

Les Structures de Données

Les arbres AVL

Pr F.Omary

2019-2020

Introduction

► Pourquoi

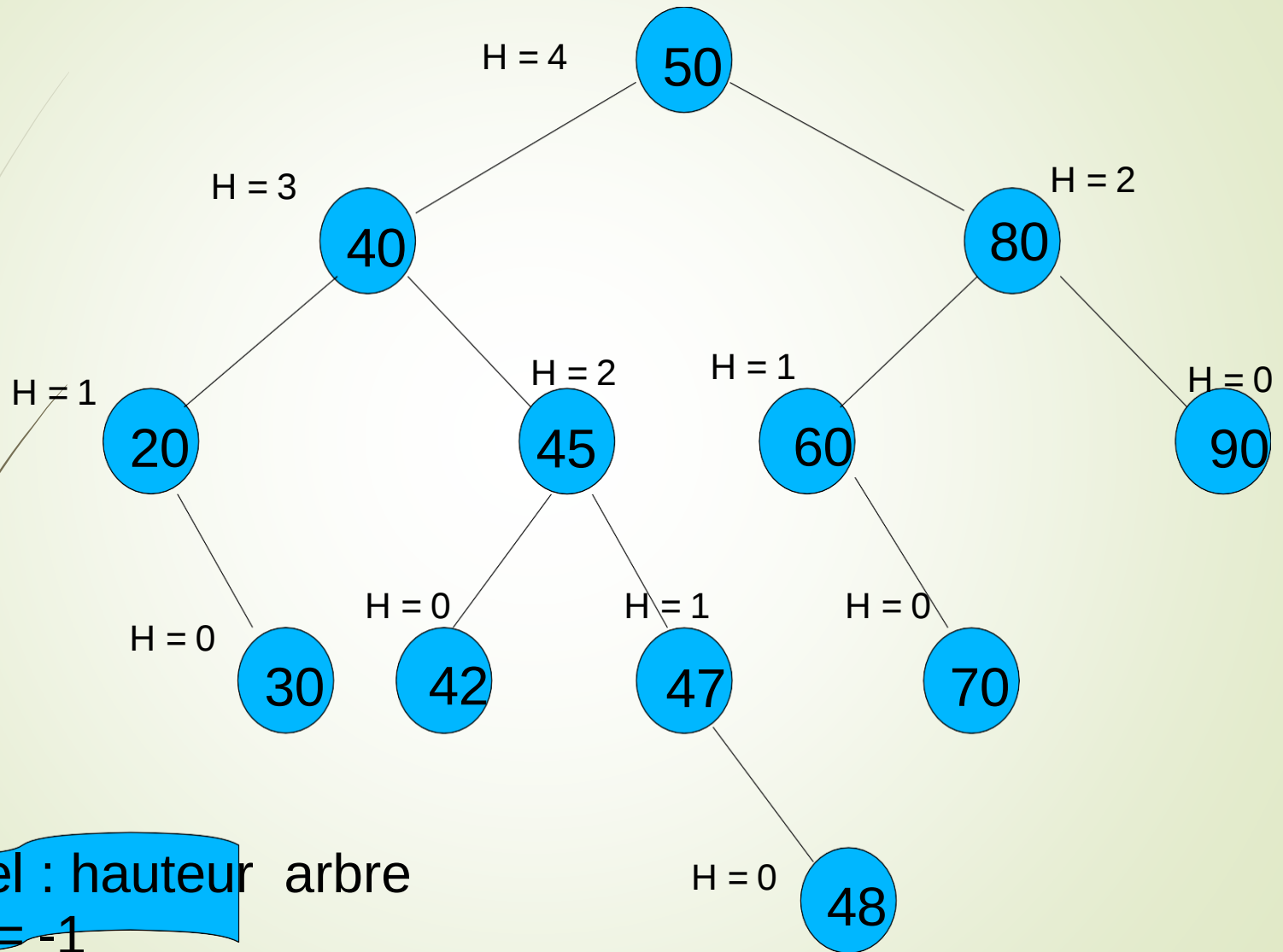
- Les arbres équilibrés rendent les recherches plus efficaces
- Trouver comment maintenir un arbre relativement équilibré au fur et à mesure des insertions (et suppression)

► Solution :

- Les arbres AVL (Adelson-Velskii et Landis) : pour tout sommet, les hauteurs des sous- arbres gauche et droit diffèrent d'au plus 1.
- Rmq : un arbre AVL N'EST PAS un arbre équilibré

