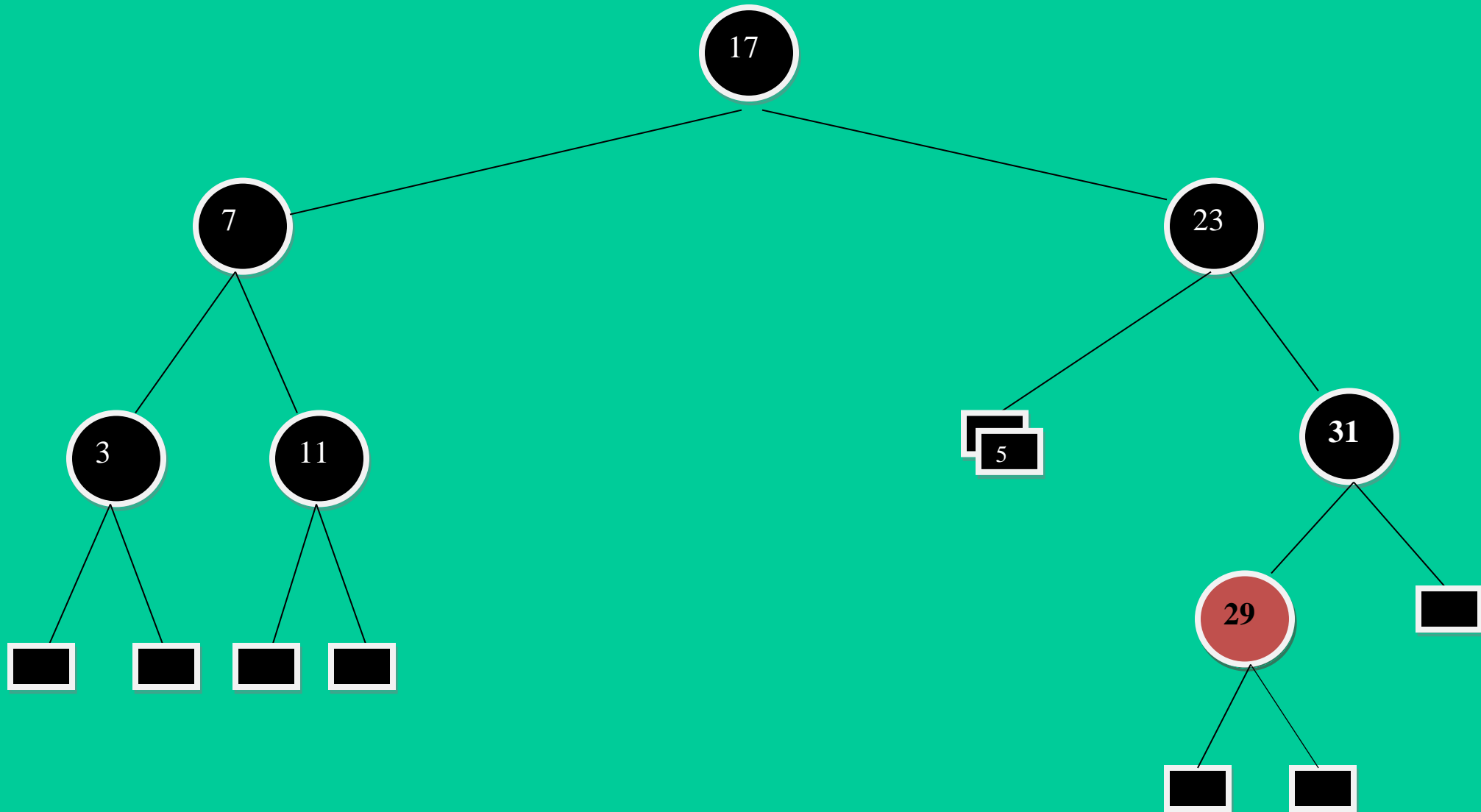
Acte 1: le **18** remplace le **19** et devient **noir**⁺



Exemple 2: supprimer le nœud 18



Acte 1: la sentinelle **5** remplace le **18** supprimé et devient **noir**²



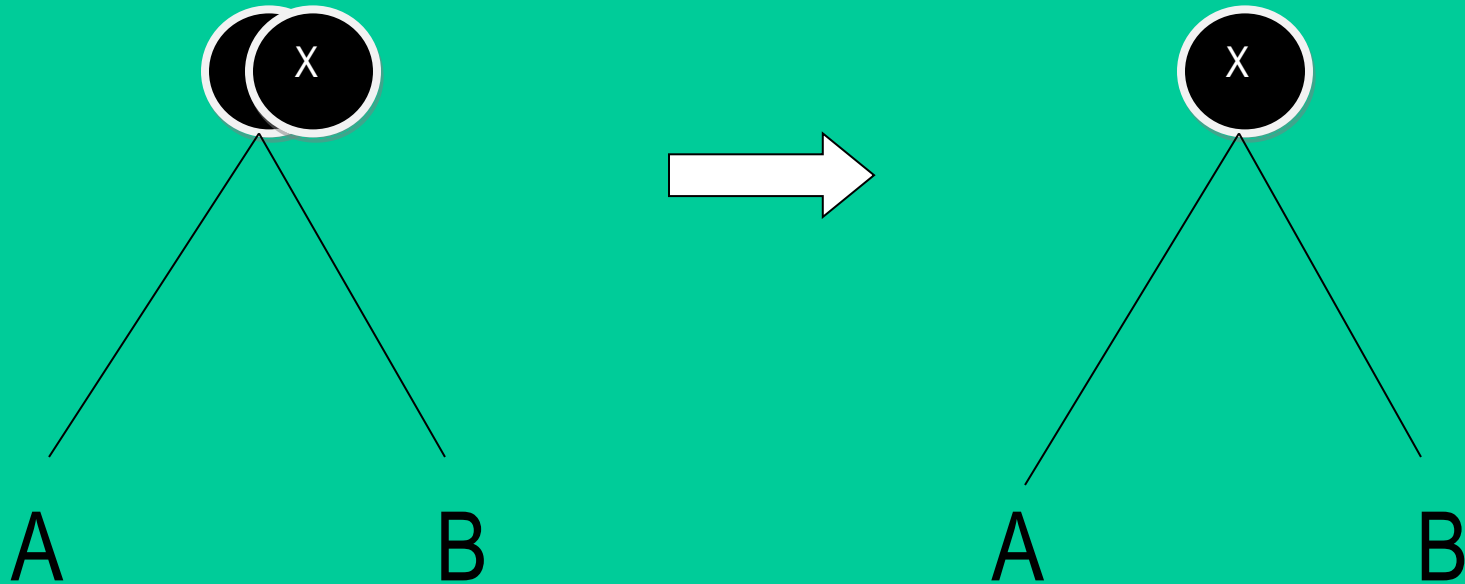
Le problème:

Comment supprimer la propriété noir² ?

L'algorithme effectue des modifications dans l'arbre à l'aide de **rotations**.

Soit **x** le nœud **noir²**.

Cas 0 : x est la racine de l'arbre



Le nœud **x** devient simplement noir.

La propriété P4 est maintenant vérifiée.

La propriété P5 le reste.

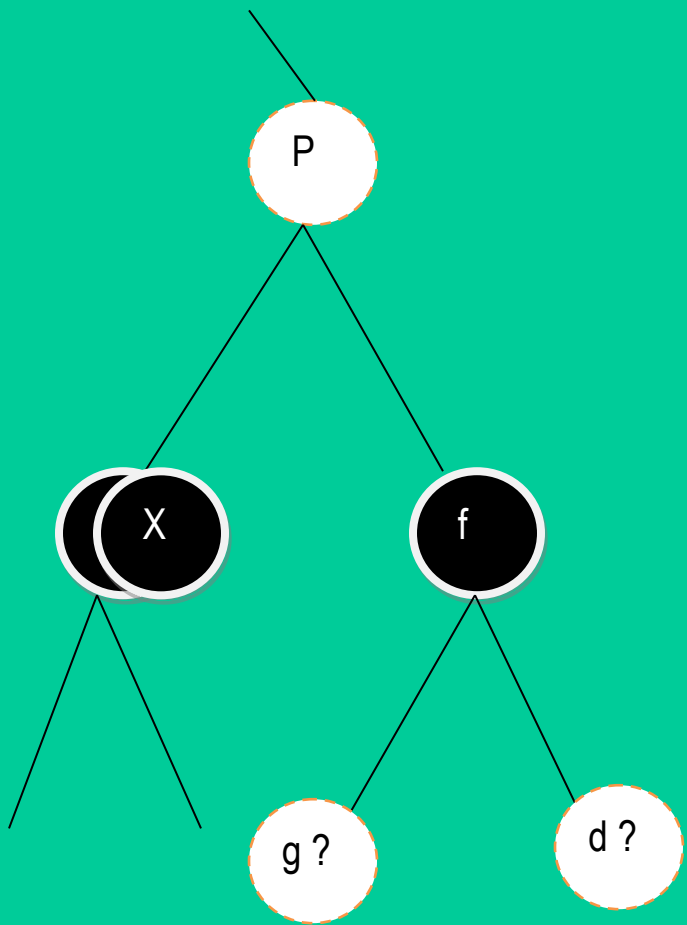
C'est le seul cas où la hauteur noire de l'arbre **diminue**.

Cas 1 : **x** n'est pas racine de l'arbre
le frère **f** de **x** est **noir**.

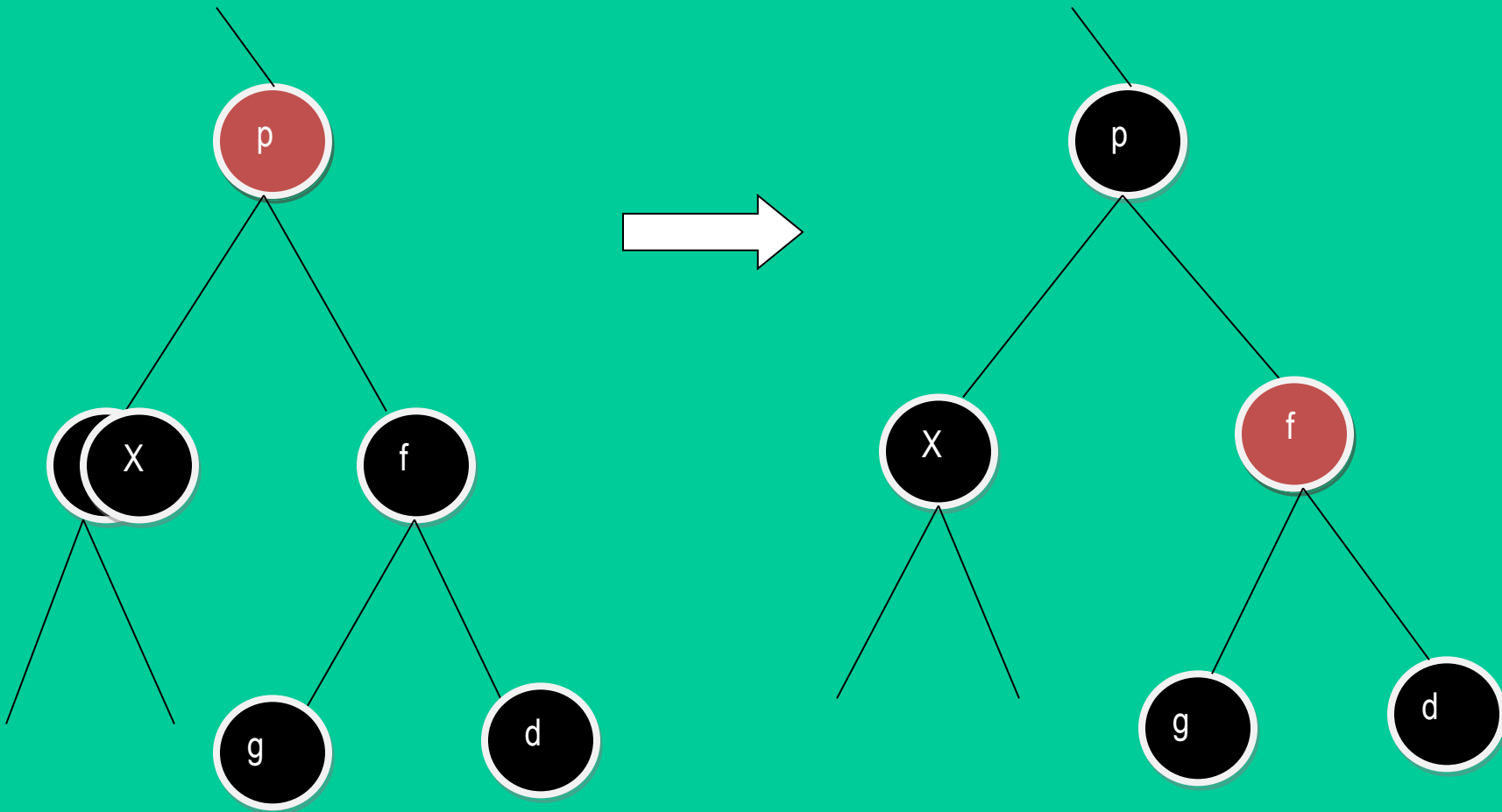
On suppose que **x** est le fils gauche de **p**: **f** est donc le fils droit de **p**.

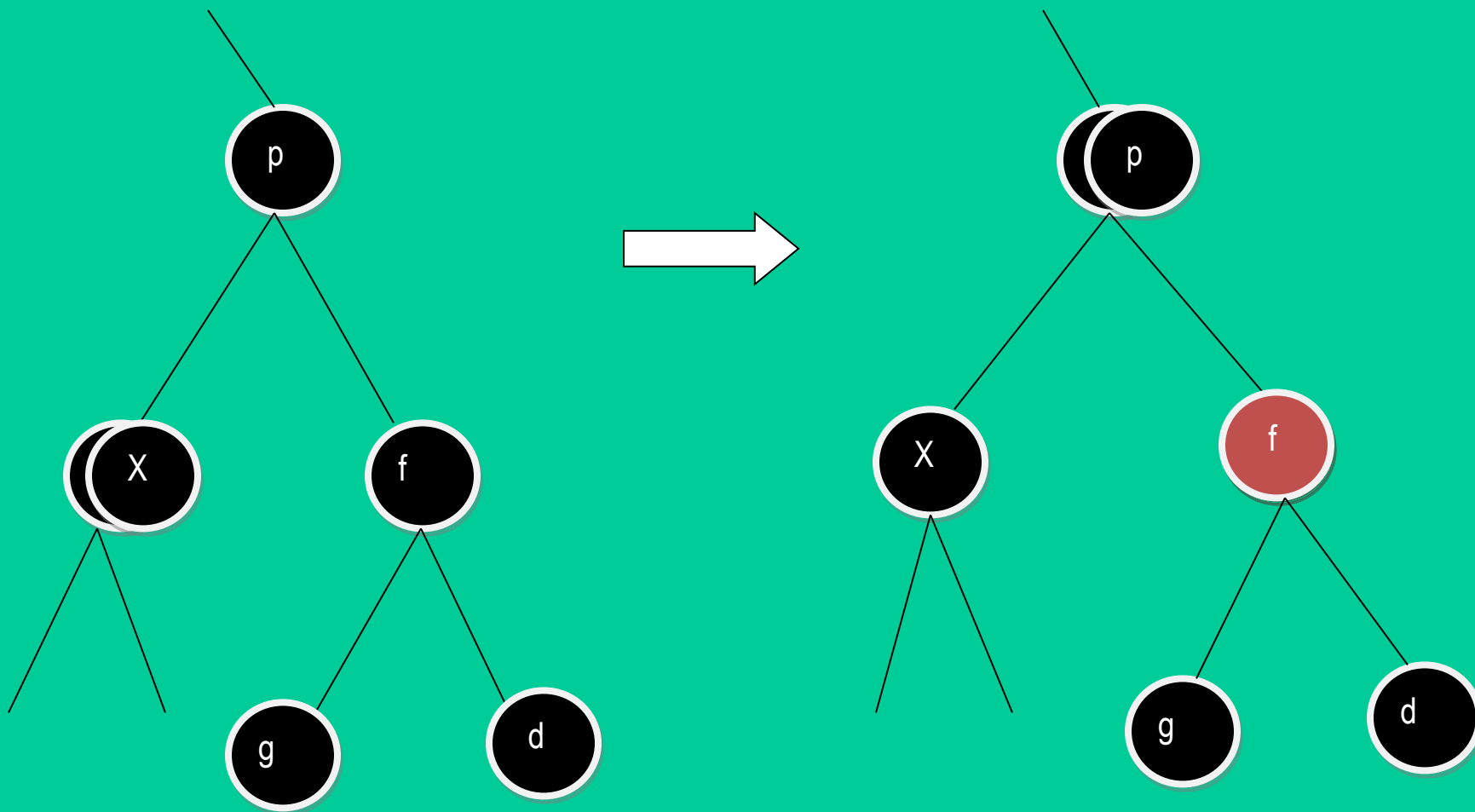
Soient **g** et **d** les fils gauche et droit de **f**.

L'algorithme distingue 3 cas suivant les couleurs des nœuds **g** et **d**.



Cas 1a : les deux fils **g** et **d** de **f** sont **noirs**.



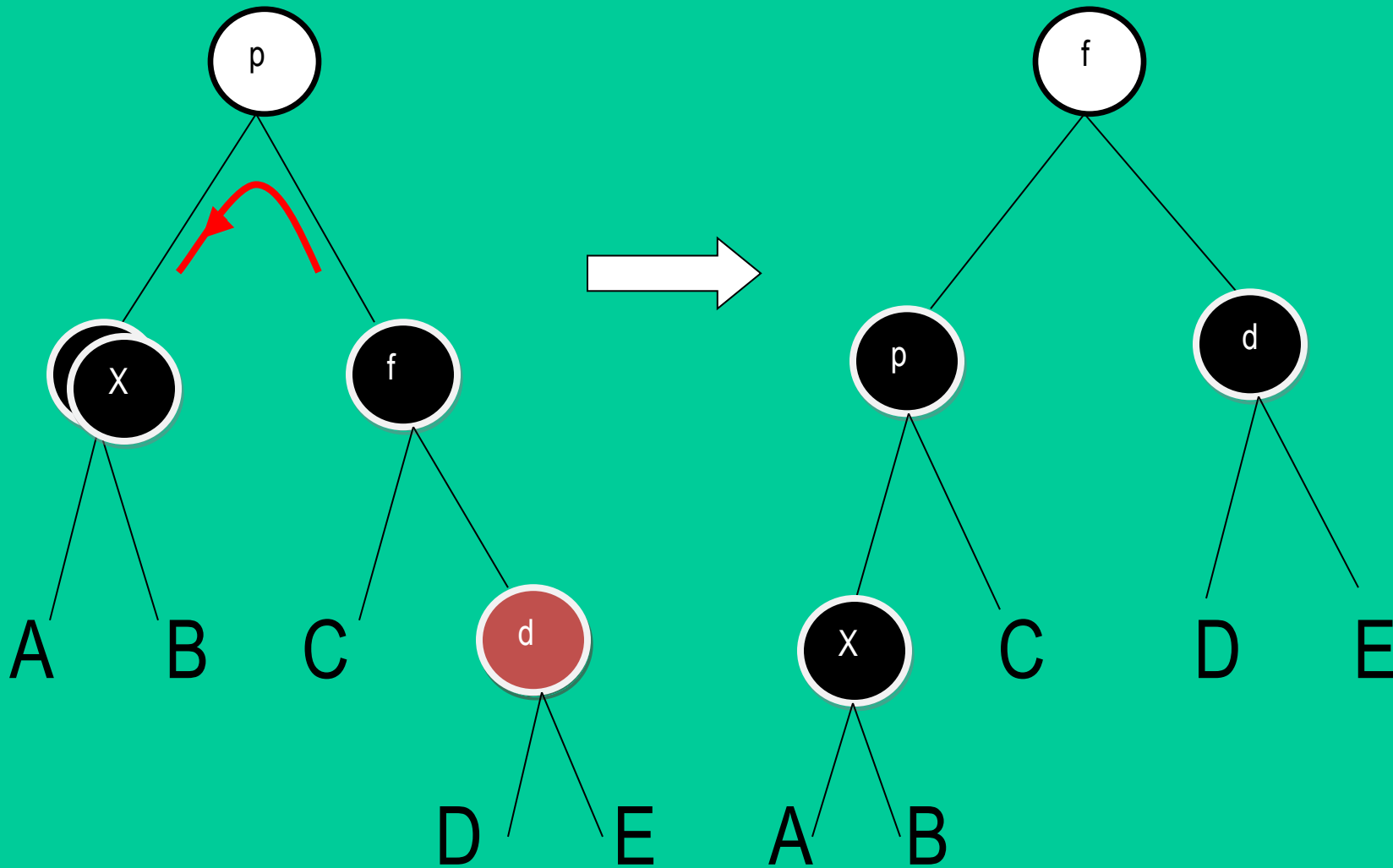


Le nœud **x** devient **noir** et le nœud **f** devient **rouge**.