Exemple d'arbre AVL

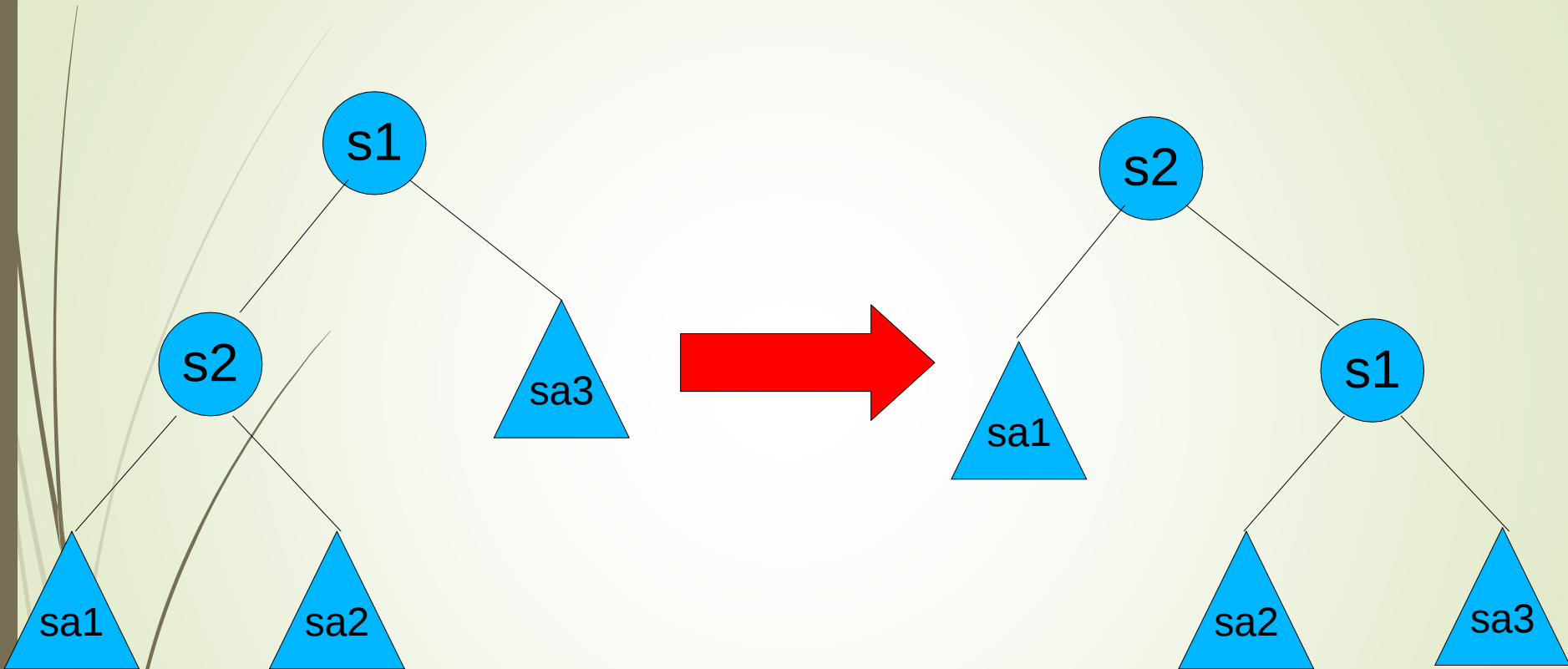
3



Rappel : hauteur arbre
vide = -1

Préambule : Rotation droite

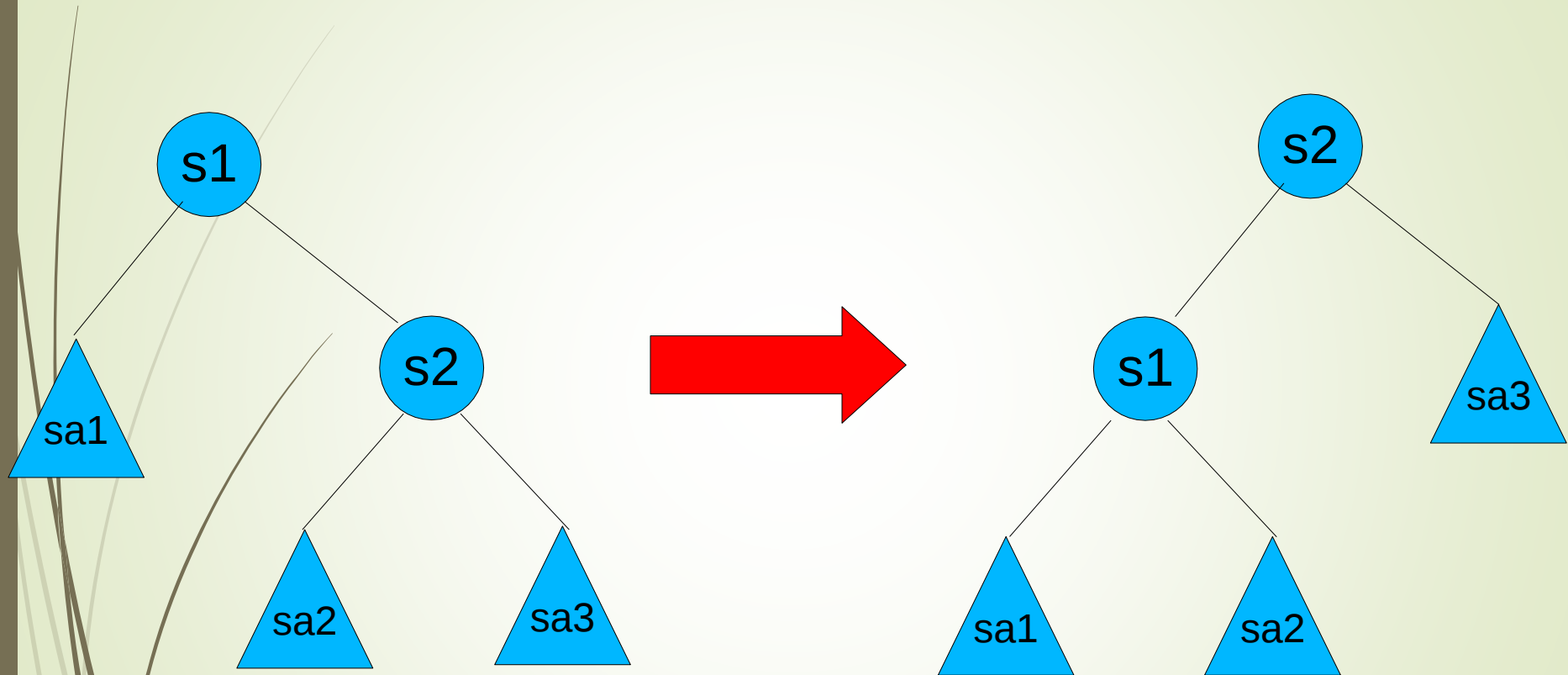
4



Rotation droite autour de $S1$, notée $rd(S1)$

Préambule : Rotation gauche

5



Insertion dans un arbre AVL

- Le principe de l'insertion dans un arbre AVL est le suivant :
 - ↖ insérer le nouveau nœud au bon endroit
 - ↖ au fur et à mesure de la remontée dans l'arbre (du nœud père du nœud inséré à la racine), rééquilibrer l'arbre en effectuant les rotations appropriées

Choix des rotations

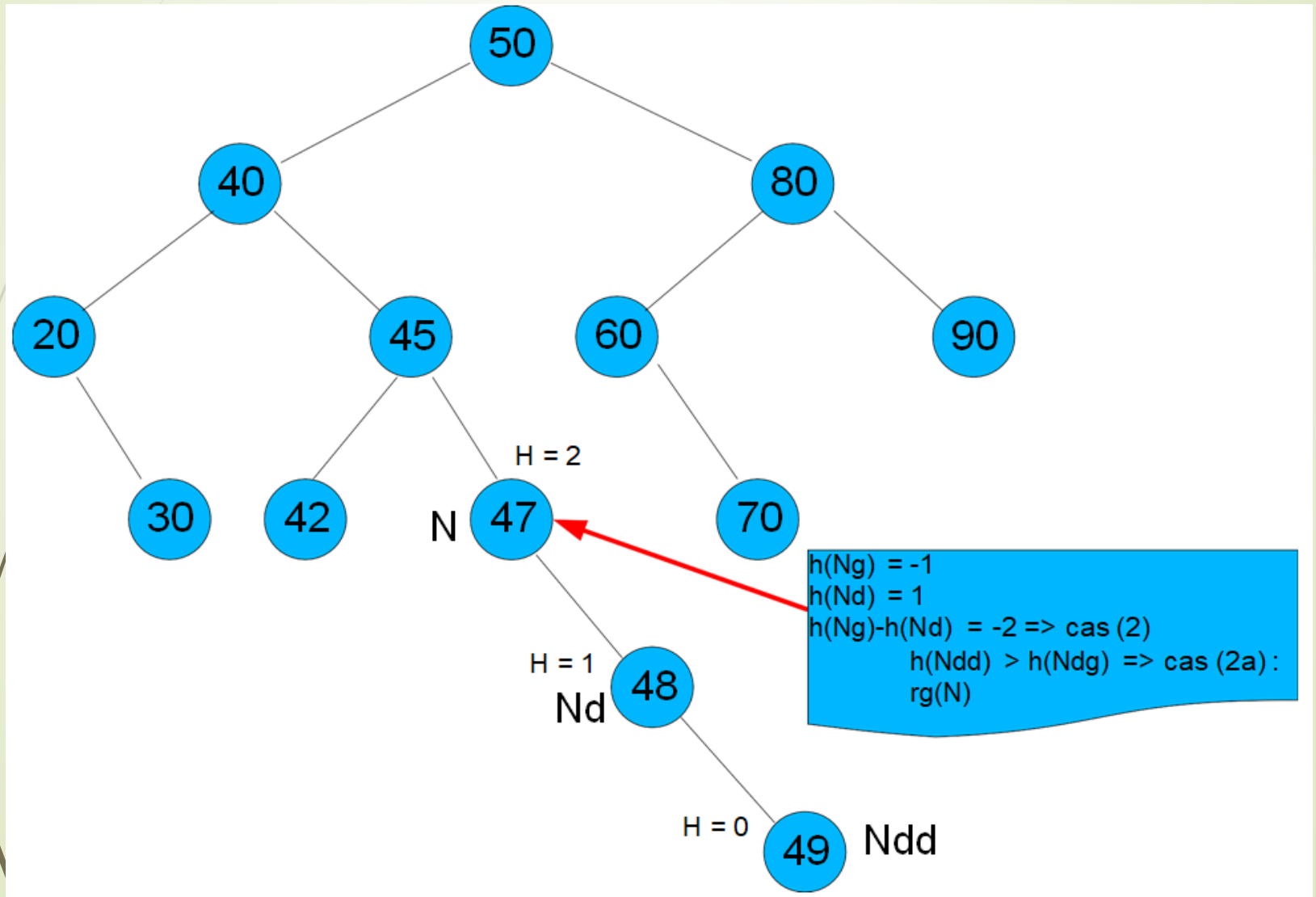
Notations

- Soient N le noeud courant, N_g son fils gauche et N_d son fils droit.
- Soient N_{gg} le fils gauche de N_g et N_{gd} le fils droit de N_g
Soient N_{dg} le fils gauche de N_d et N_{dd} le fils droit de N_d
- Soit $h(x)$ la hauteur de l'arbre de racine le noeud x .