Le nœud **p** devient **noir** +

Si **p** est **noir**, il devient **noir**² mais il s'est déplacé vers la racine de l'arbre. (vers le cas 1)

Cas 1b : le fils droit **d** de **f** est **rouge**.



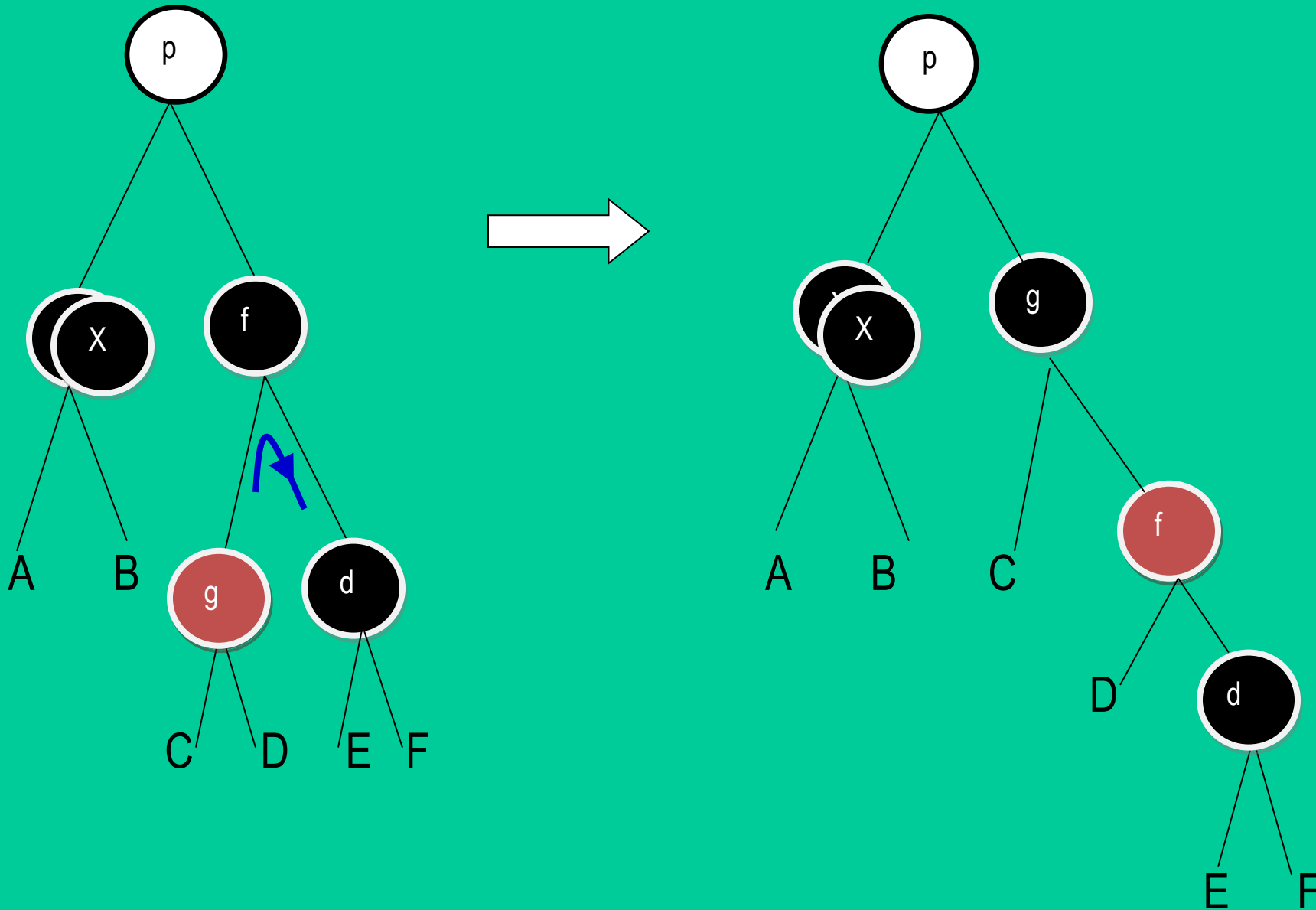
L'algorithme effectue une rotation **rg**(**f**)

Le nœud **f** prend la couleur du nœud **p**.

Les nœuds **x**, **p** et **d** deviennent noirs.

L'algorithme se termine.

Cas 1c : g est **rouge** et d est noir



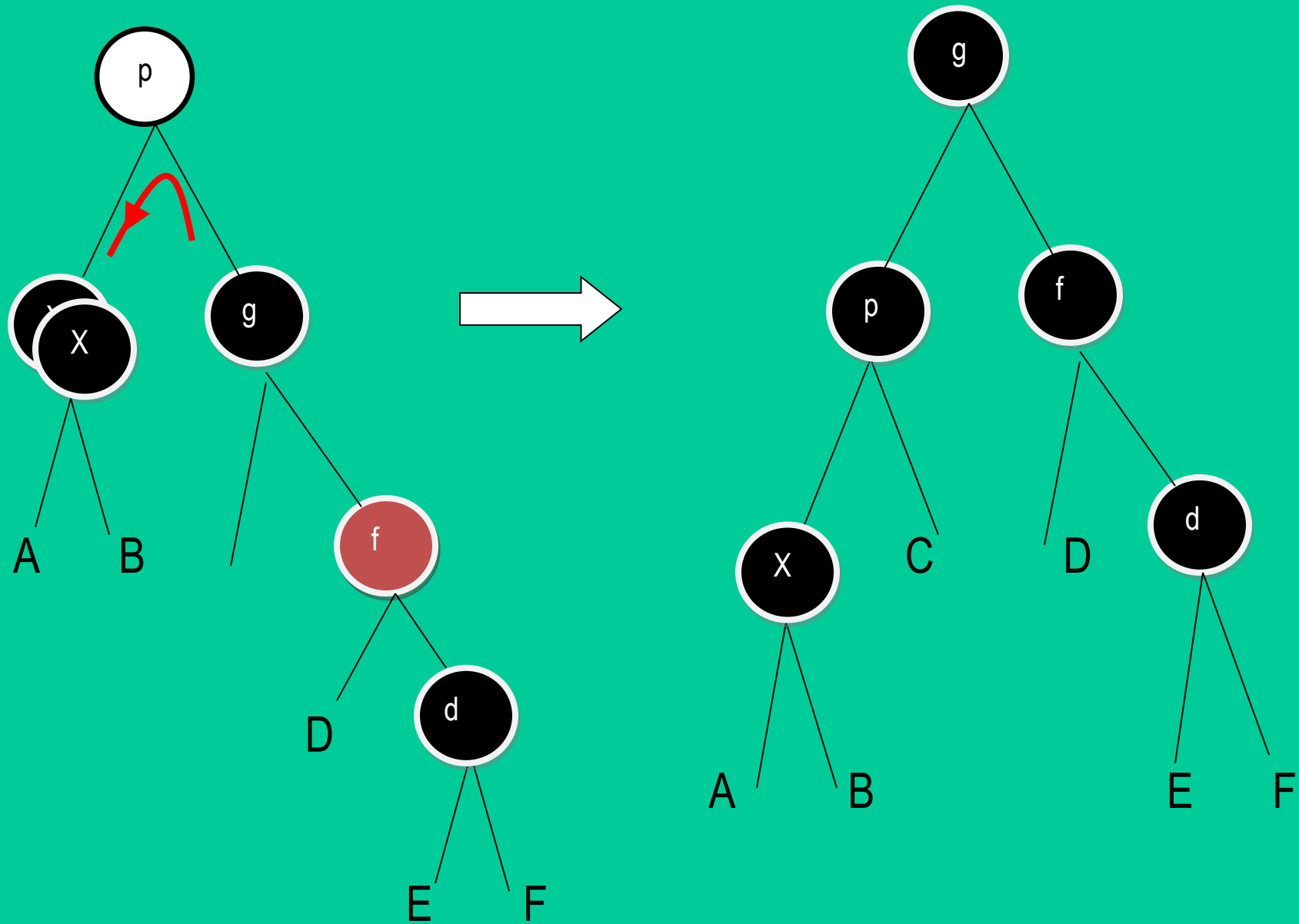
L'algorithme effectue une **rd(g)**

Le nœud **g** devient **noir** et le nœud **f** devient **rouge**.

Il n'y a pas deux nœuds rouges consécutifs puisque la racine du sous- arbre D est **noire**

On est ramené au cas **1b** puisque maintenant :

- le frère de **x** est **g** et il est **noir**
- et le fils **droit** de **g** est **rouge**.



L'algorithme effectue alors une rotation **rg**(g).

Le nœud **f** redevient **noir**.

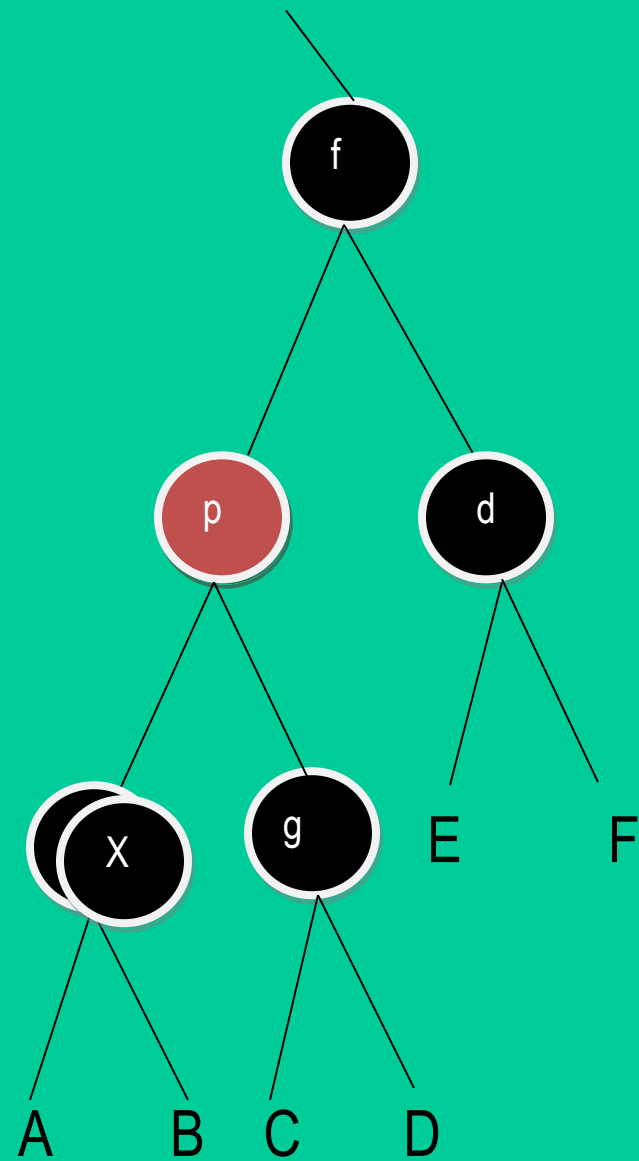
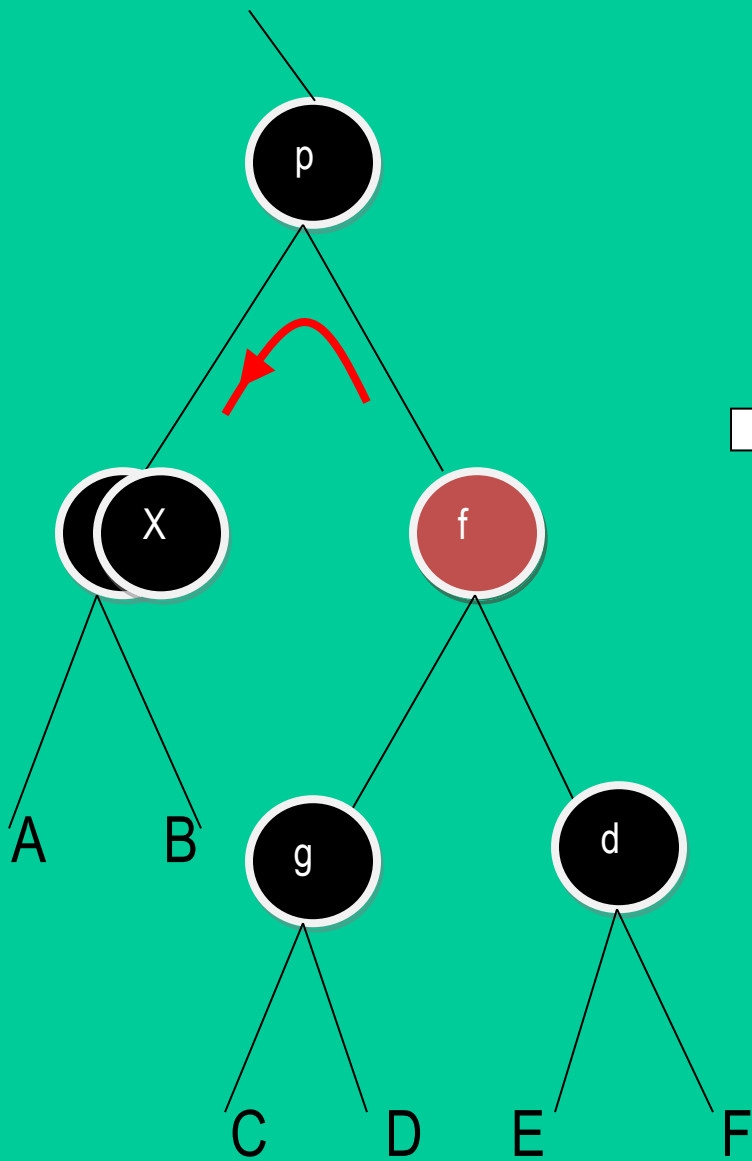
L'algorithme se termine.

Cas 2 : le frère f de x est rouge.

On suppose que x est le fils gauche de p : f est donc le fils droit de p .

Puisque f est **rouge**, le père p de f doit être **noir**.

Les deux fils de f : g et d , doivent être aussi **noirs**.

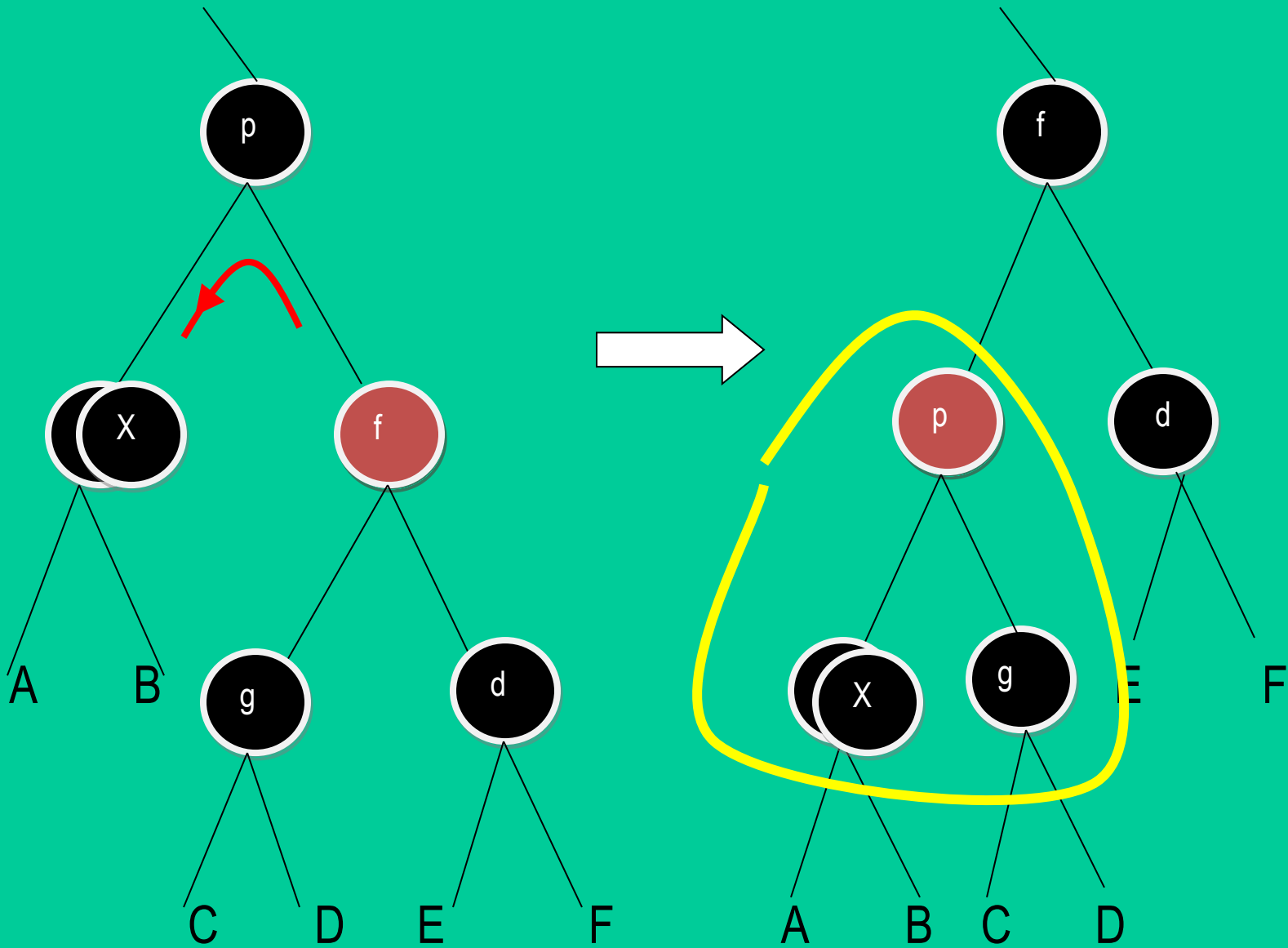


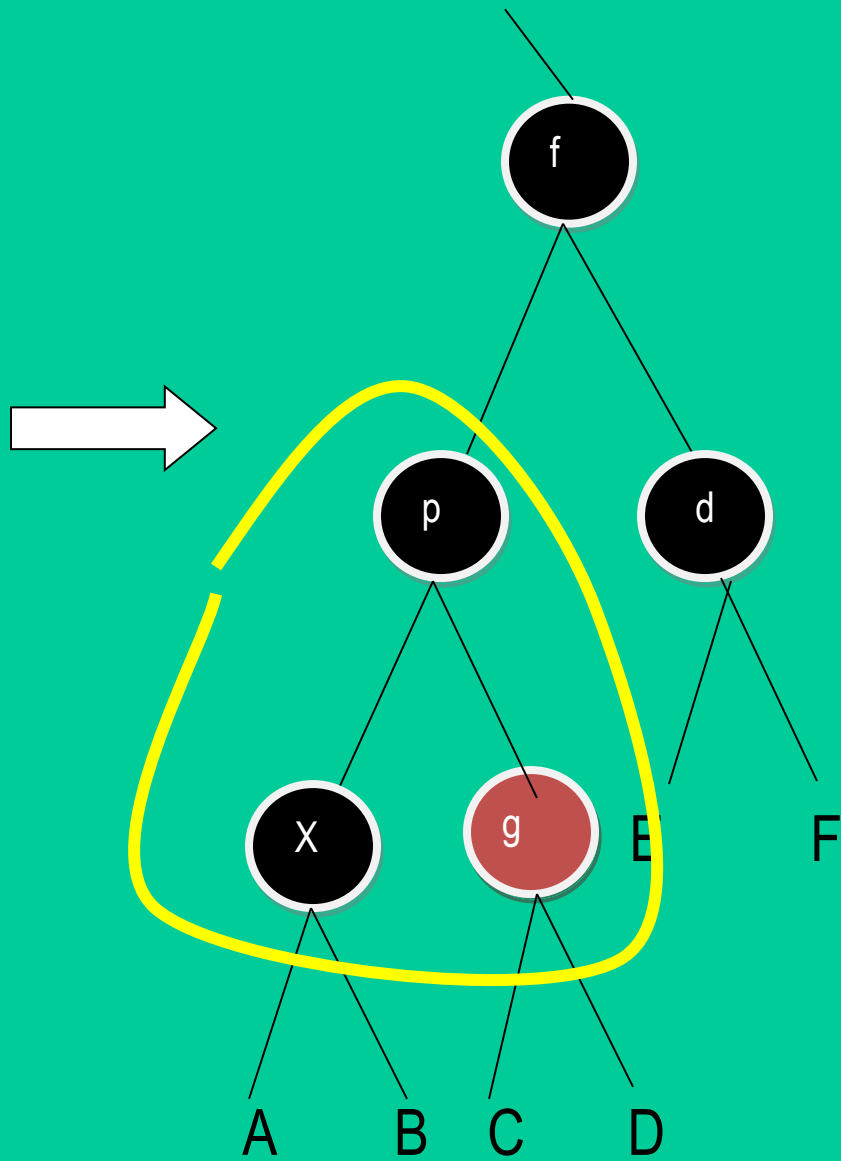
L'algorithme effectue alors une rotation **rg**(f).

Ensuite **p** devient **rouge** et **f** devient **noir**.