Algorithme

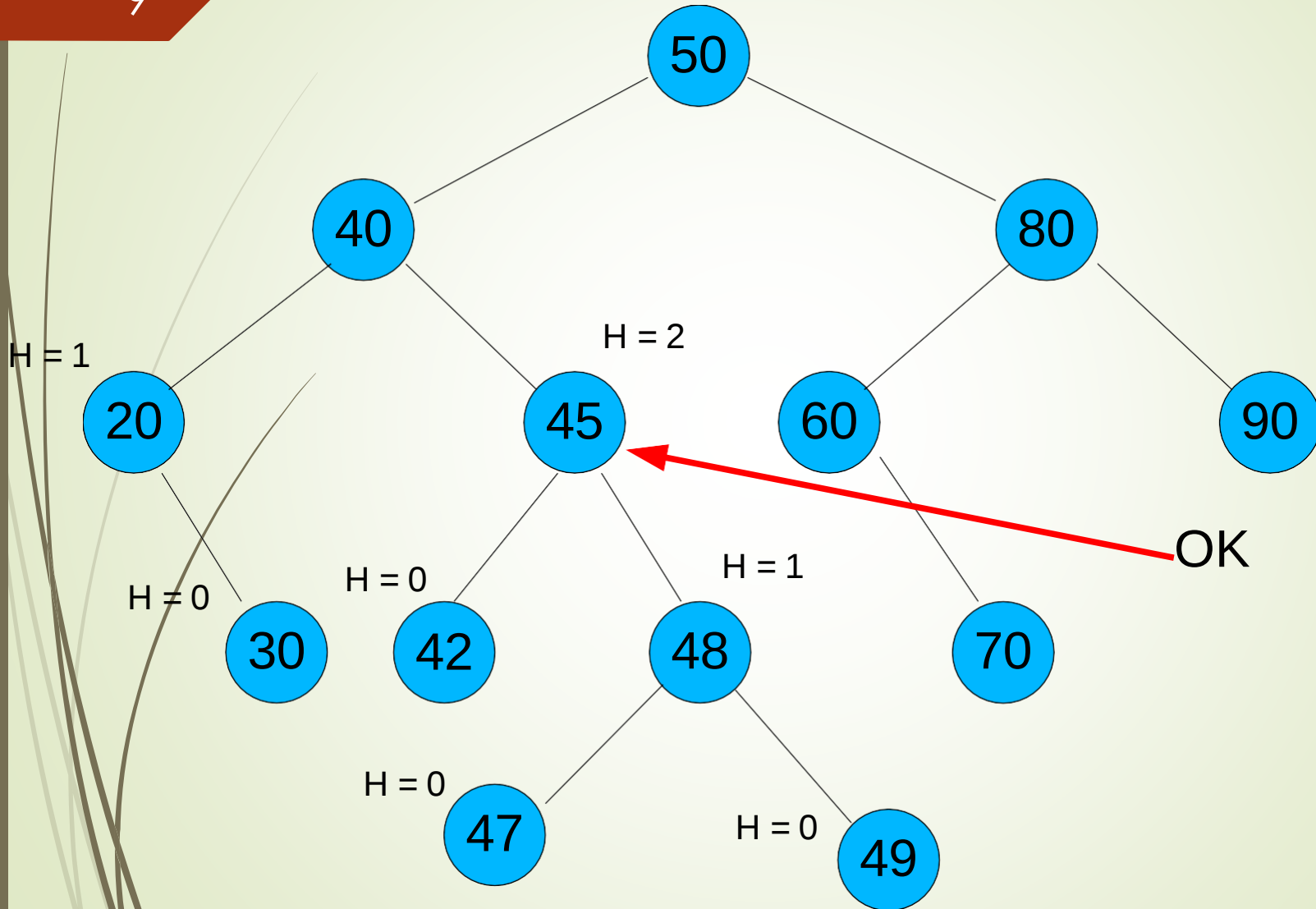
- Si $|h(N_g) - h(N_d)| \leq 1$, ne rien faire
- Sinon
 - Si $h(N_g) - h(N_d) = 2$ **cas (1)**
 - Si $h(N_{gg}) > h(N_{gd})$ Alors $rd(N)$ **cas (1a)**
 - Sinon $rg(N_g)$ puis $rd(N)$ **cas (1b)**
 - Sinon $(h(N_g) - h(N_d) = -2)$ **cas (2)**
 - Si $h(N_{dd}) > h(N_{dg})$ Alors $rg(N)$ **cas (2a)**
 - Sinon $rd(N_d)$ puis $rg(N)$ **cas (2b)**
- Fsi
- Fsi

Exemple d'ajout : 49



Exemple d'ajout : 49

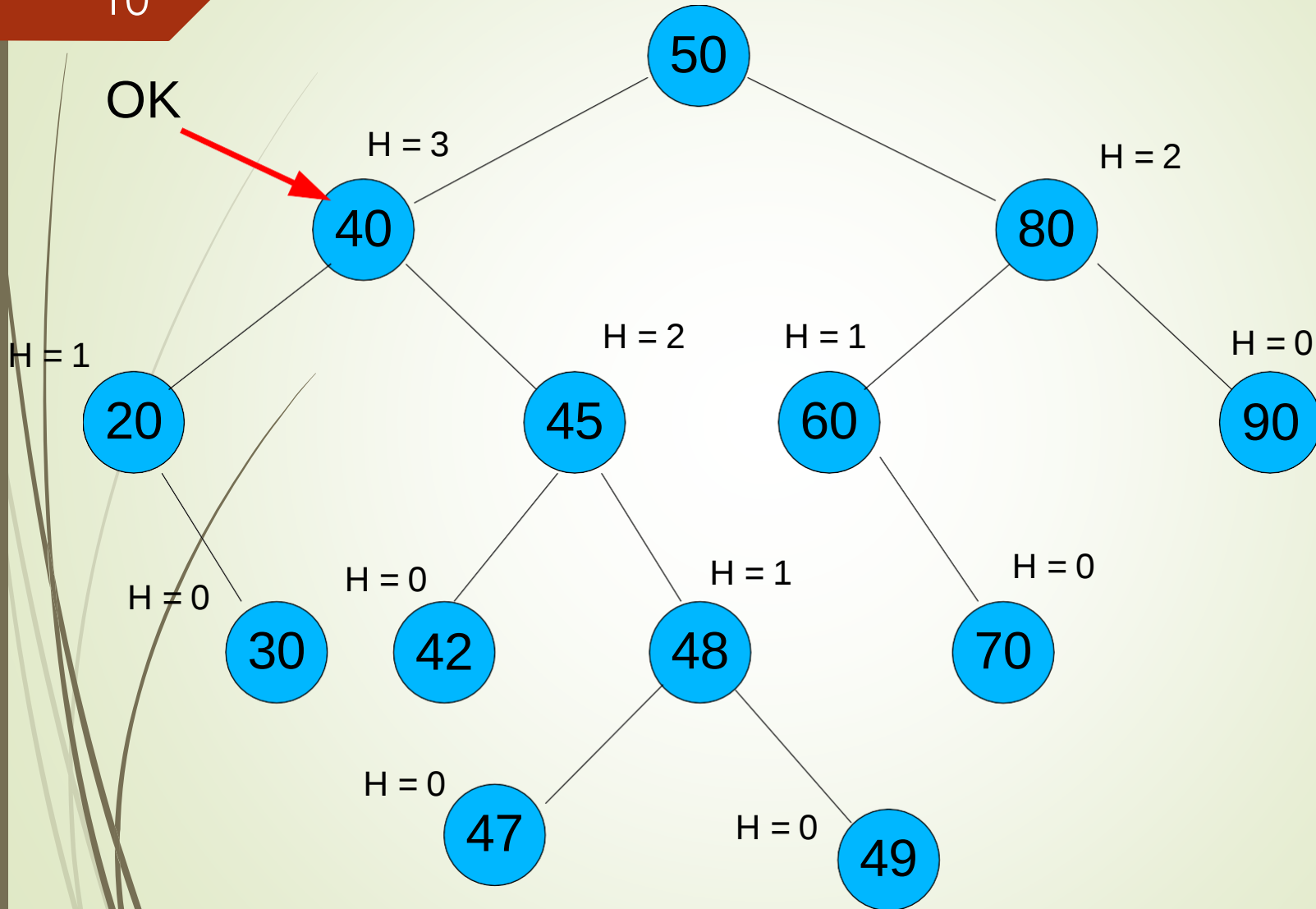
9



Exemple d'ajout : 49

10