Le nœud **x** reste noir² mais son frère est maintenant le nœud **g** qui est noir.

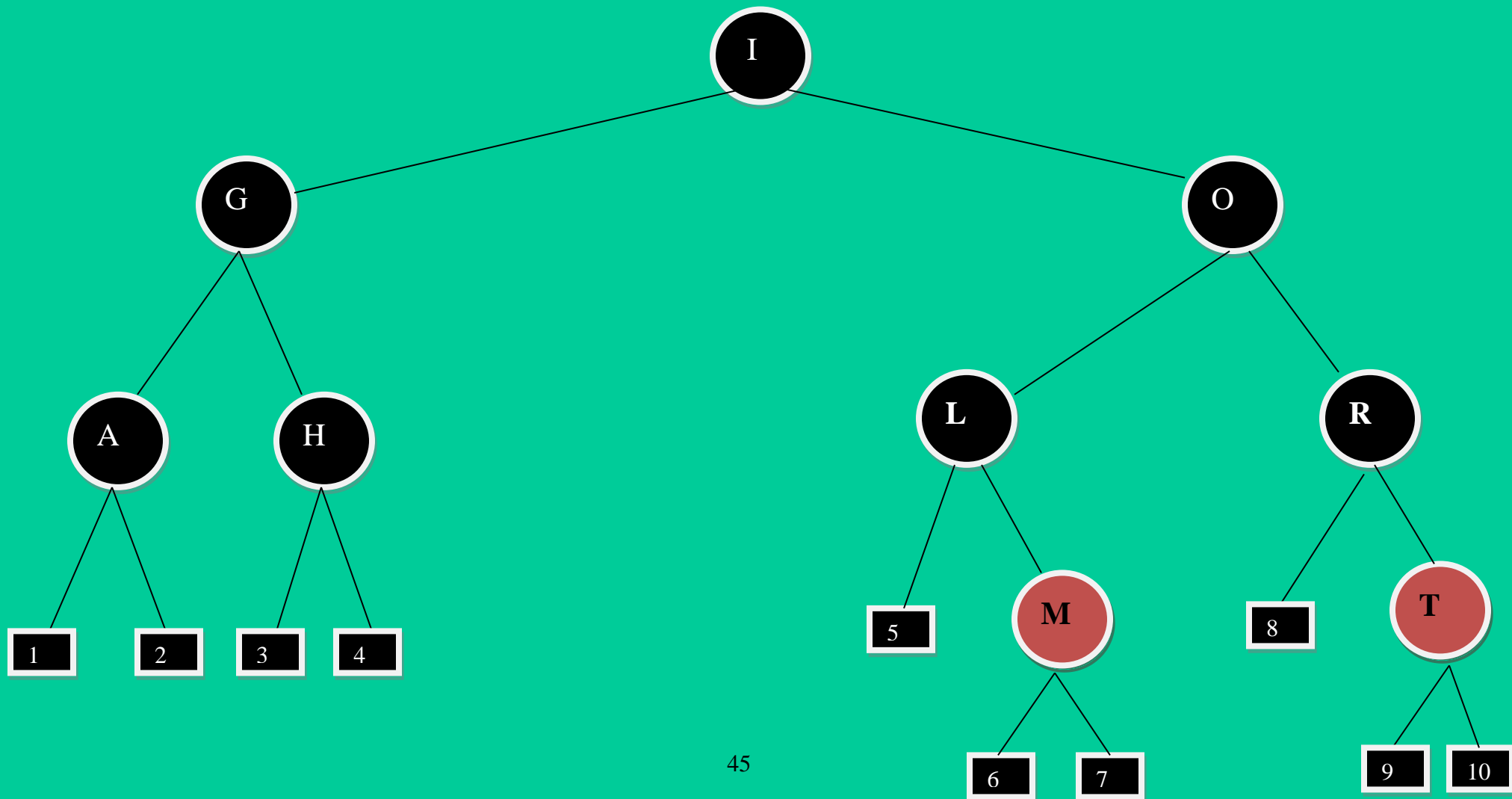
On est donc ramené au cas1.



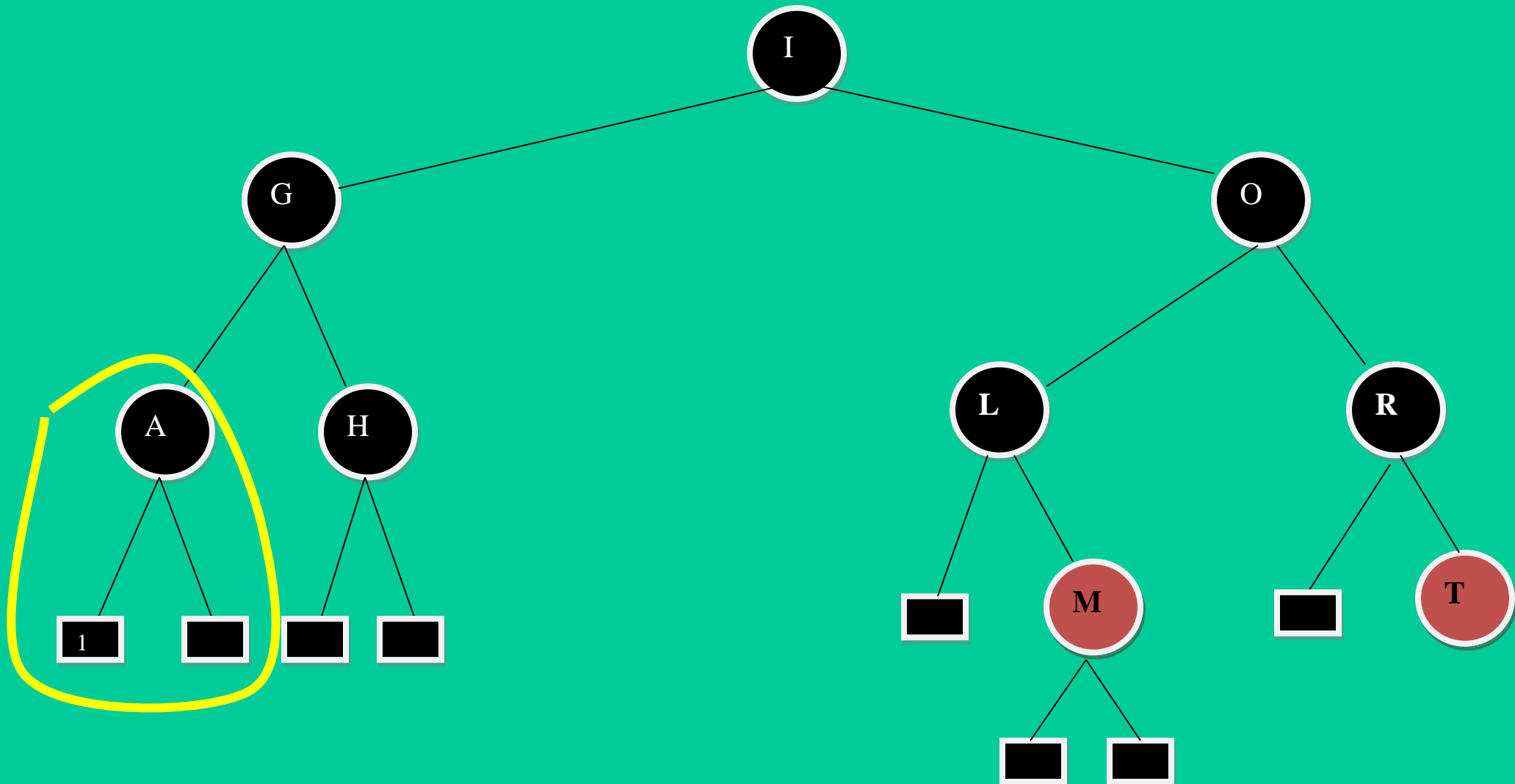


Exemple d'application type

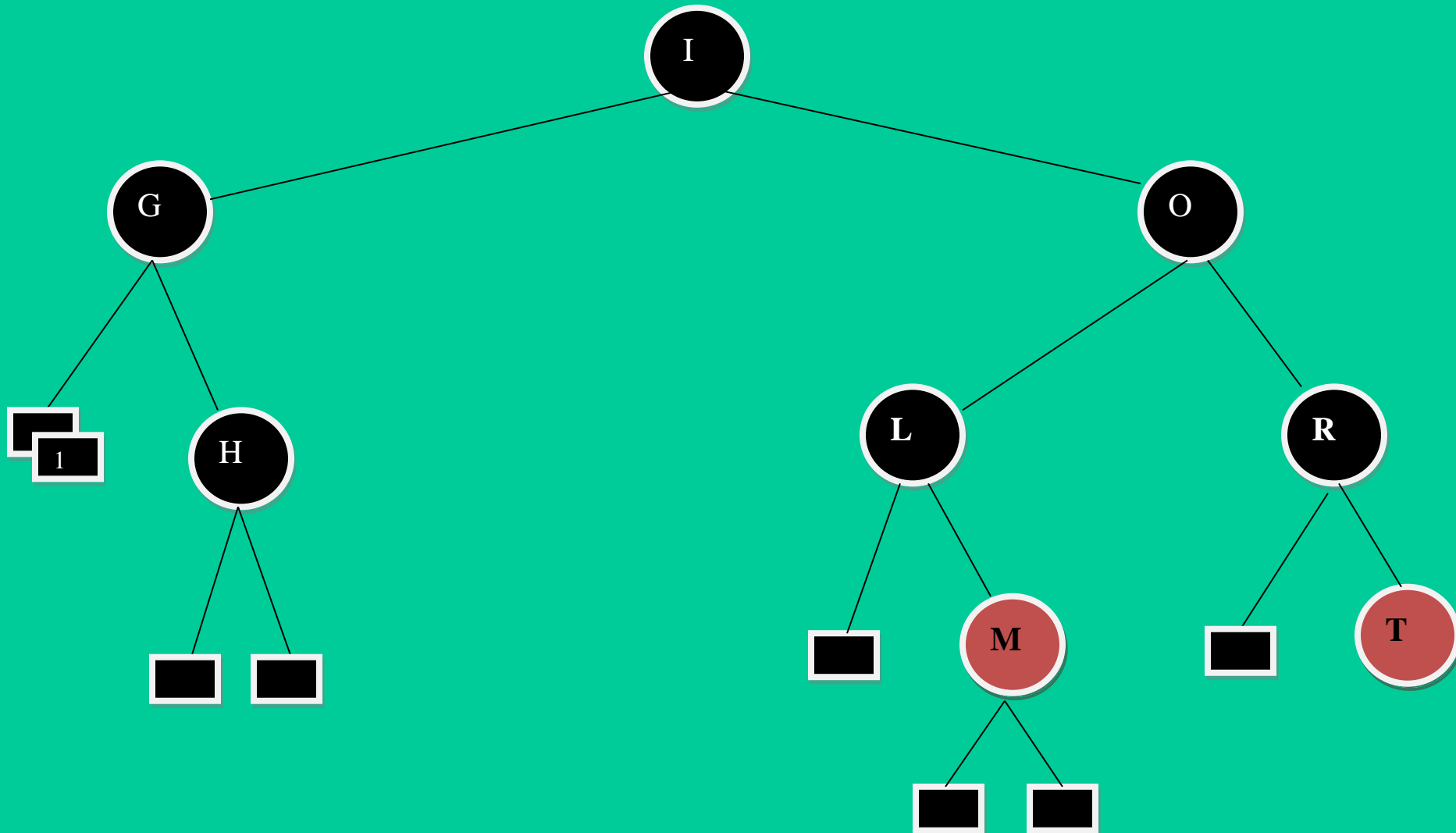
Supprimer les nœuds A, L, G, O, R, I, T, H, M



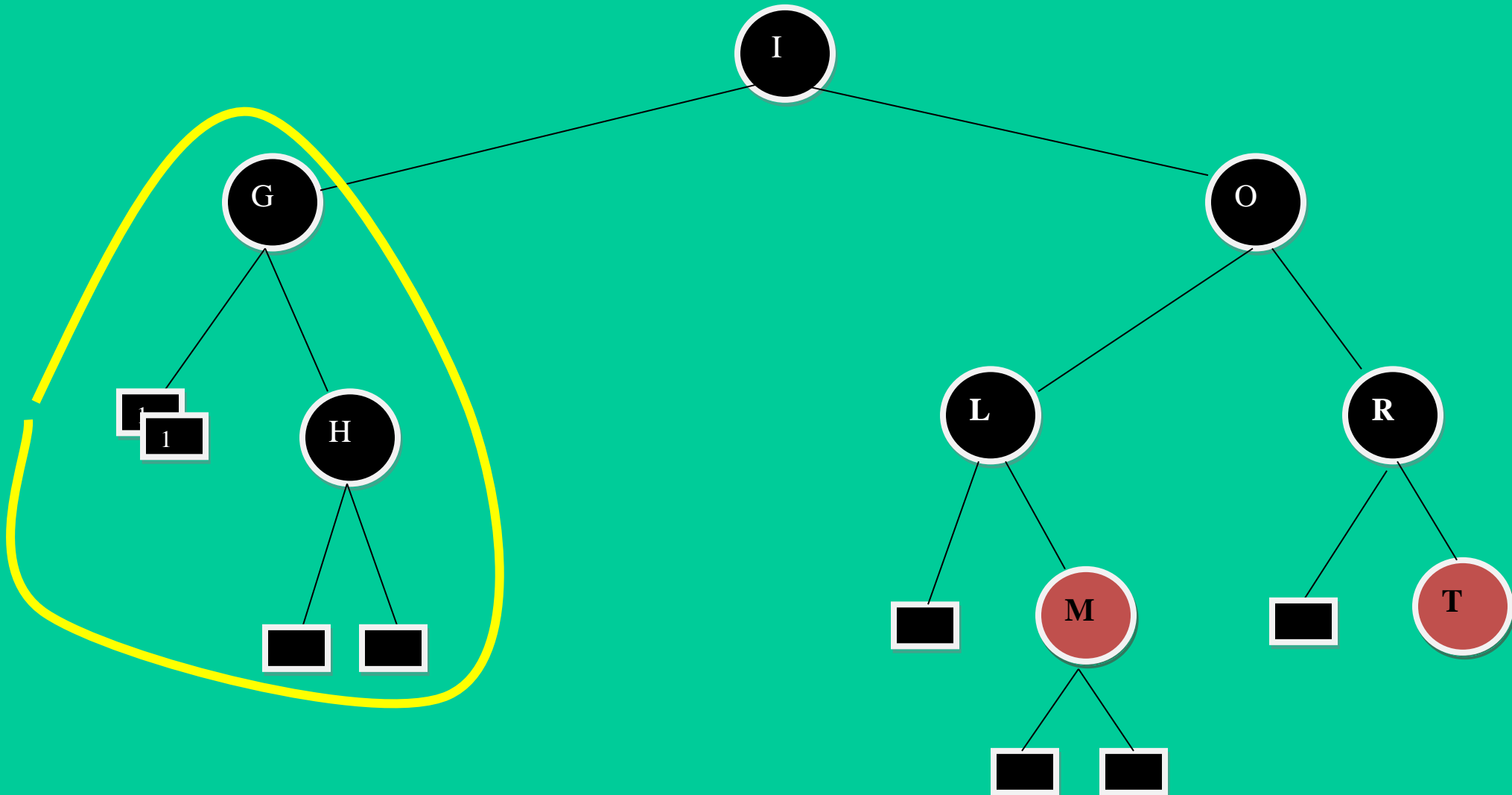
Acte 0: supprimer A : A sera supprimé



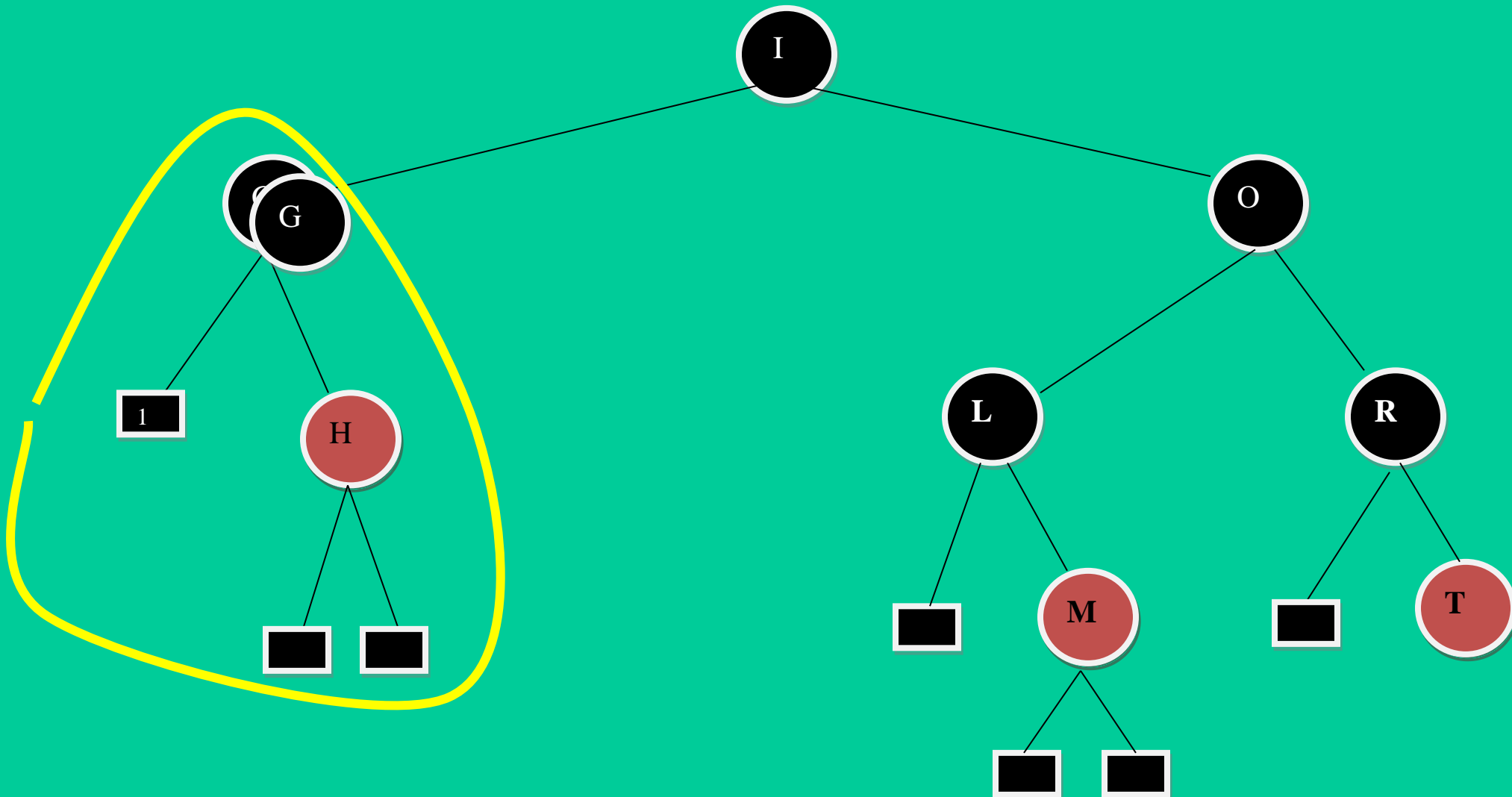
Acte 1: A est supprimé
...et la sentinelle 1 qui remplace A devient **noir**²