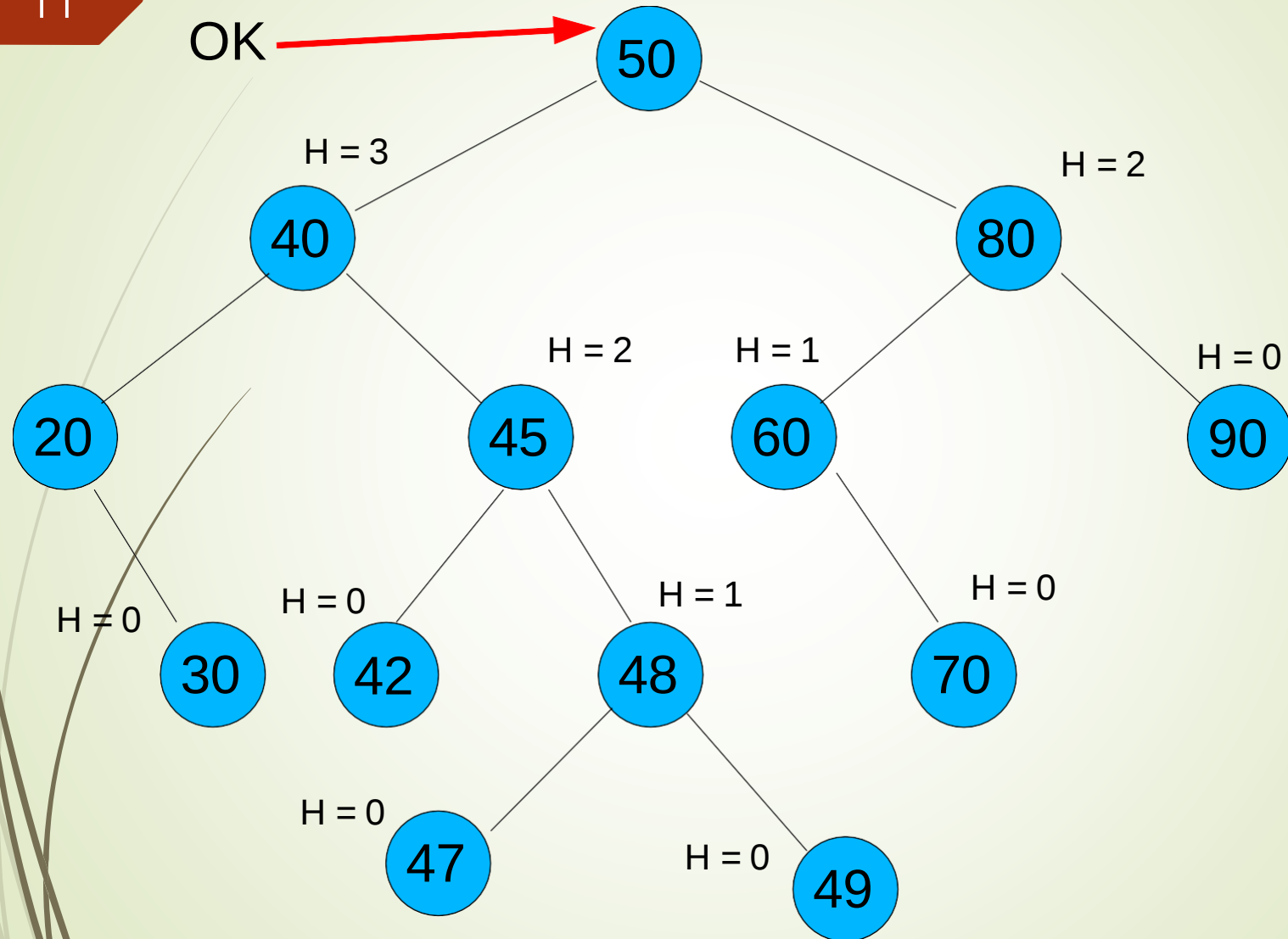
OK



Exemple d'ajout : 49

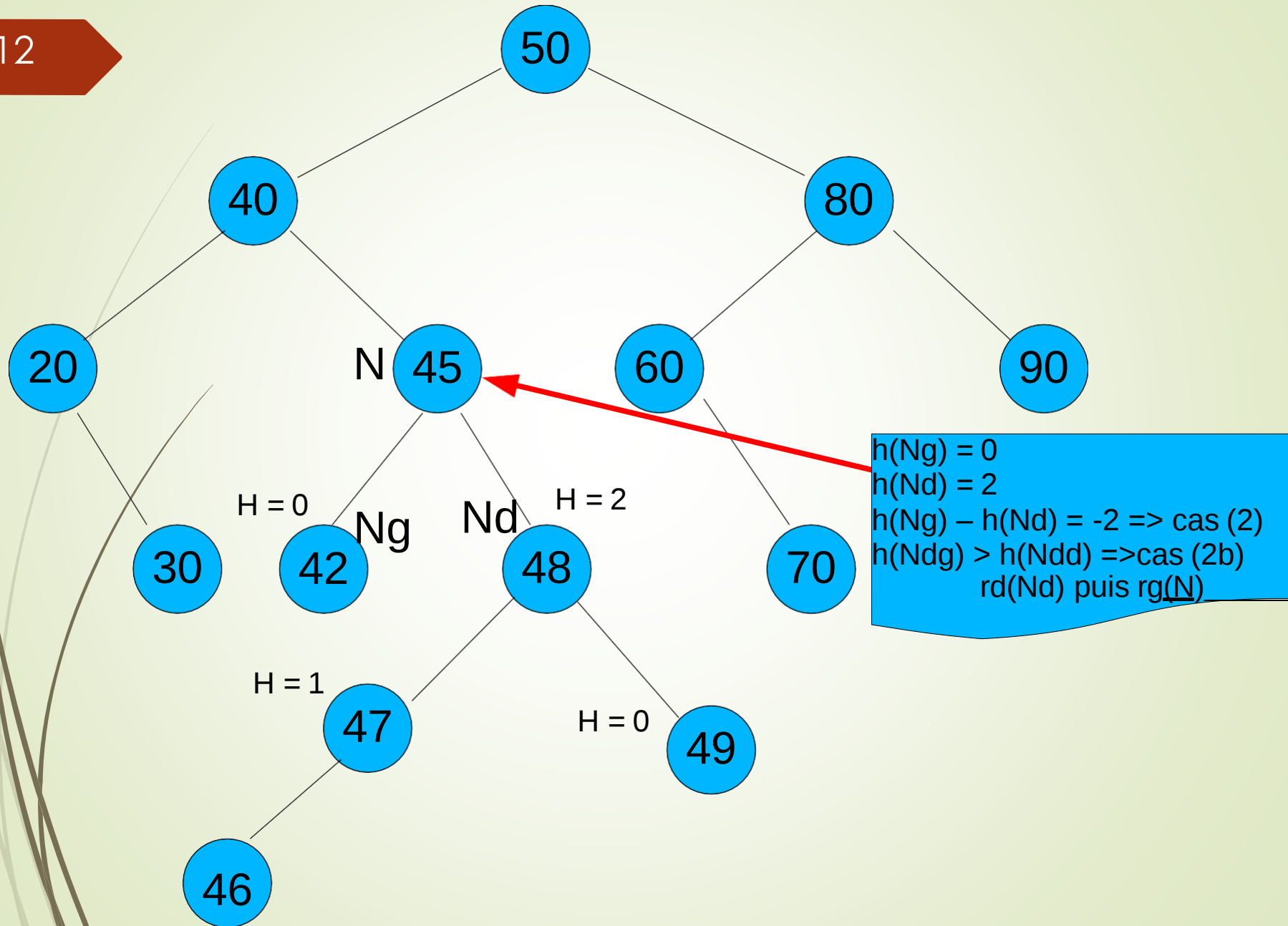
11

OK