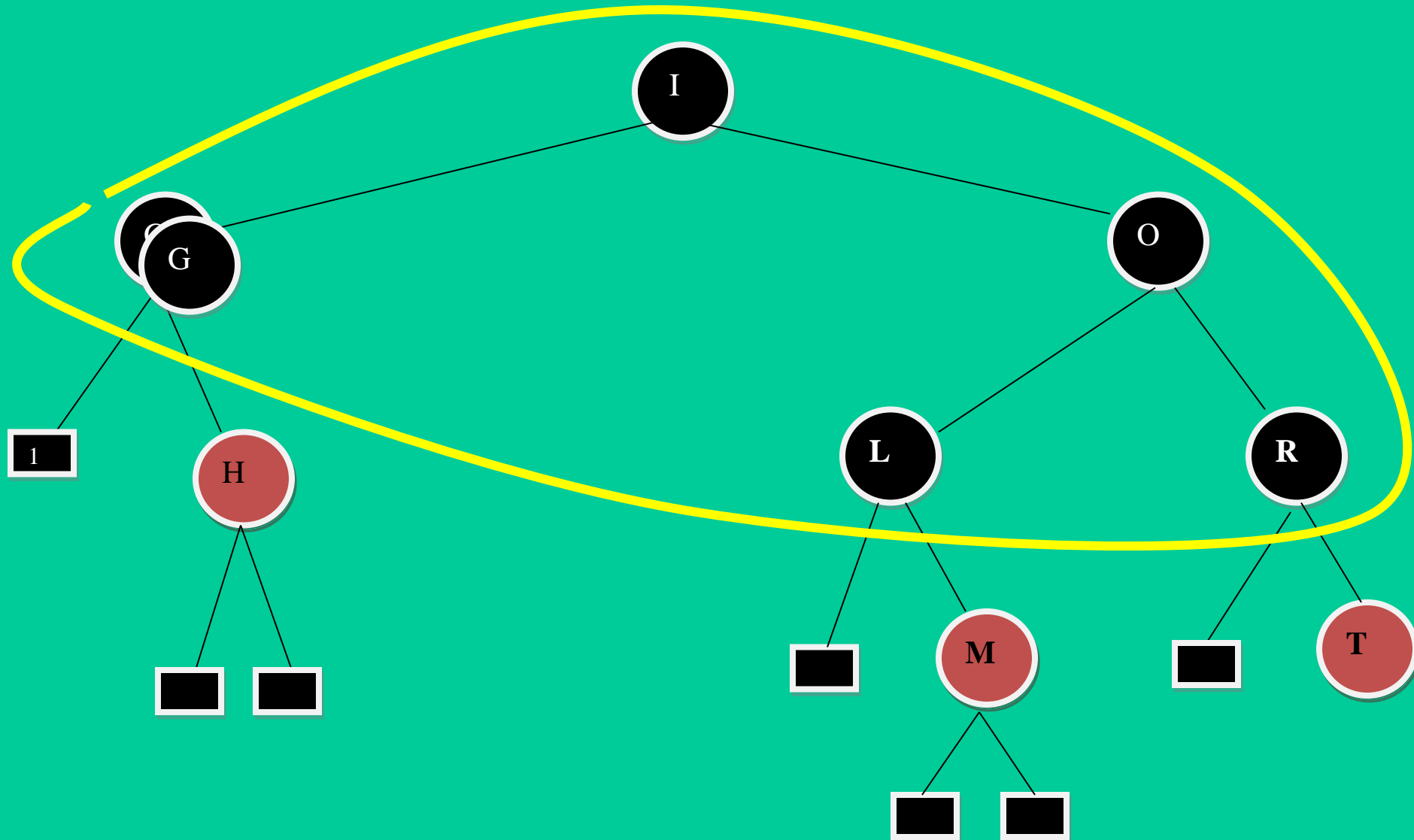
Acte 2 : appliquer le cas 1.a



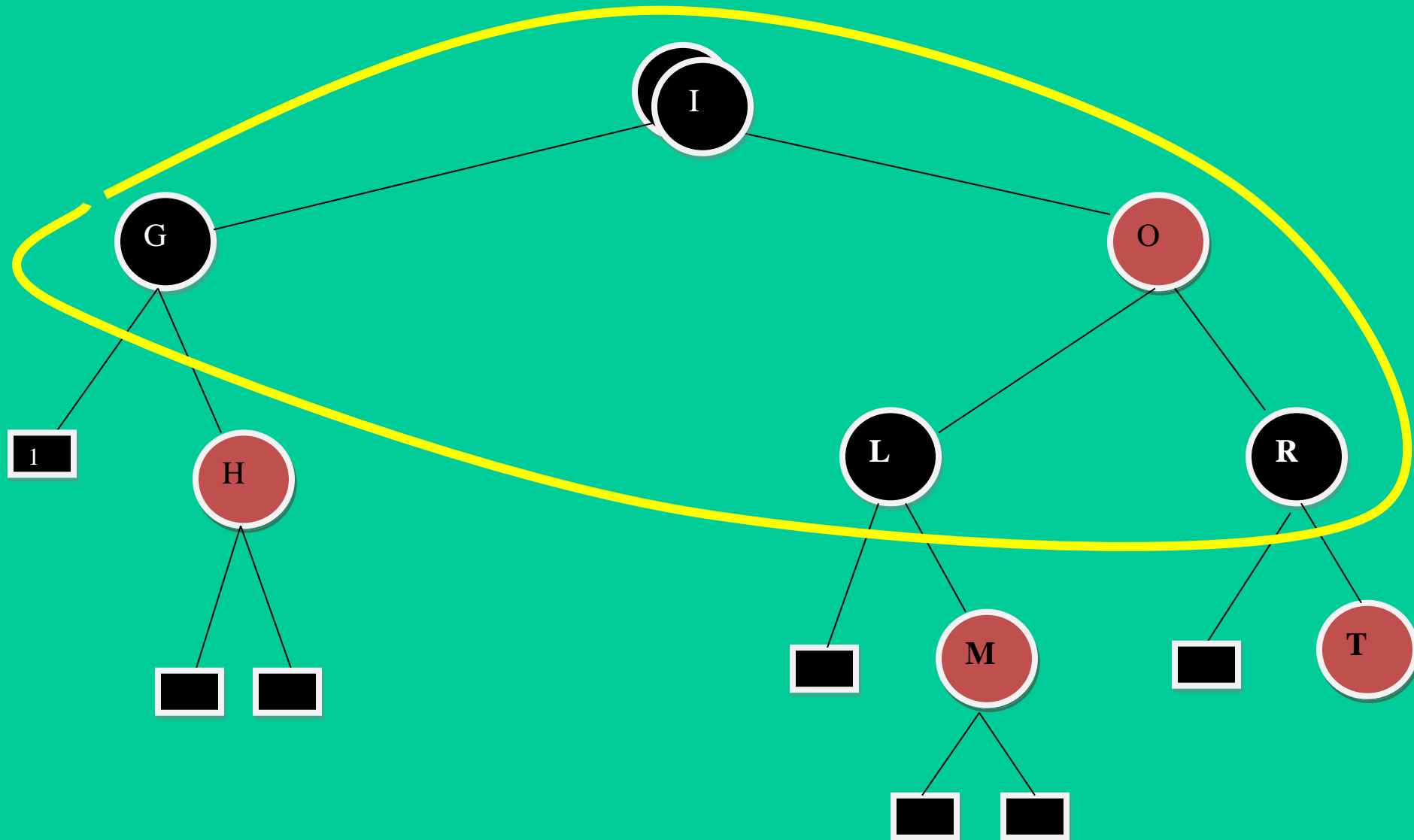
Résultat: **G** en noir², **H** en rouge et sentinelle 1 en noir



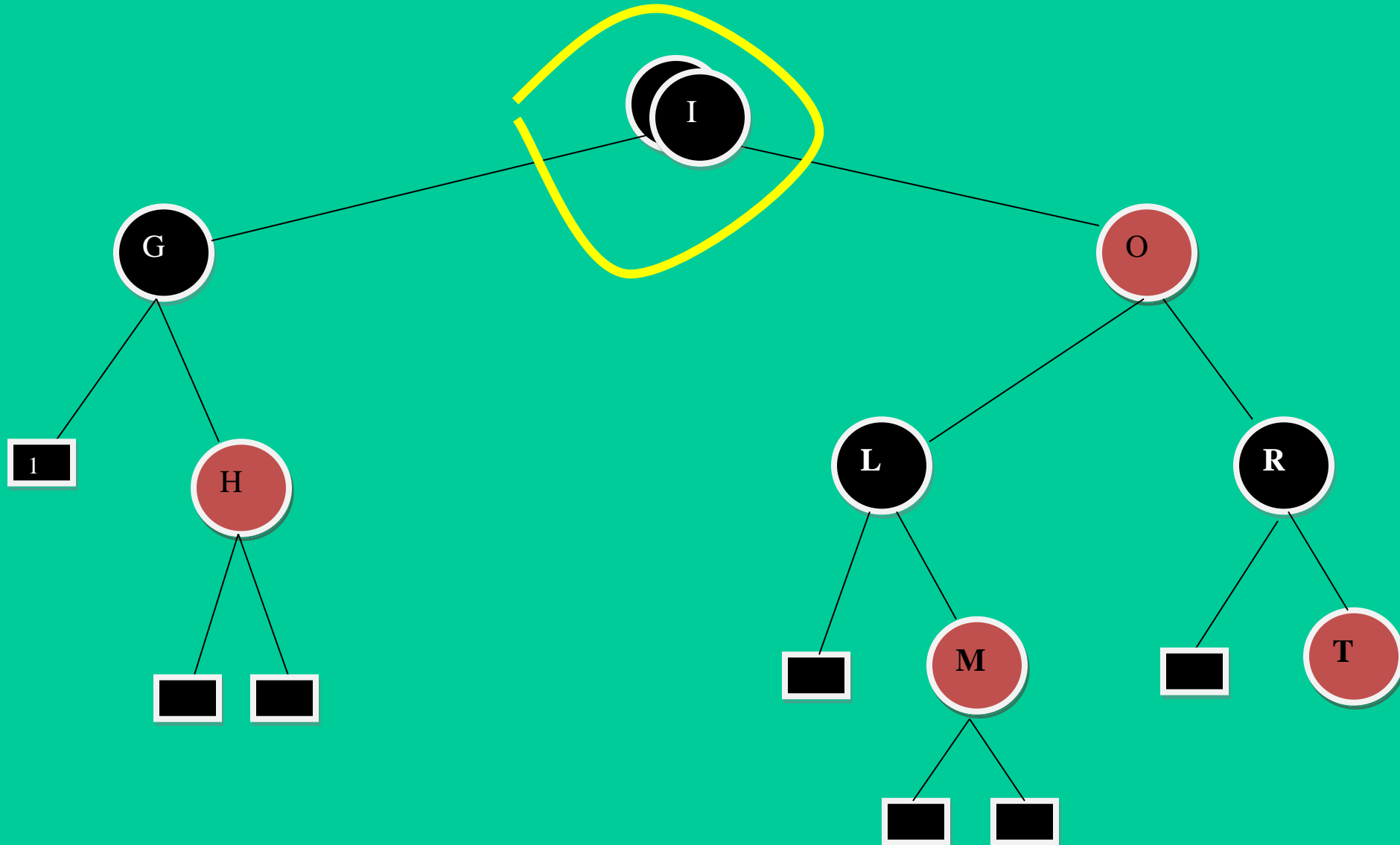
Acte 3: appliquer le cas 1.a



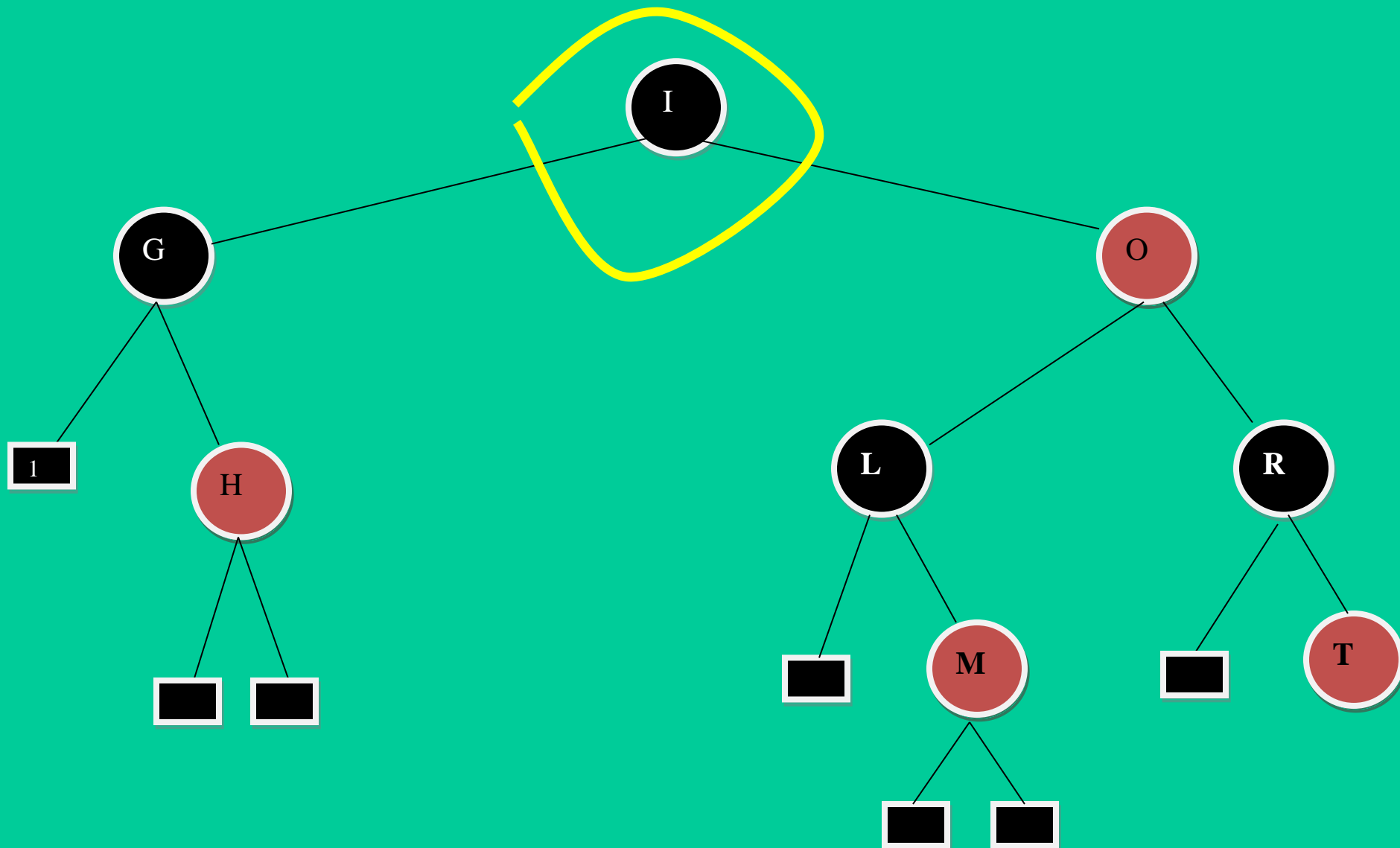
Résultat: G en noir, I noir² et O en rouge