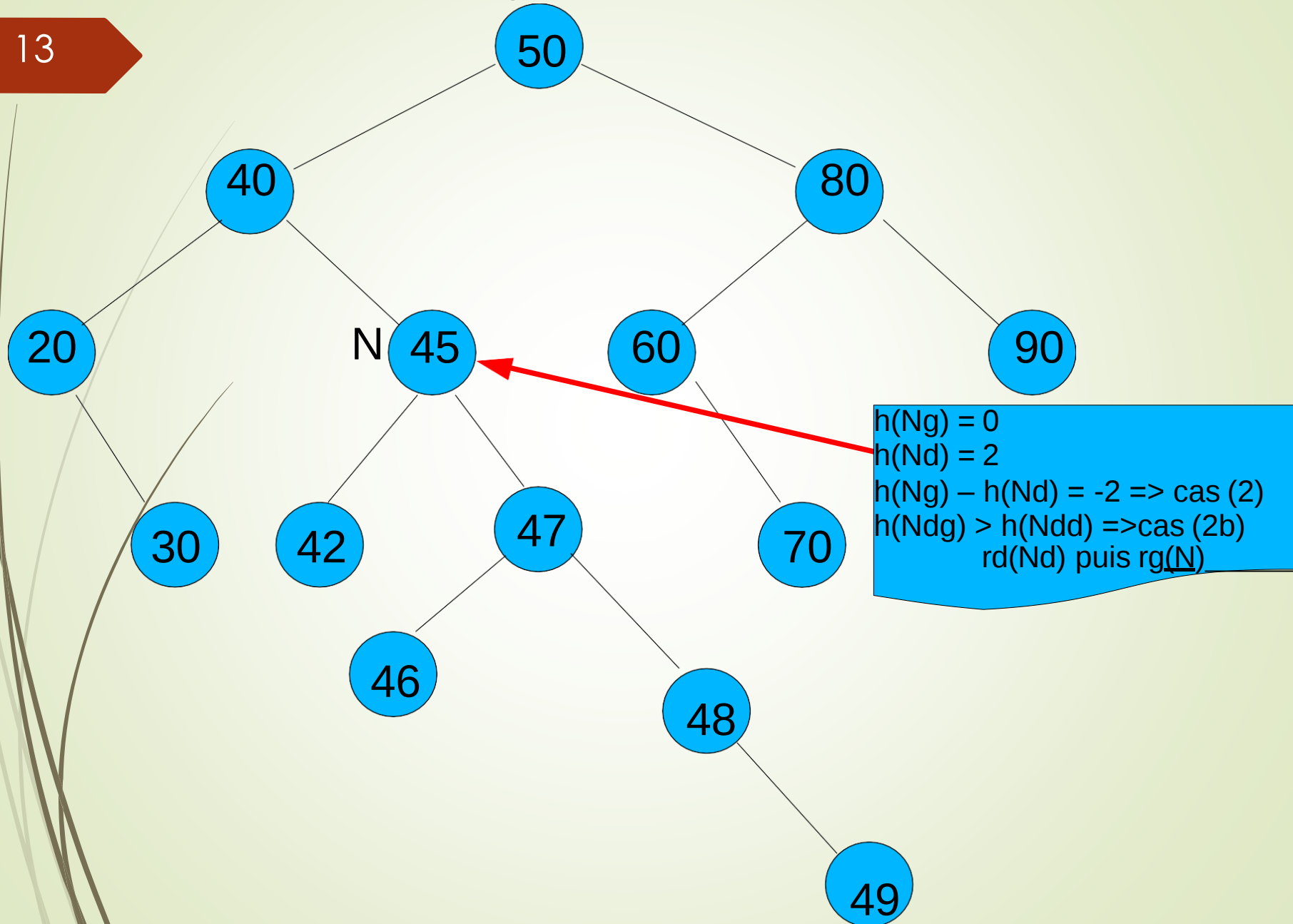
Exemple d'ajout : 46

12



Exemple d'ajout : 46

13



Exemple d'ajout : 46