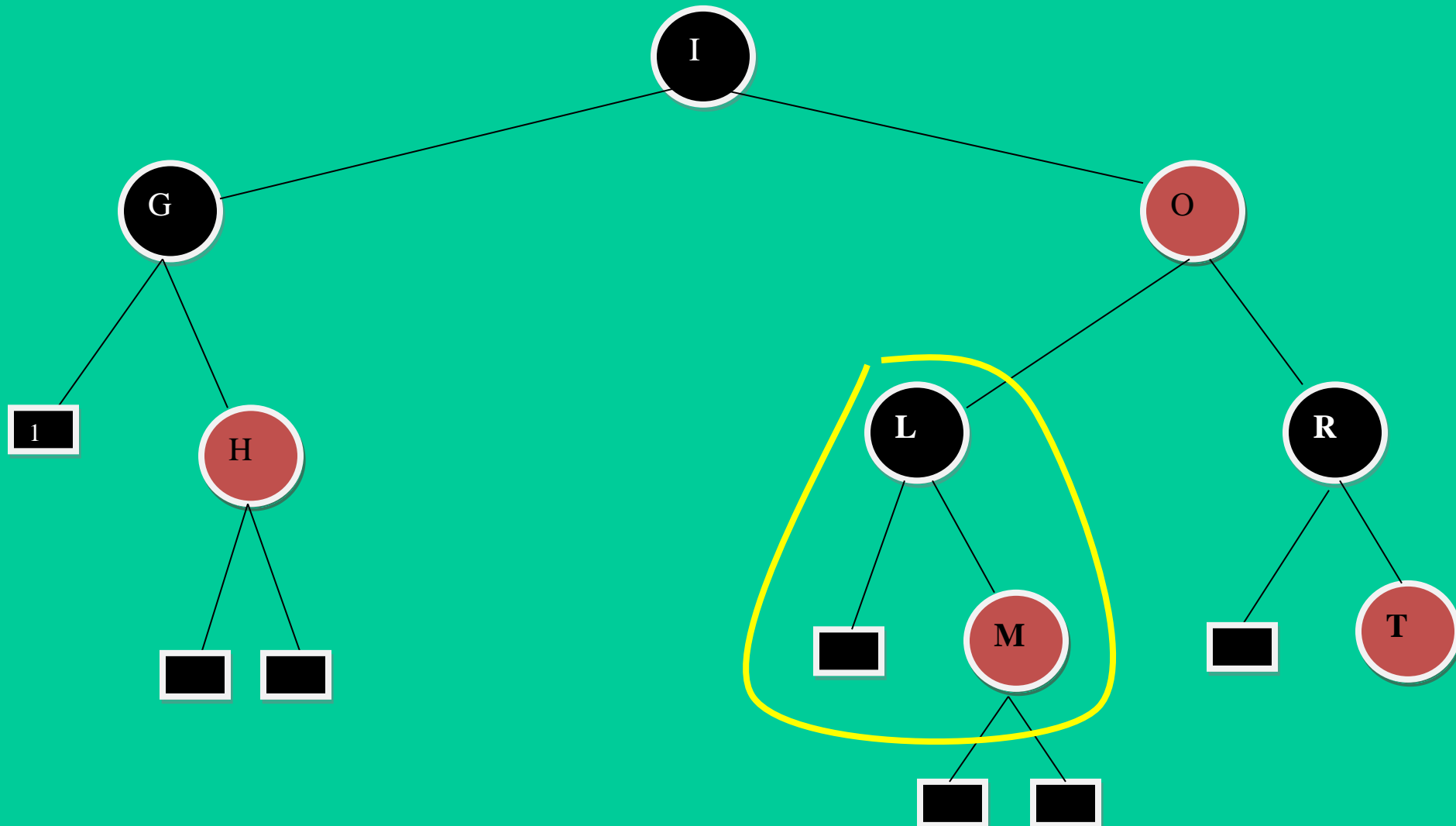
Acte 4: appliquer **cas 0**



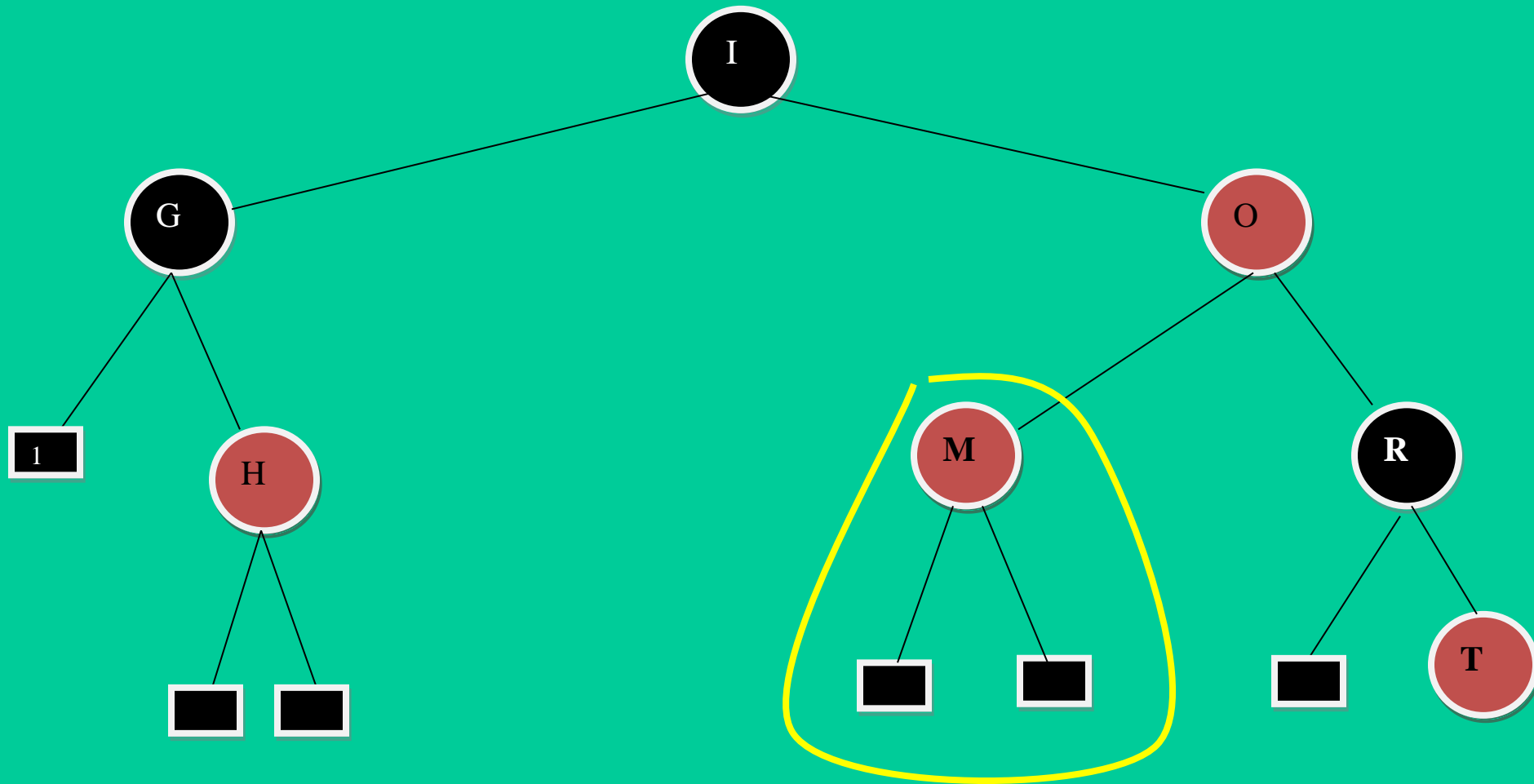
Résultat: I en noir ; $Hn(I) = 1$



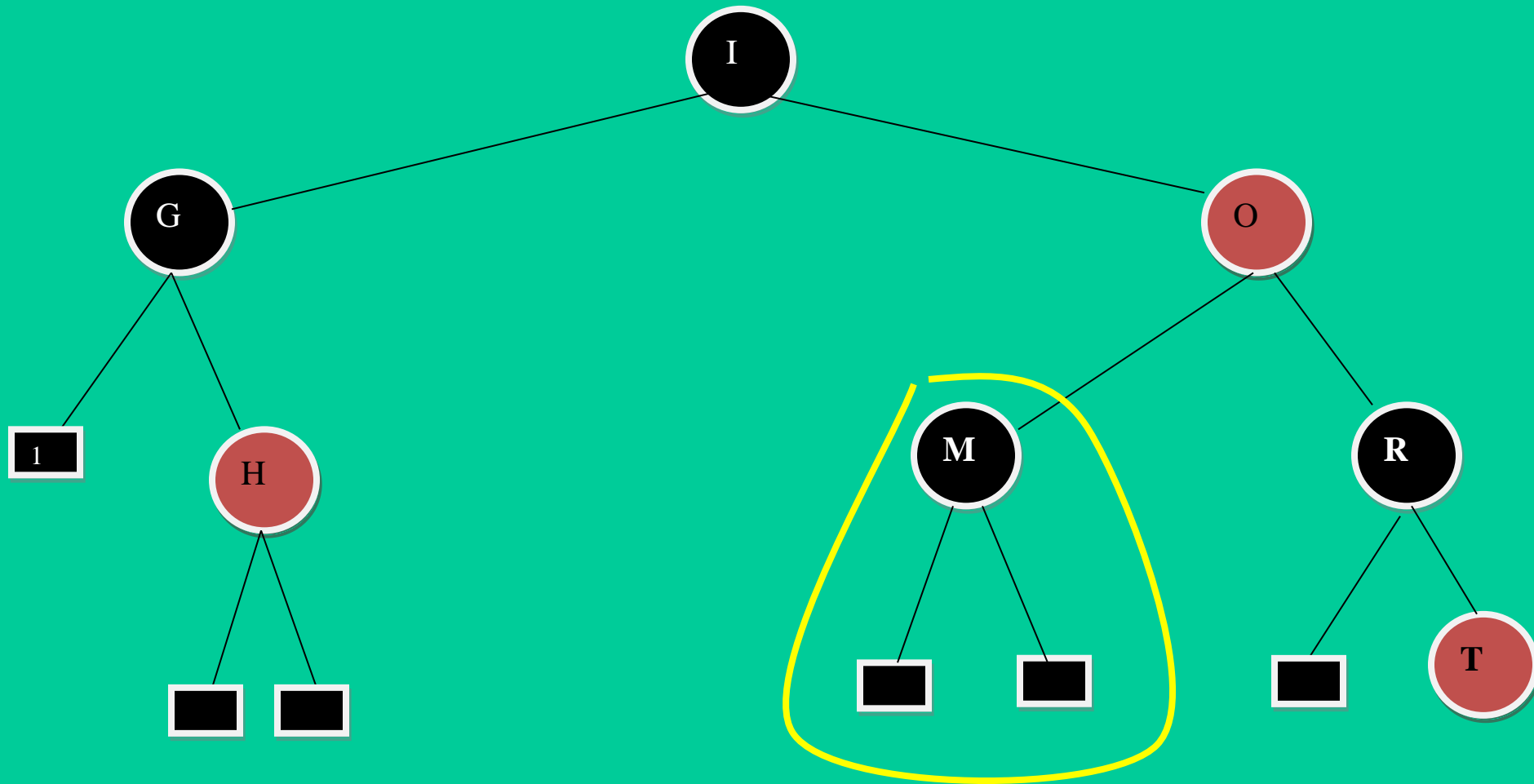
Acte 0: Supprimer L : L sera supprimé



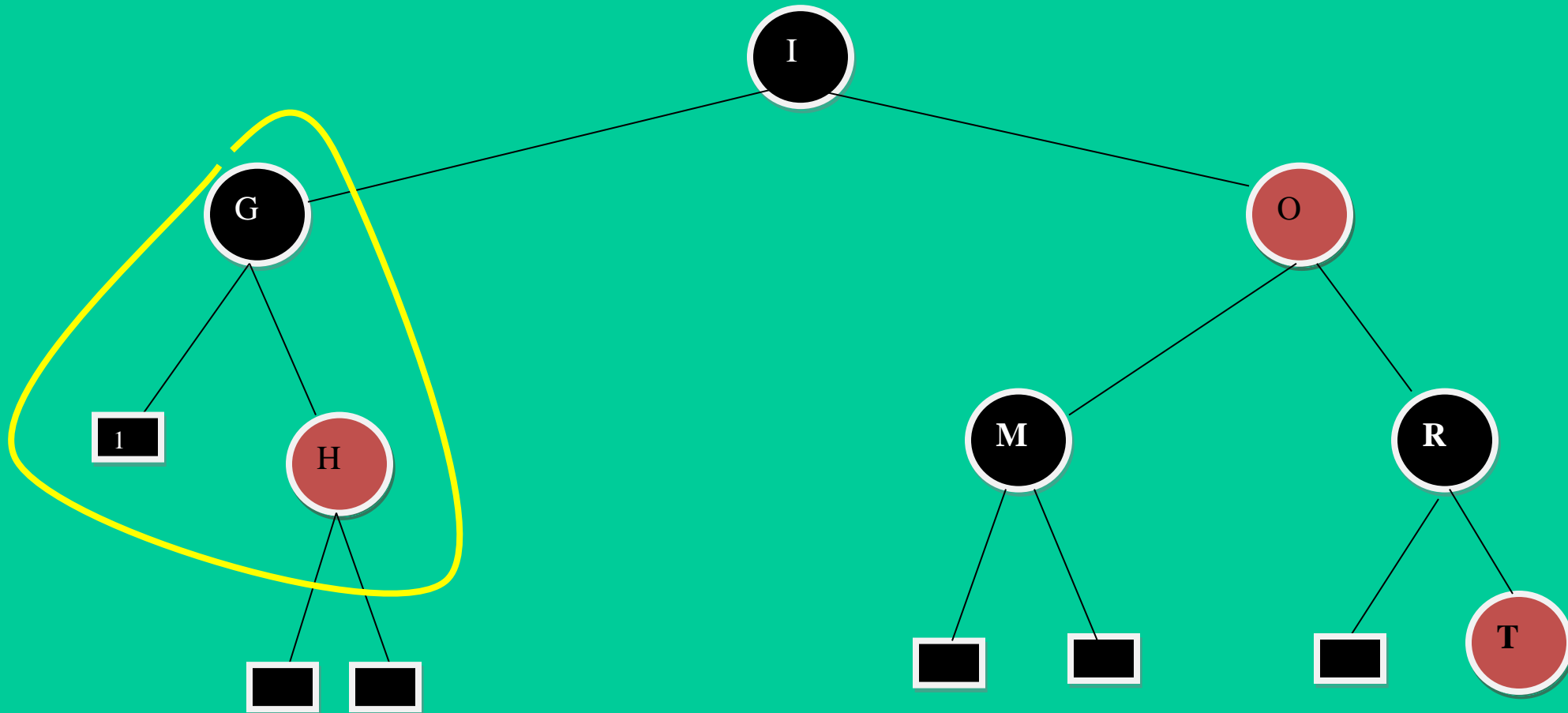
Acte 1: supprimer L ...et le remplacer par **M** en noir⁺



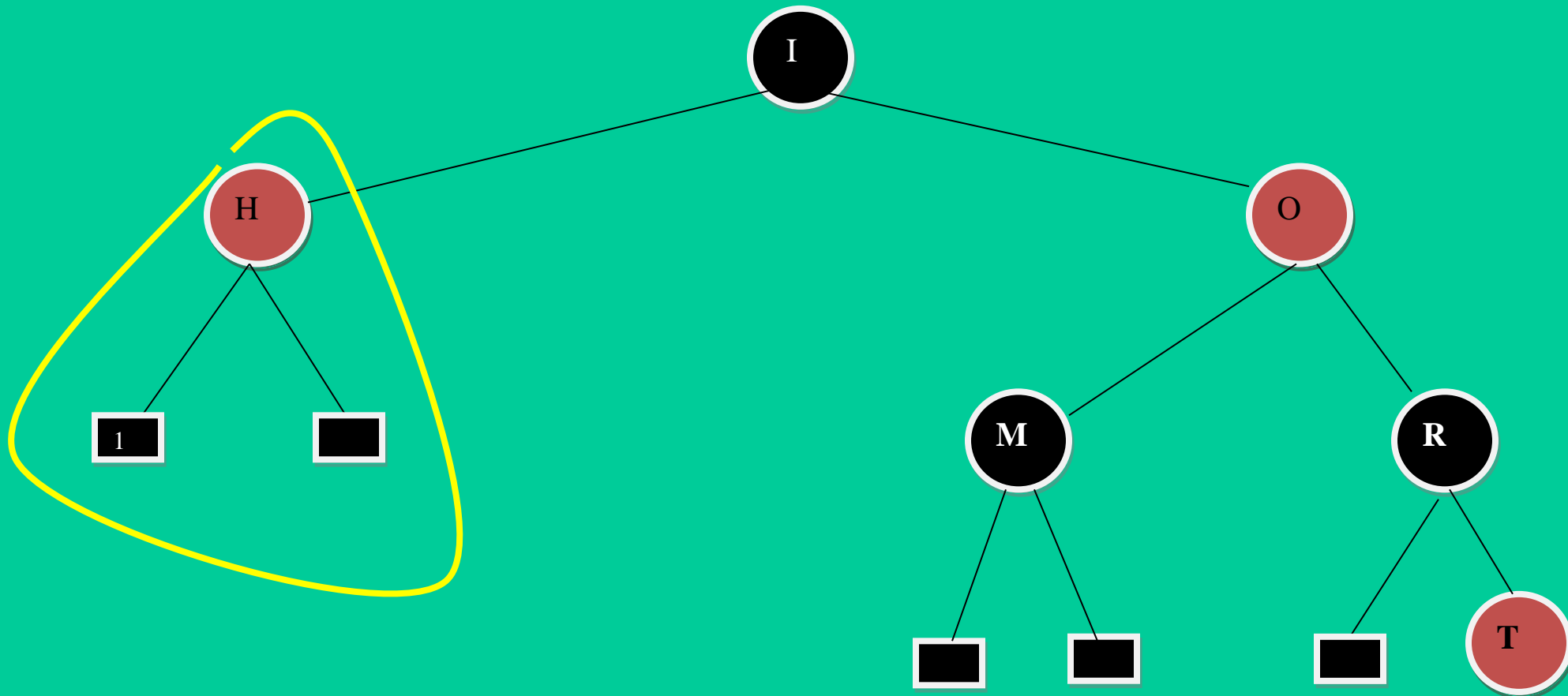
Acte 2 : **M** devenant noir+, passe du **rouge** au **noir**



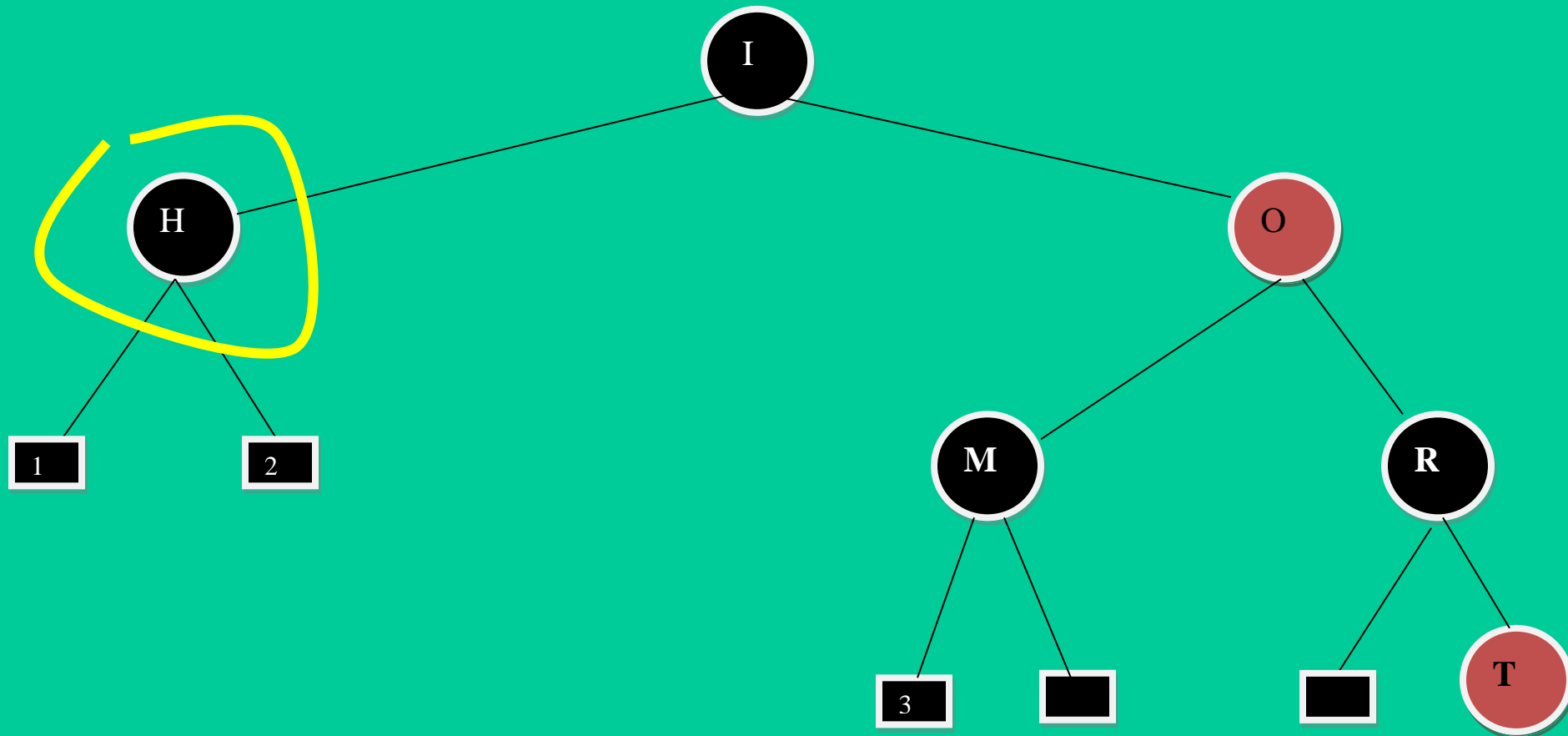
Acte 0 : supprimer G: G sera supprimé



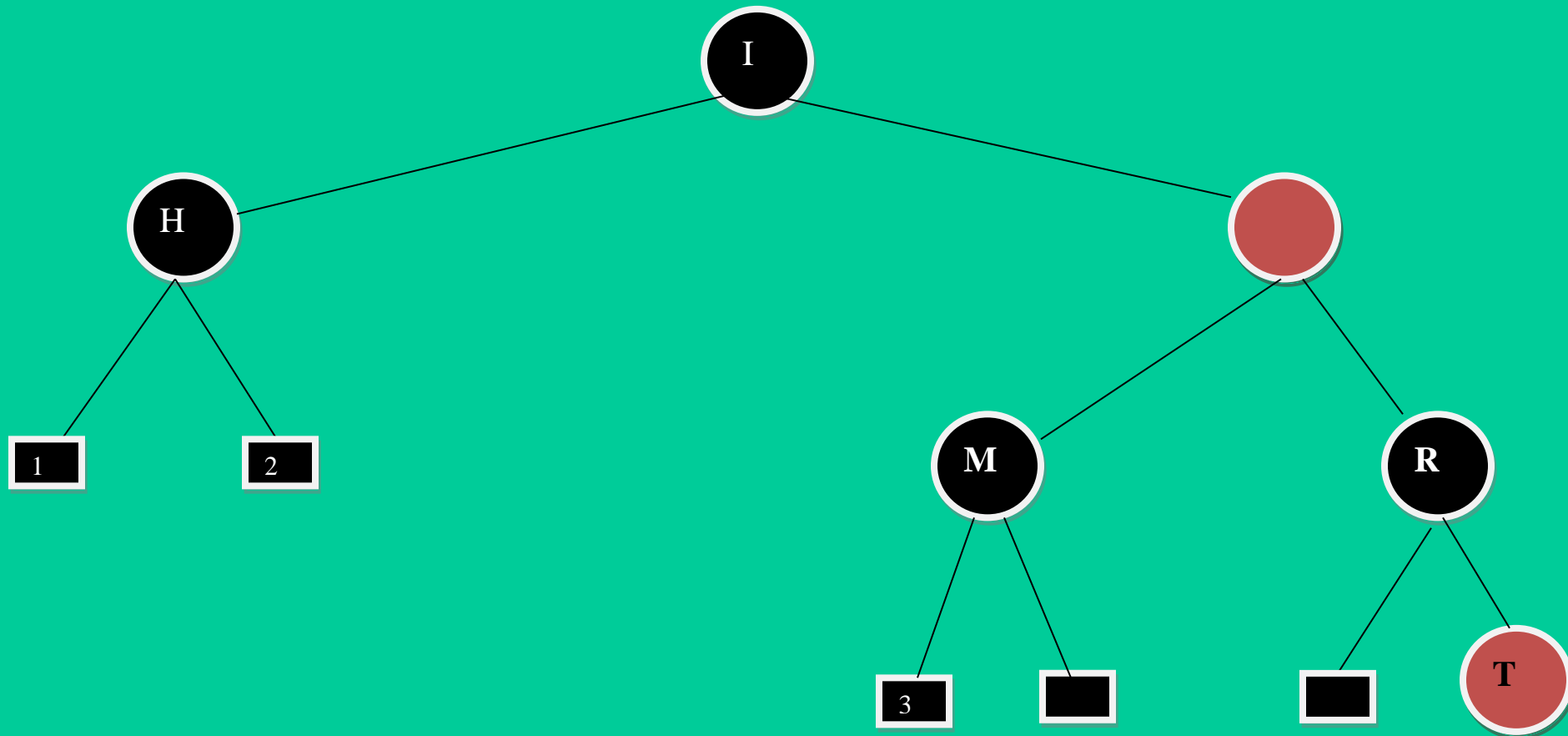
Acte1 : **G** supprimé ...est remplacé par **H** qui deviendra noir+



Acte 2 : H devenant noir+, passe du rouge au noir



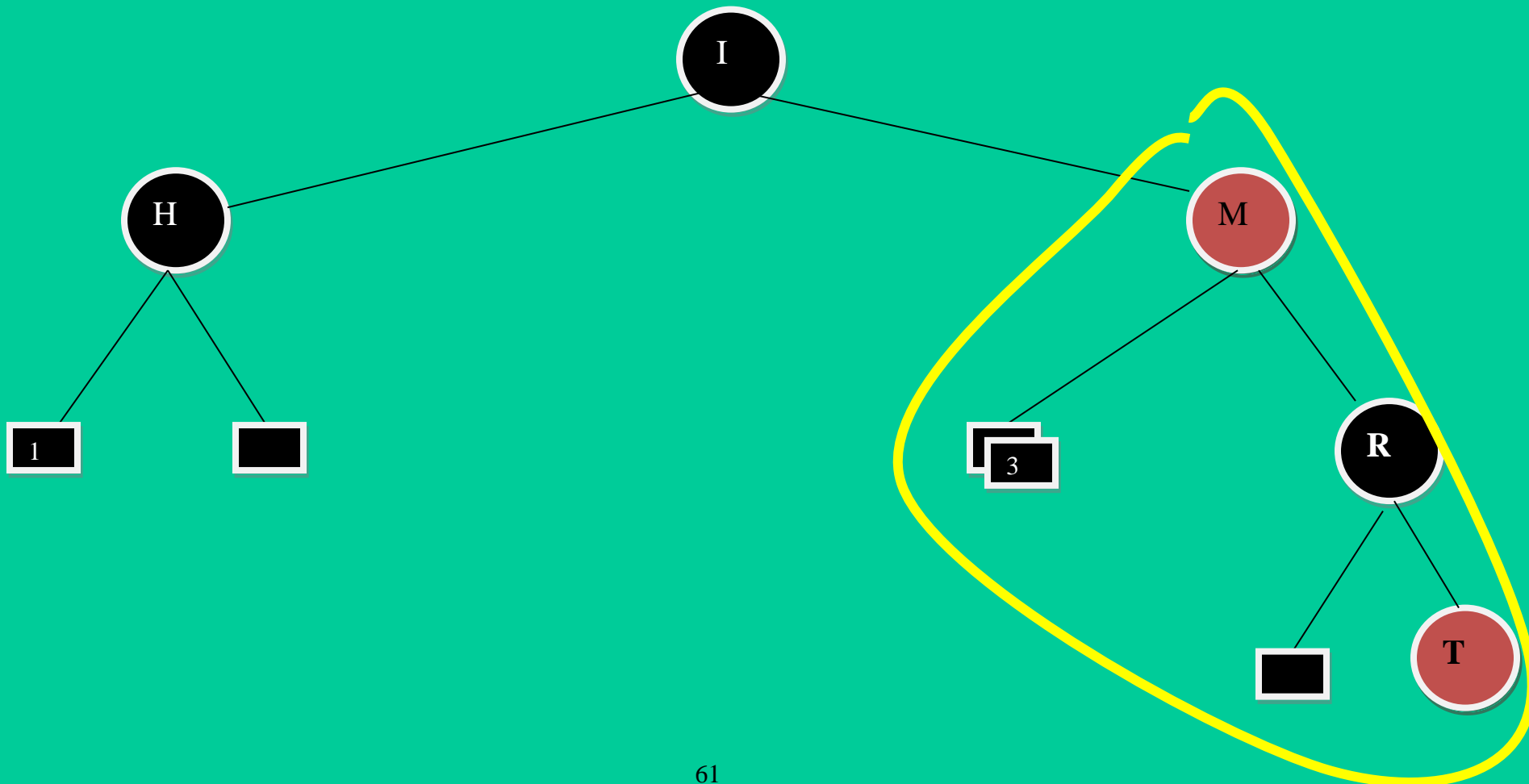
Acte 0 : supprimer 0: 0 n'est pas supprimé



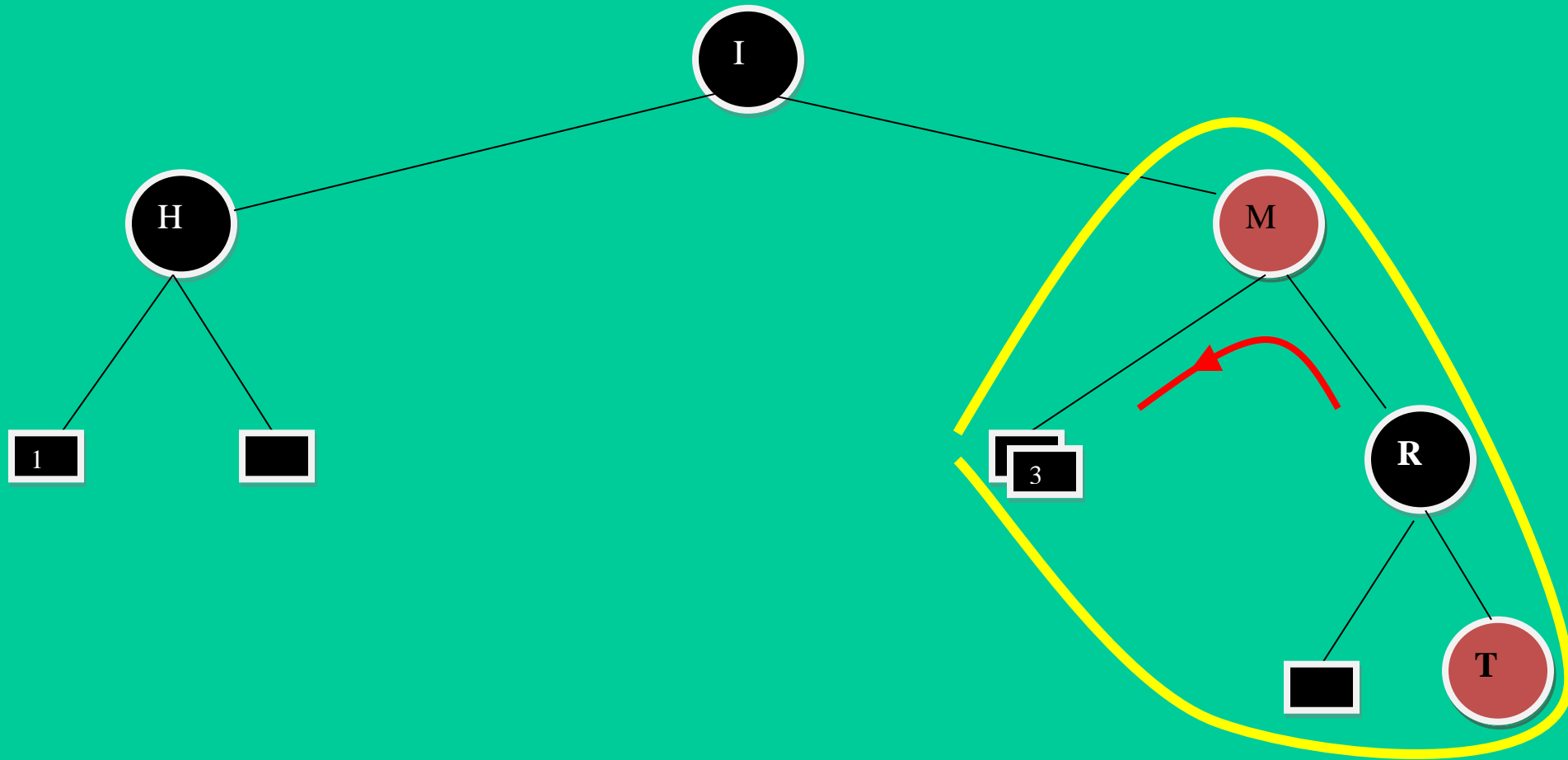
Acte1 : **M** remplace **O** et devient rouge

M supprimé ;

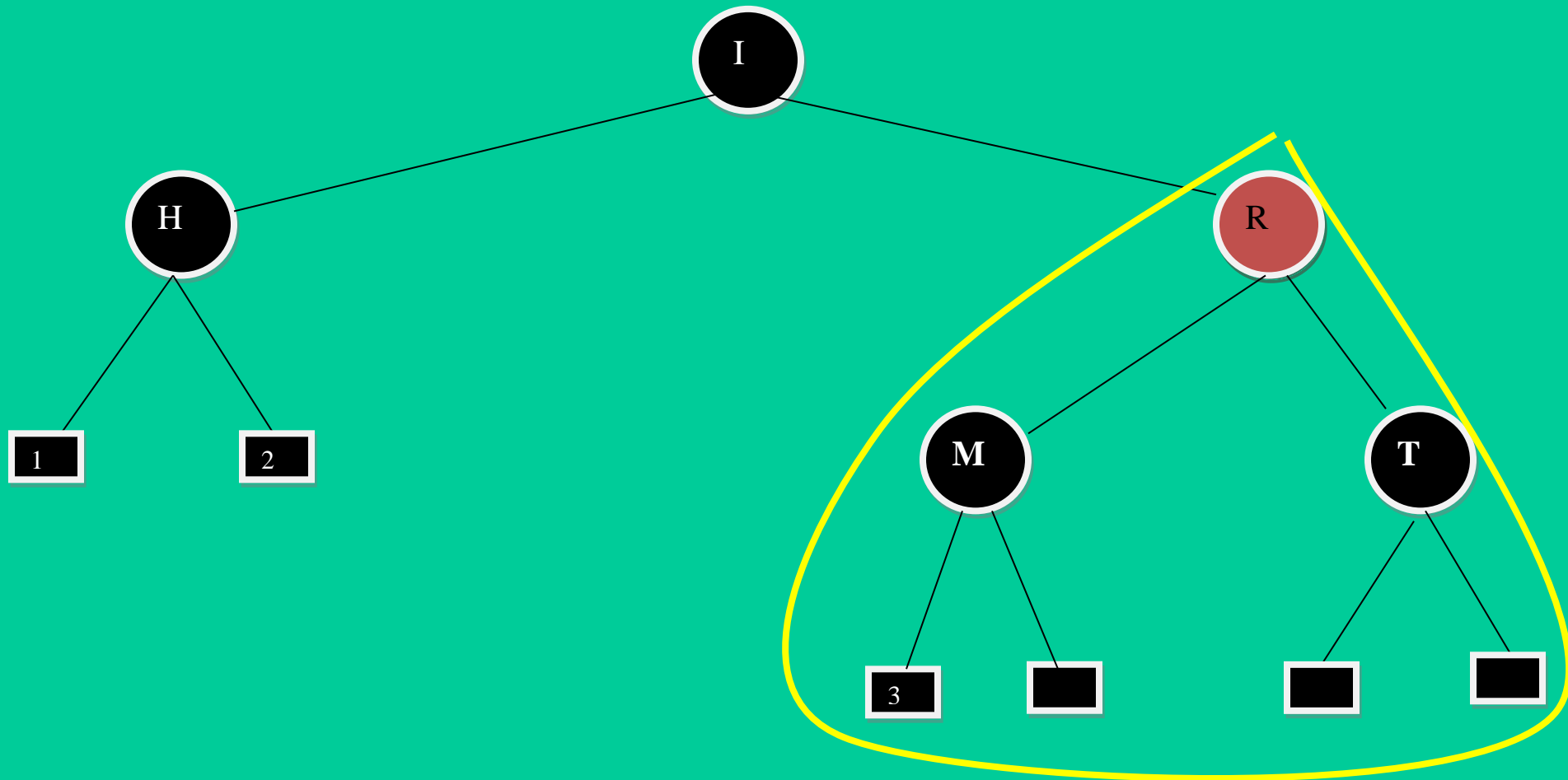
La sentinelle **3** remplace **M** et devient **noir**²



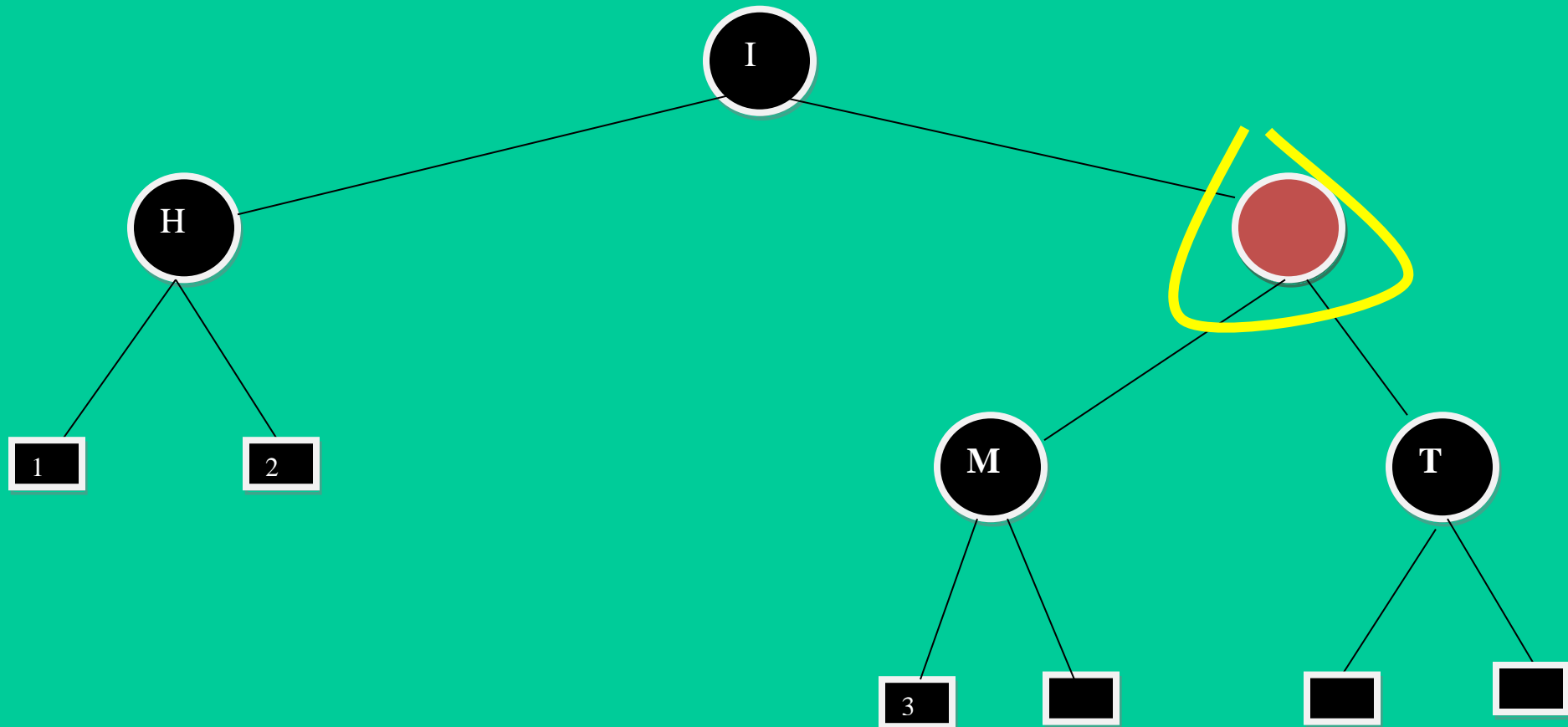
Acte 2 : appliquer cas 1.b



Résultat :



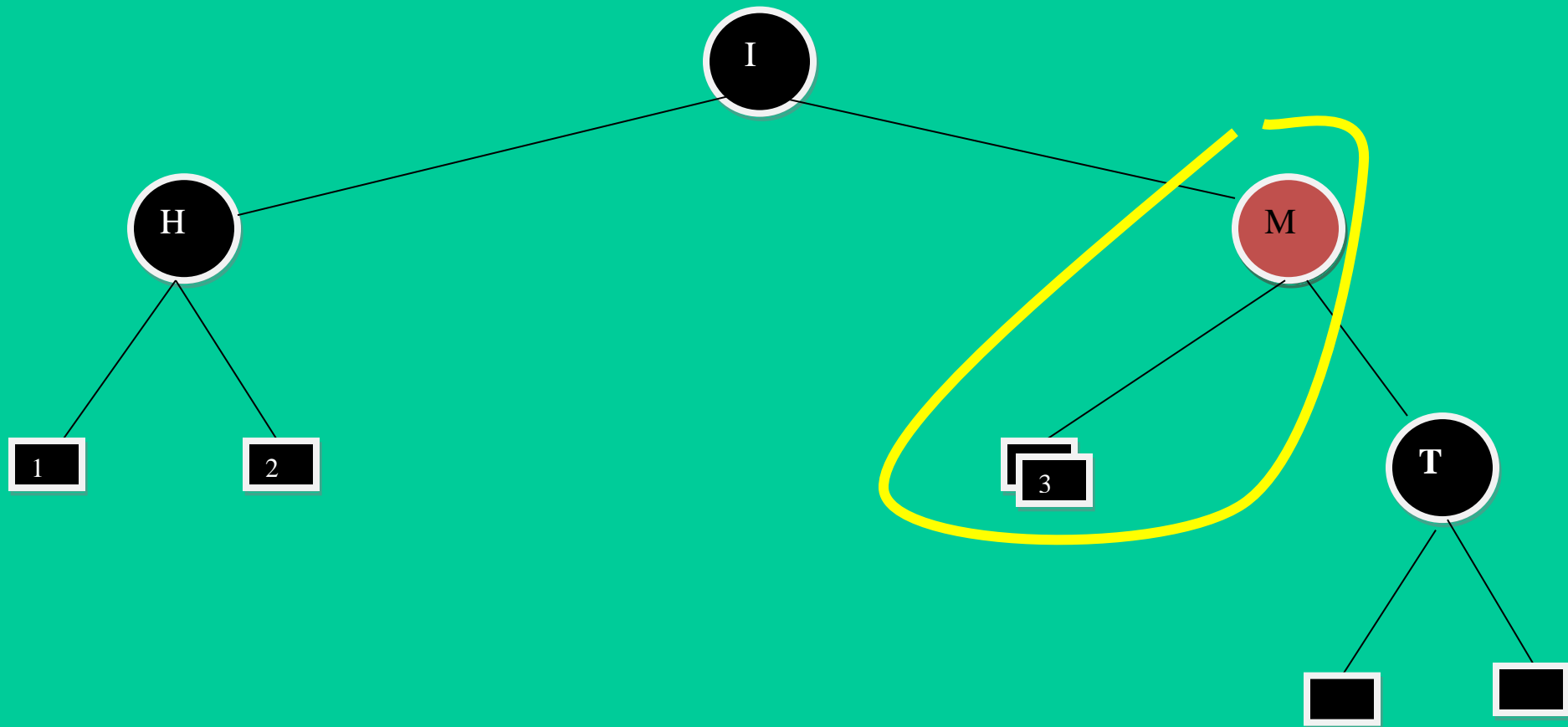
Acte 0 : supprimer **R** : **R** ne sera pas supprimé



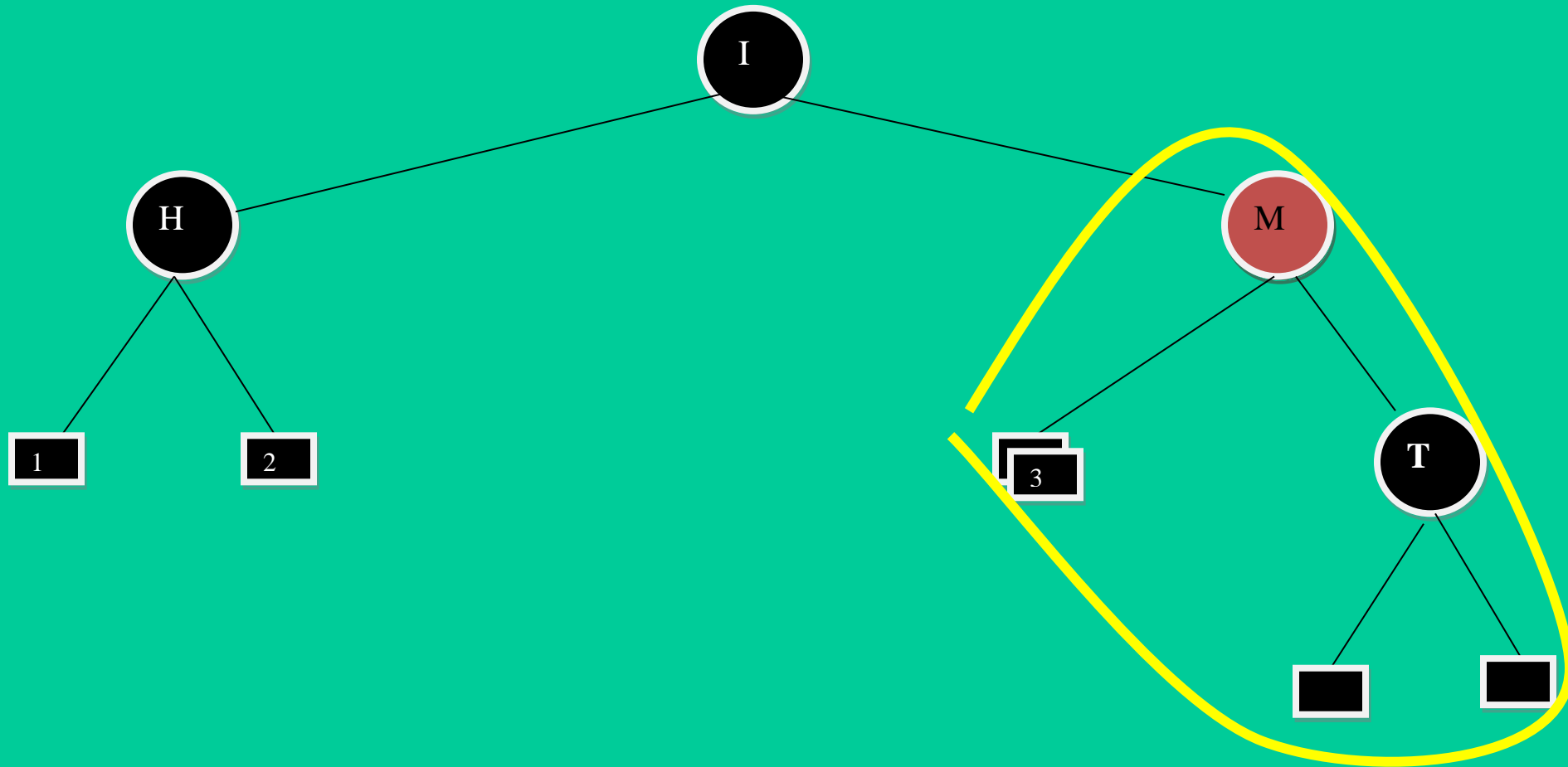
Acte1 : M remplace R et devient rouge

M supprimé

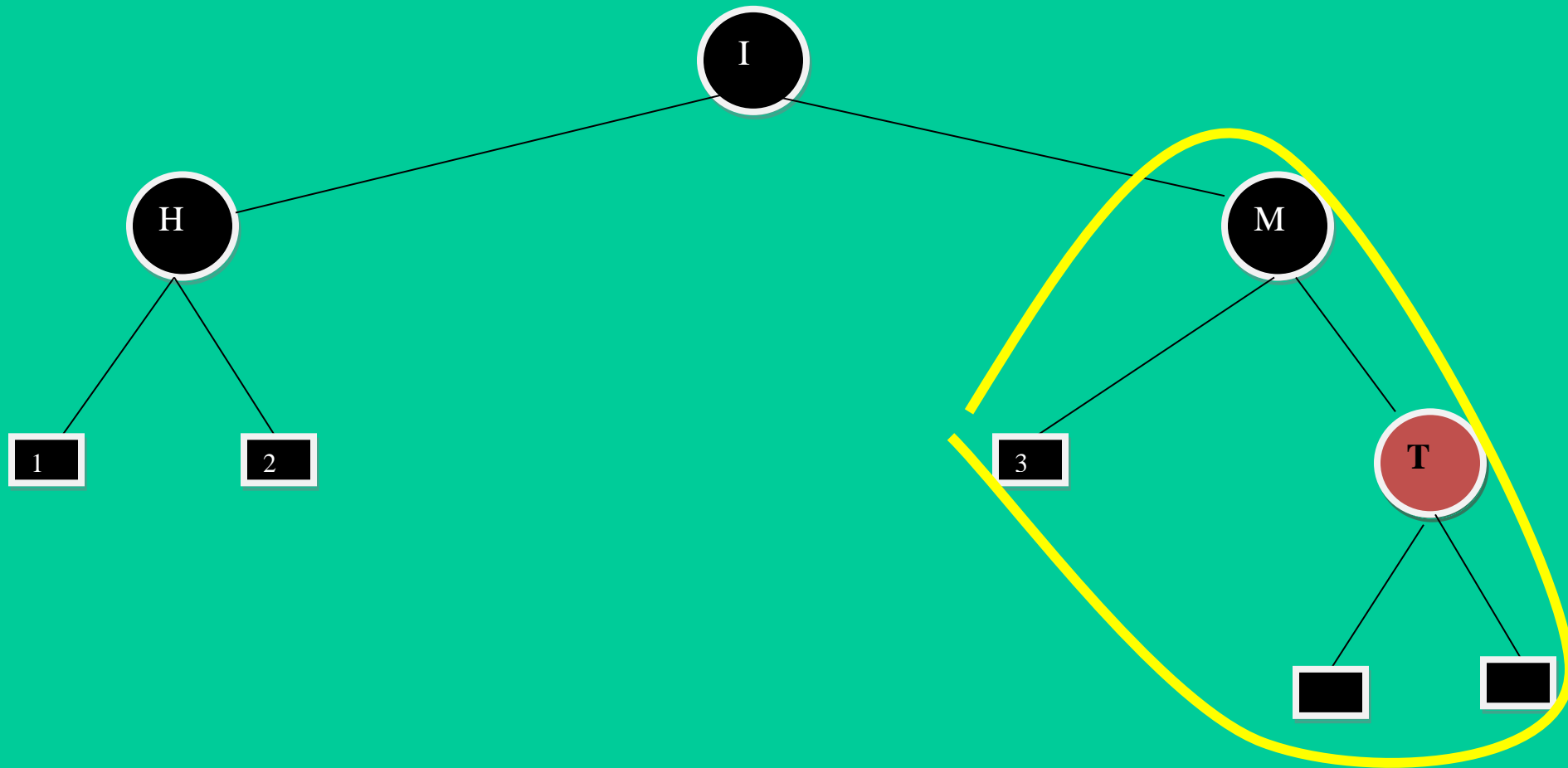
La sentinelle 3 remplace M et devient noir²



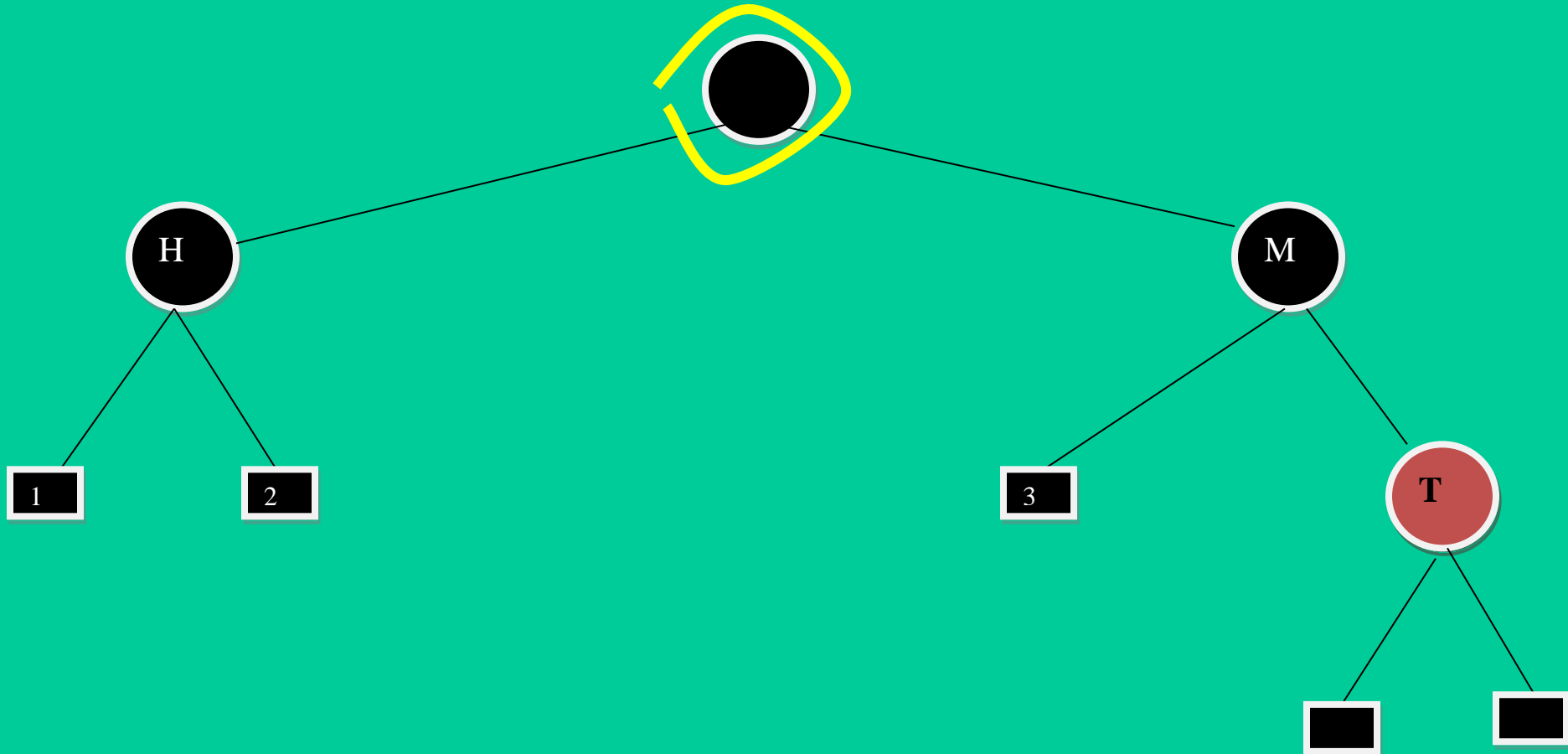
Acte 2: appliquer cas 1.a



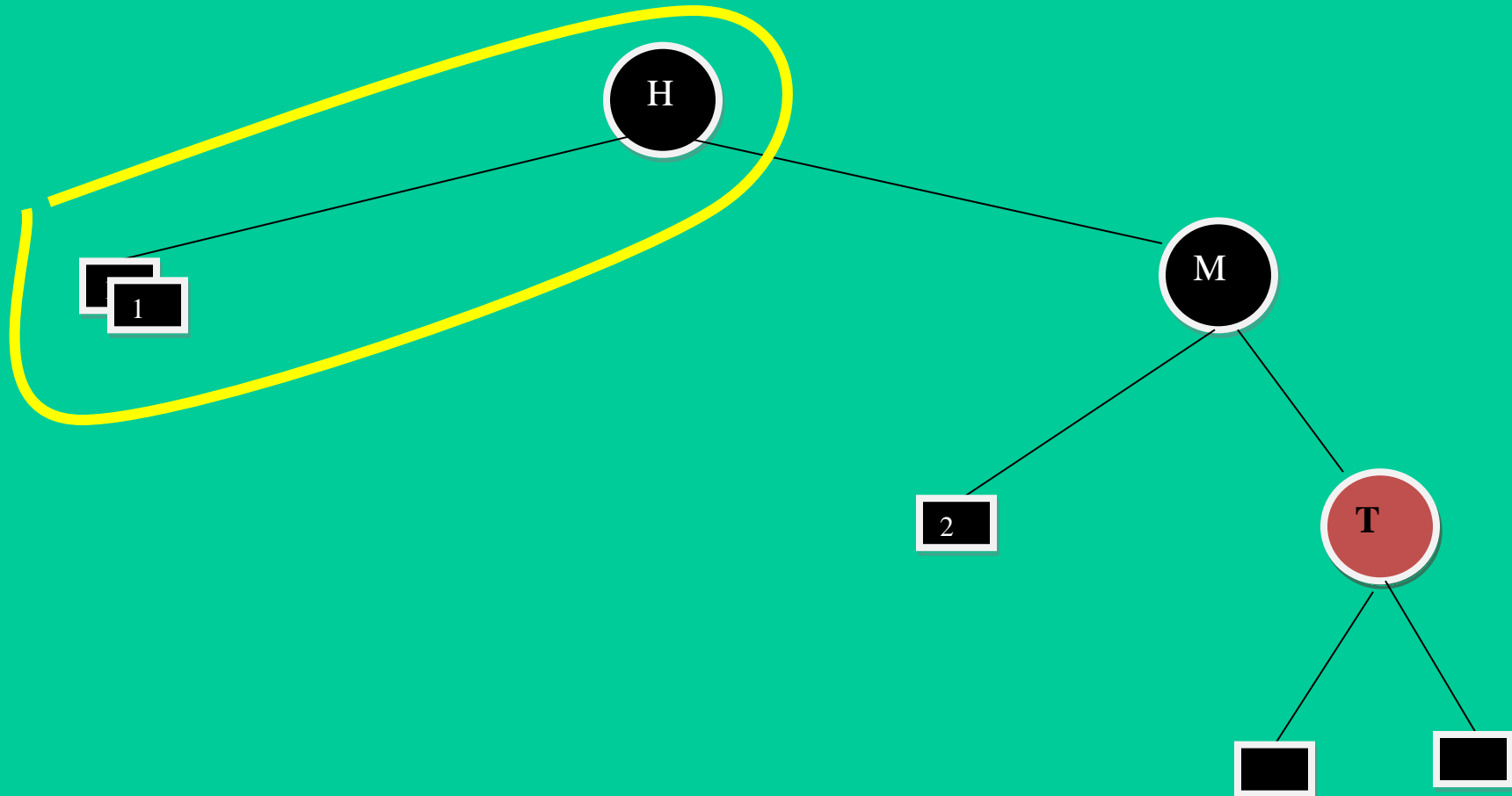
Résultat



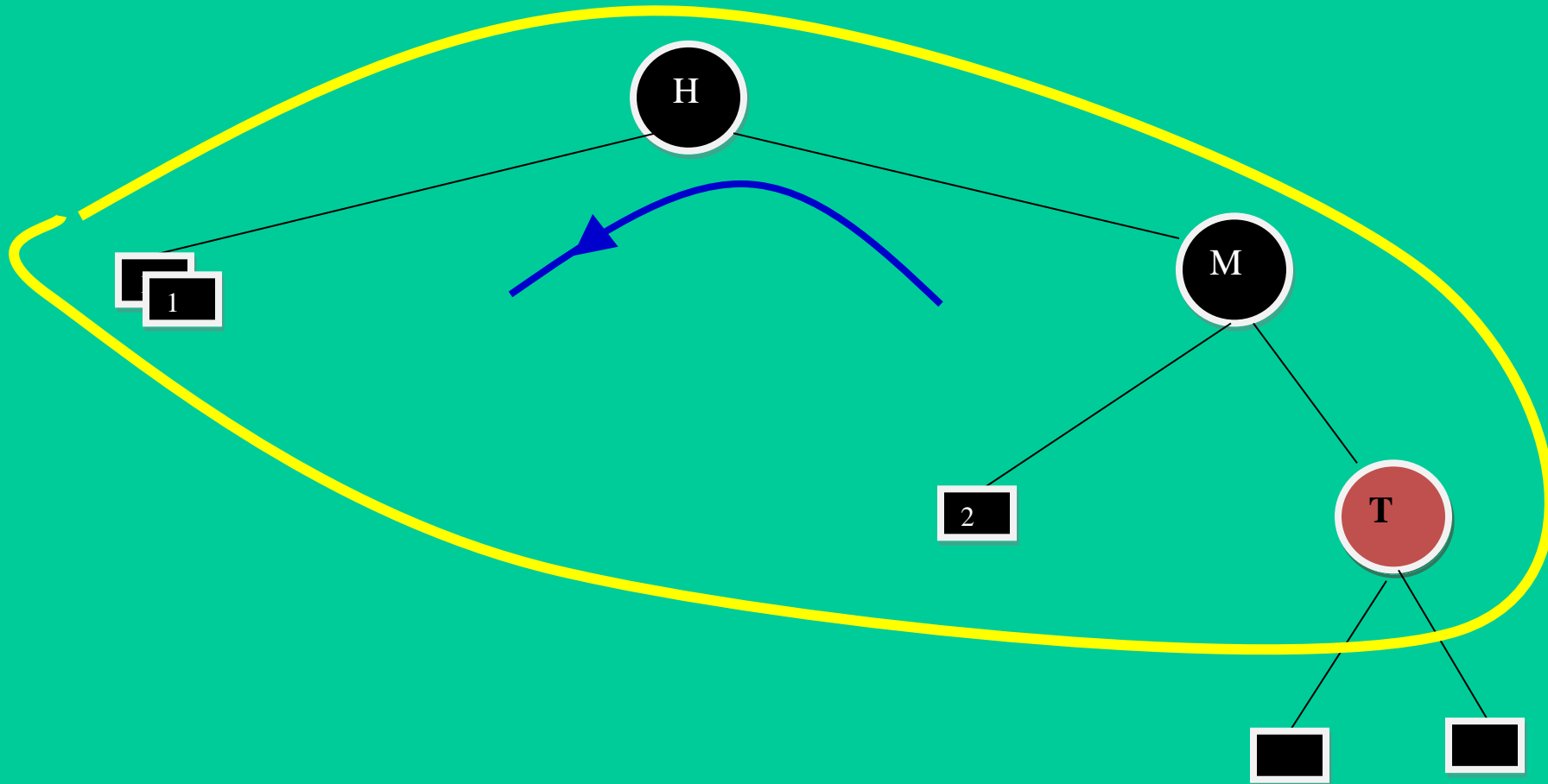
Acte 0 supprimer I : I ne sera pas supprimé



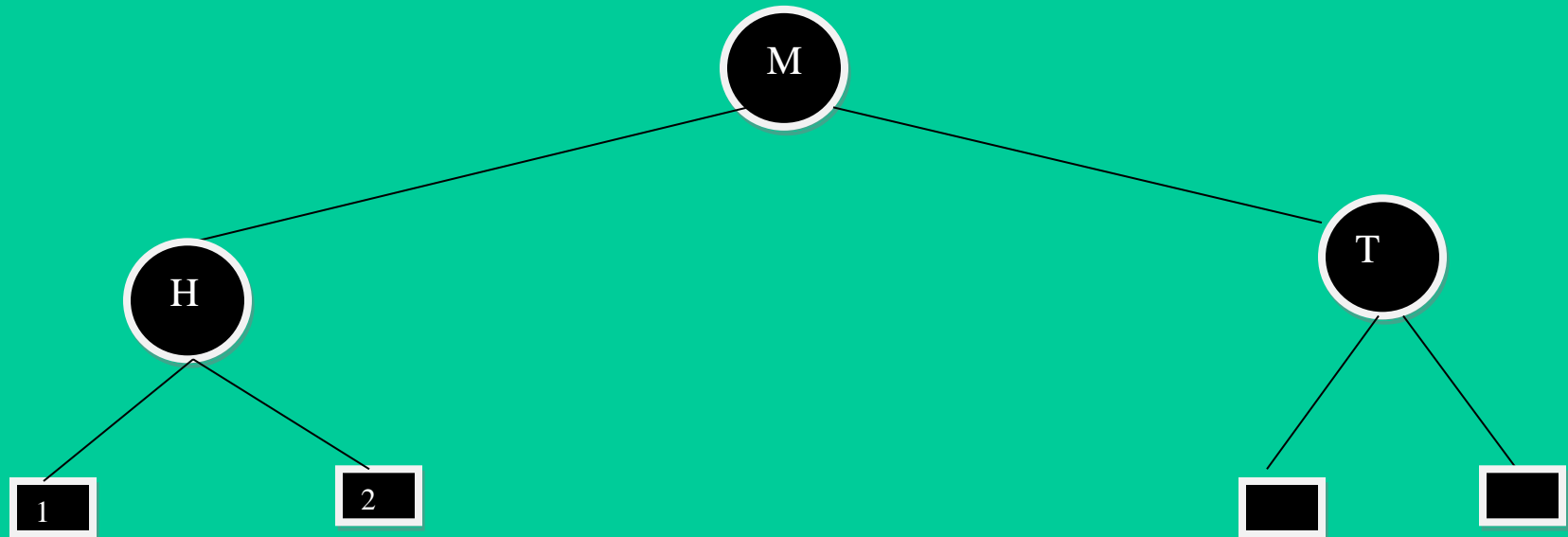
Acte1 : I est remplacé par H lequel sera supprimé
La sentinelle 1 remplace H et devient **noir**²



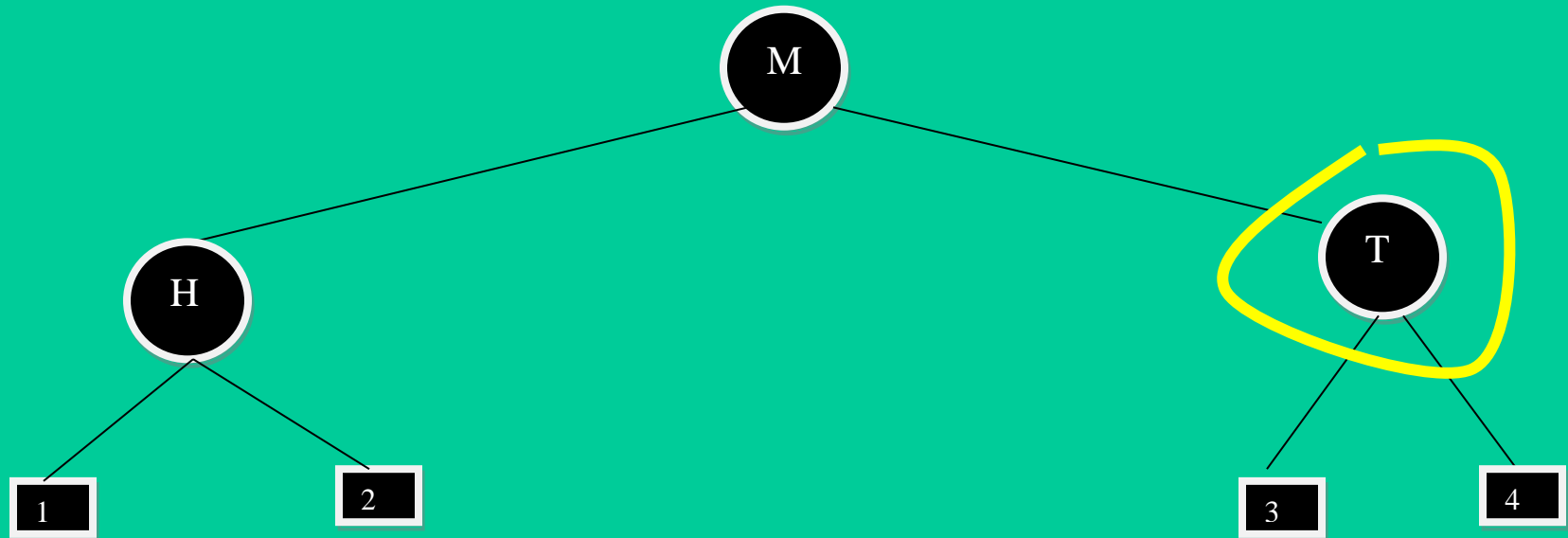
Acte 2 : appliquer cas 1.b



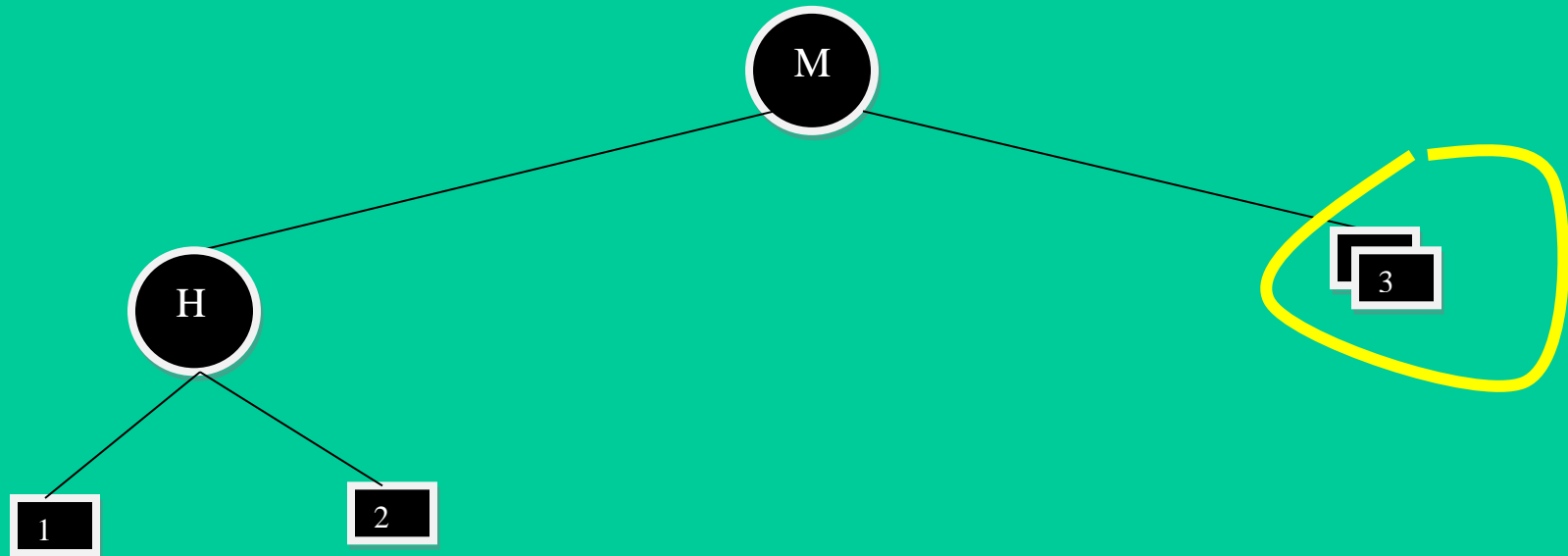
Résultat :



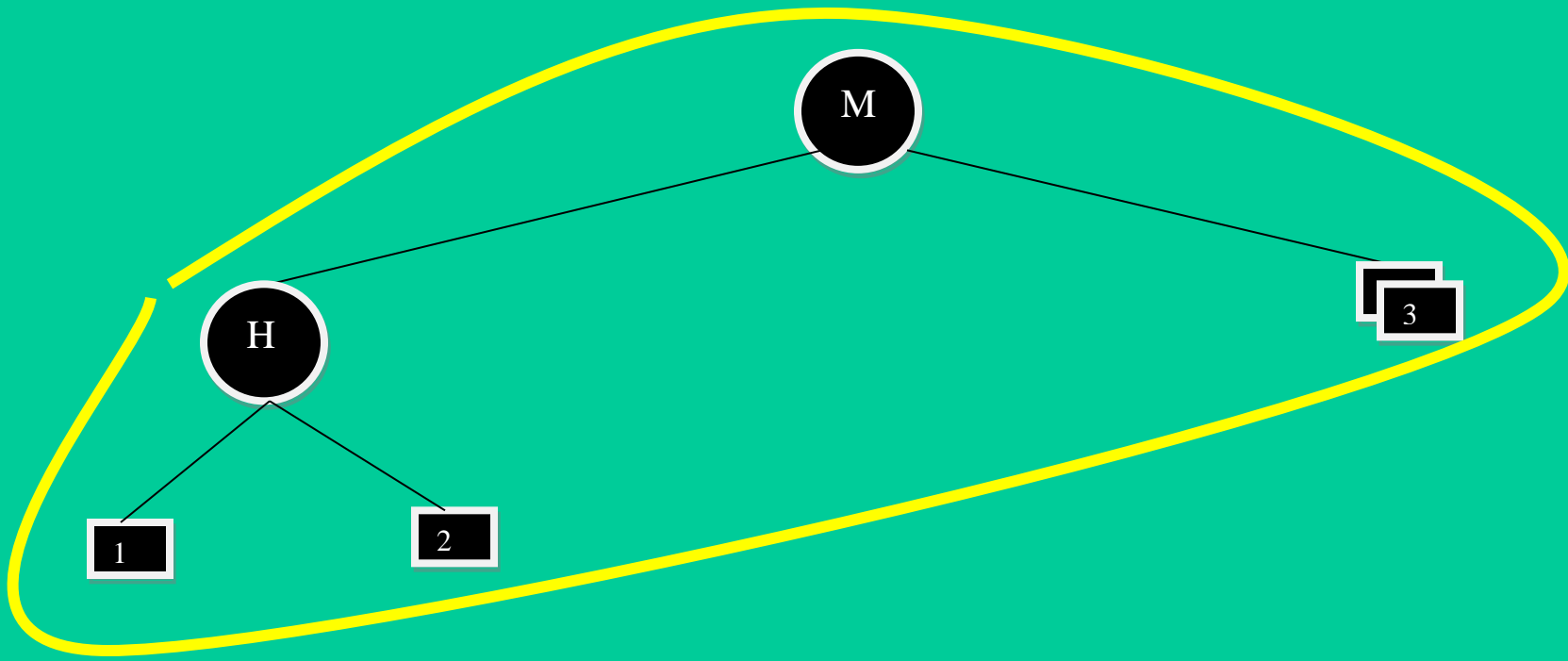
Acte 0 : supprimer T: T sera supprimé



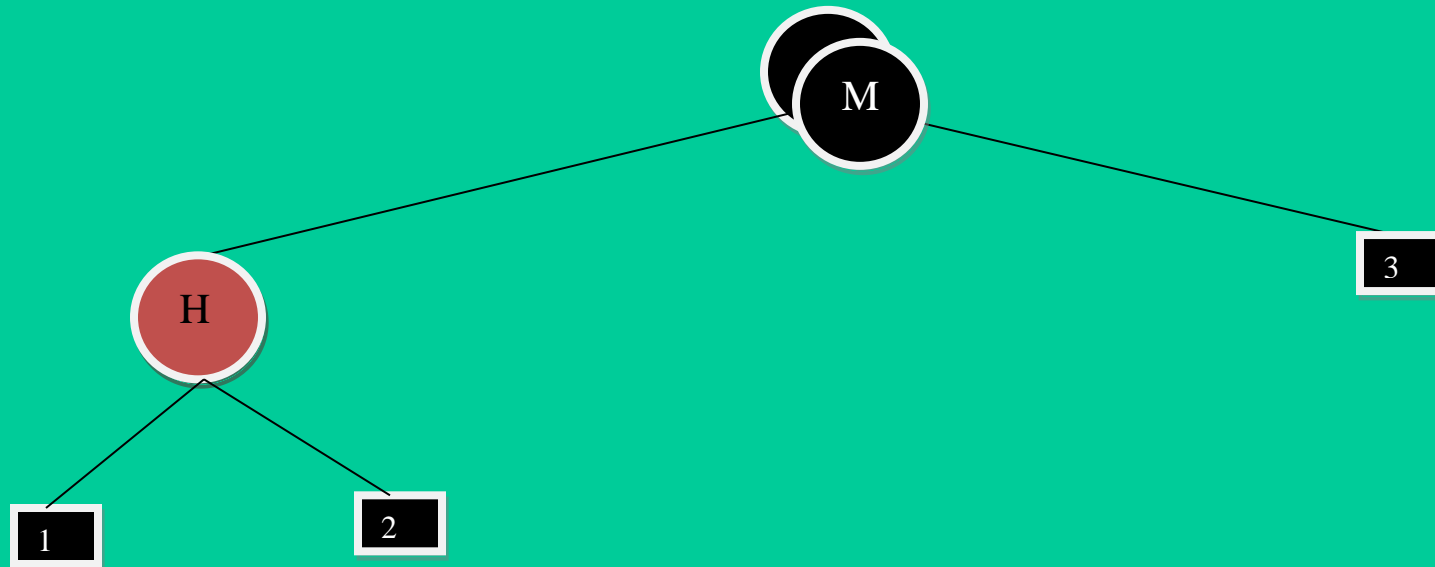
Acte 1 : La sentinelle 3 qui remplace T devient **noir**⁺



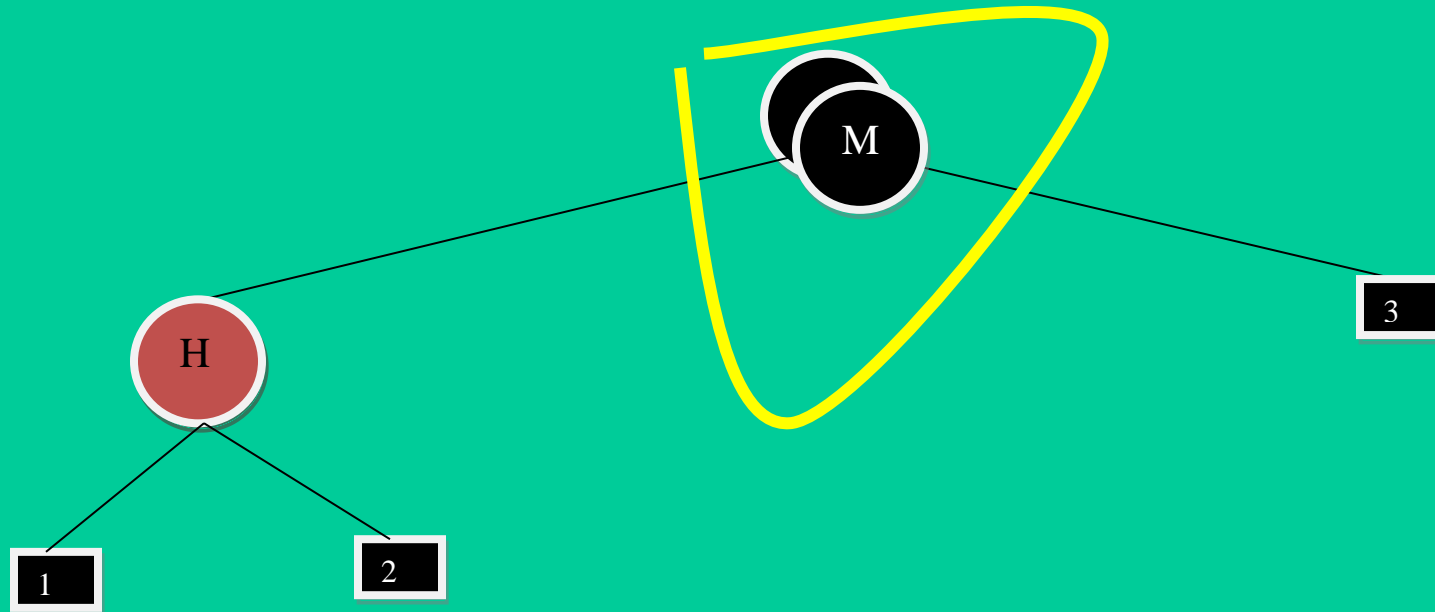
Acte 2 : appliquer cas 1.a



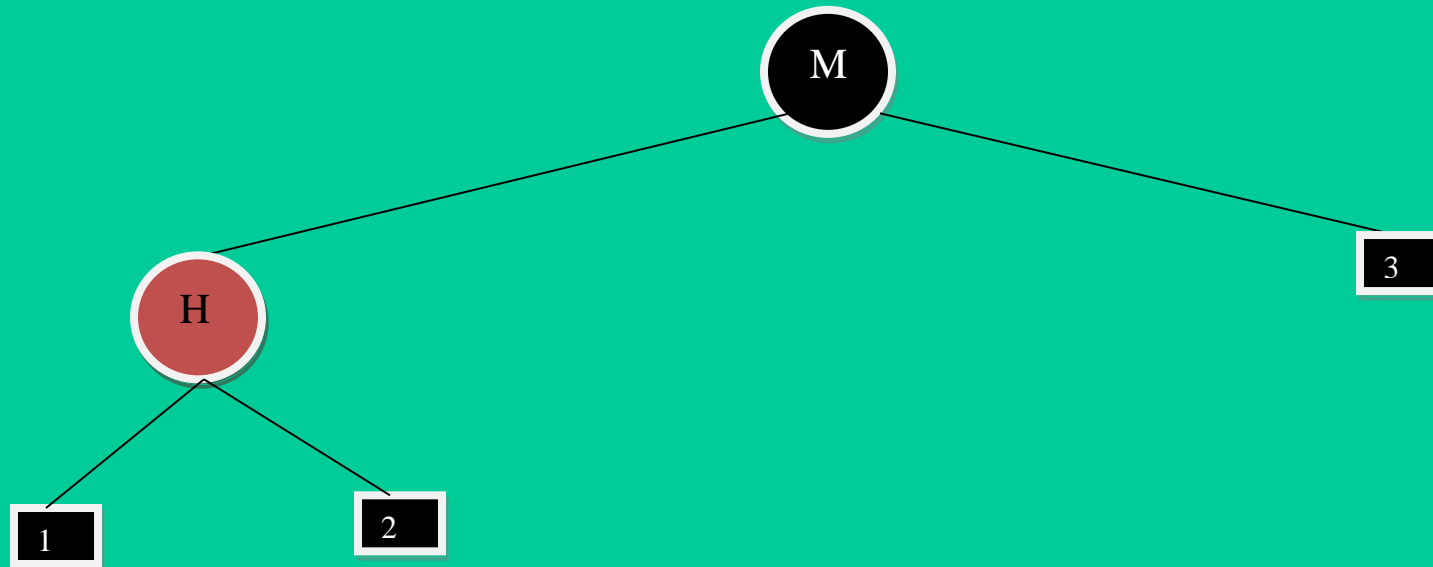
Résultat :



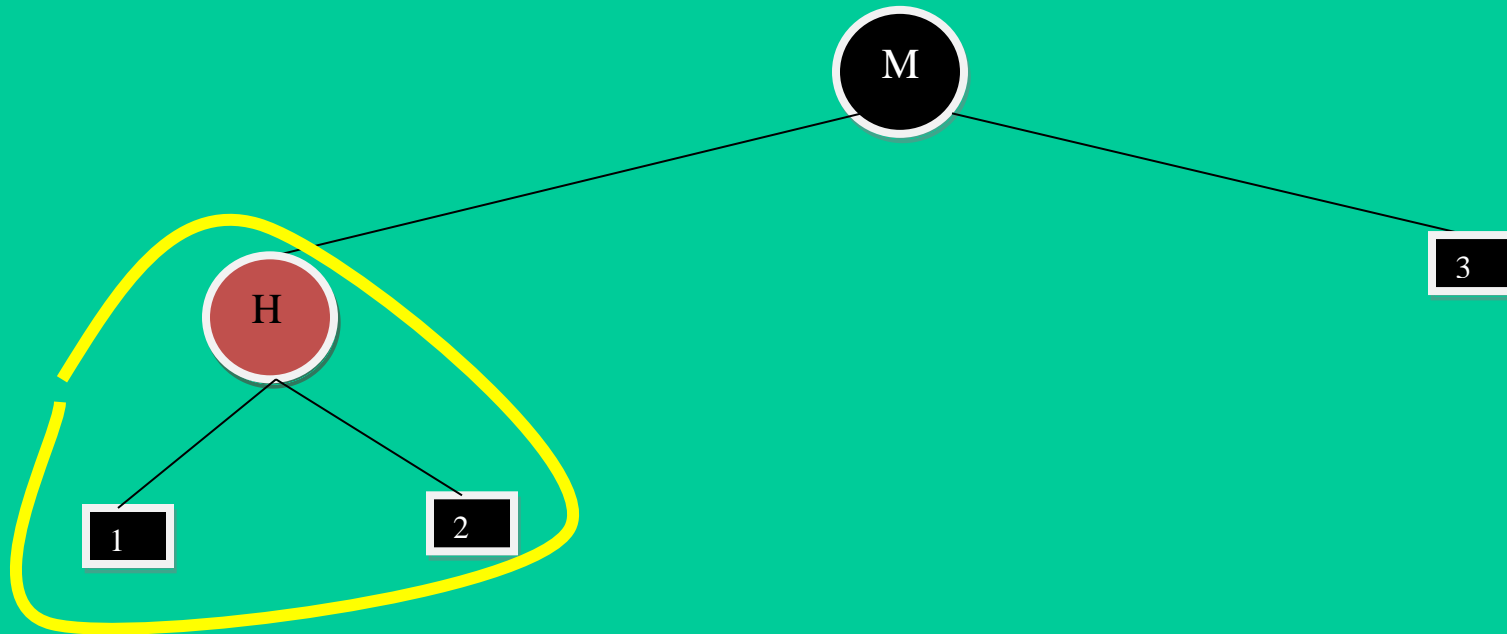
Acte3 : dans la foulée appliquer cas 0



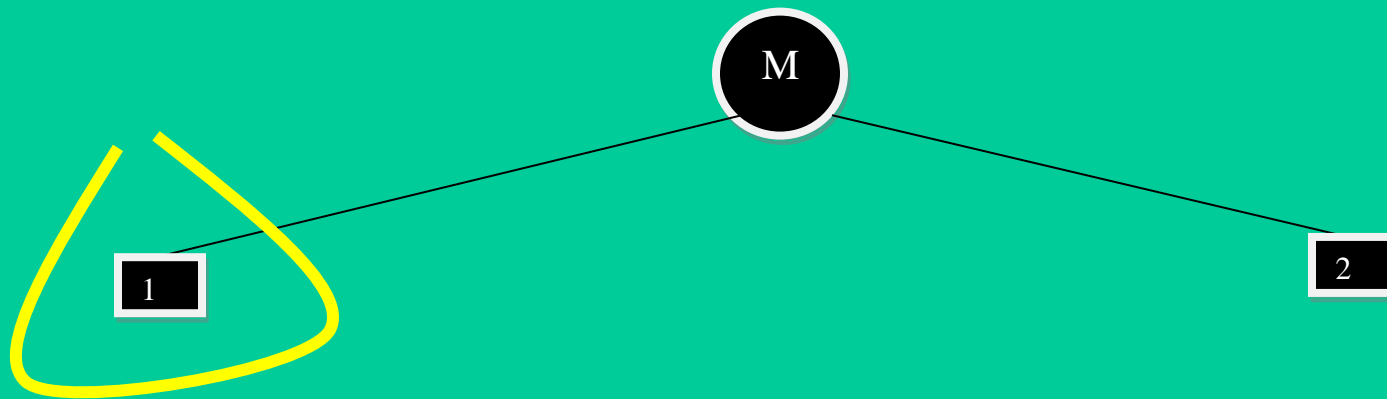
Résultat : M devient noir



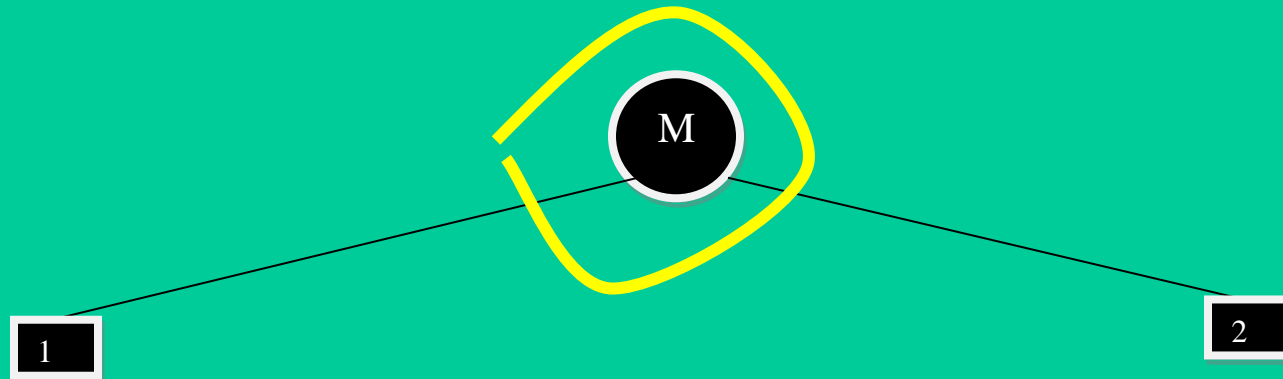
Acte 0 :supprimer **H**: **H** sera supprimé



Résultat : **H** est supprimé sans conséquence
H est remplacé par la sentinelle 1

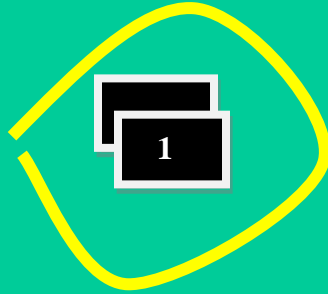


Acte 0 :supprimer **M** : **M** sera supprimé



Acte 1 : M est supprimé sans conséquence

La sentinelle 1 qui remplace M devient noir⁺



Acte 2 : appliquer cas 0

