Association France-IOI www.france-ioi.org

Indications pour le sujet "maintenir le maximum"

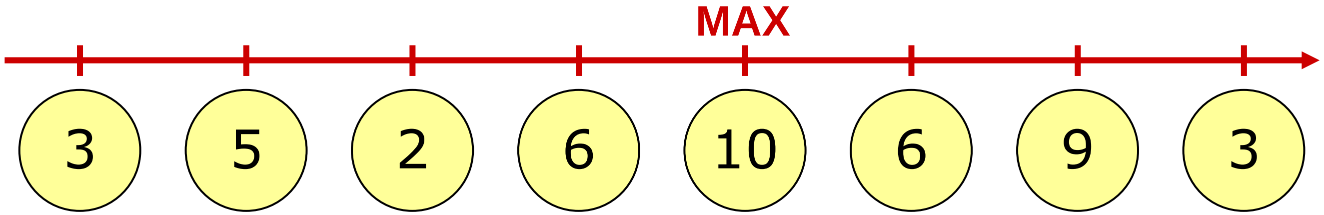
Arthur Charguéraud

Décembre 2005

Chercher le maximum

Situation : on dispose de N nombres.

Question : quel est la valeur du maximum ?



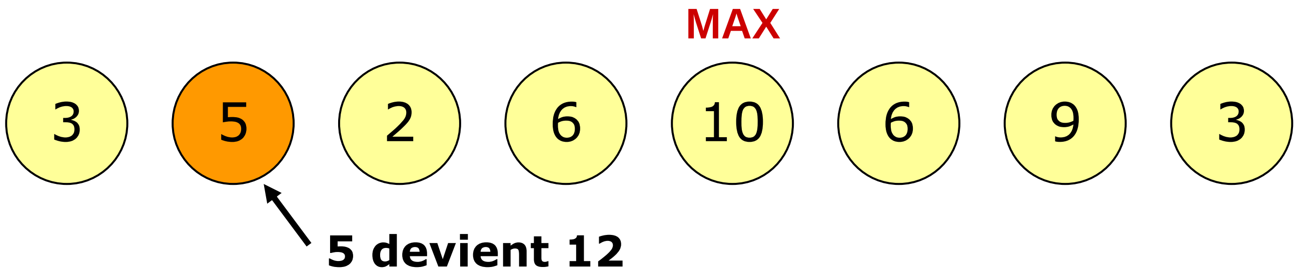
Solution : on parcourt les N valeurs.

Coût : environ N comparaisons (N-1 exactement).

Maintenir le maximum

Dynamique : on change des valeurs arbitrairement.

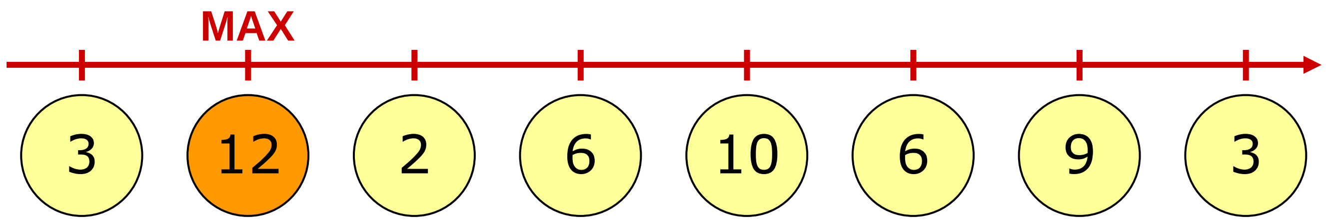
Avant :



Question : quel est la nouvelle valeur du maximum ?

Solution : reparcourir les N valeurs pour le trouver.

Après :



Question : comment faire plus efficace que cela ?

Trois cas possibles

cas 1 : la nouvelle valeur est le nouveau maximum.

3 **5** 2 6 10 6 9 3

5 devient 12 MAX

cas 2 : la valeur retirée n'est pas le maximum et la nouvelle valeur est plus petite que le maximum.

3 **5** 2 6 10 6 9 3

5 devient 7 MAX

cas 3 : on diminue la valeur du maximum.

3 5 2 6 **10** 6 9 3

10 devient 7

Pire cas

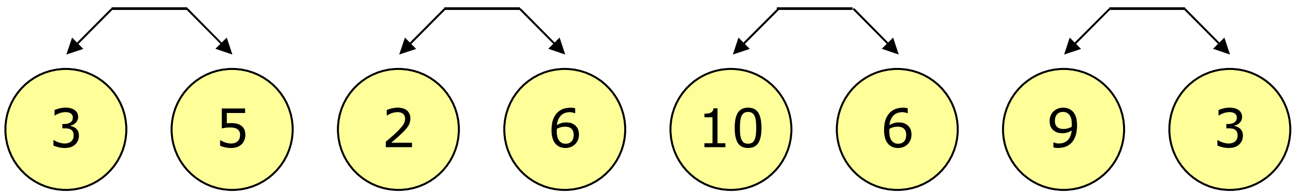
Complexité : le cas 1 et 2 sont faciles à gérer, car on connait facilement la nouvelle valeur du maximum : c'est soit la même qu'avant, soit la valeur que l'on vient d'apporter. En revanche pour le cas 3, c'est beaucoup plus délicat.

Si les modifications étaient aléatoires, le cas 3 n'arriverait pas très souvent, et tout irait bien.

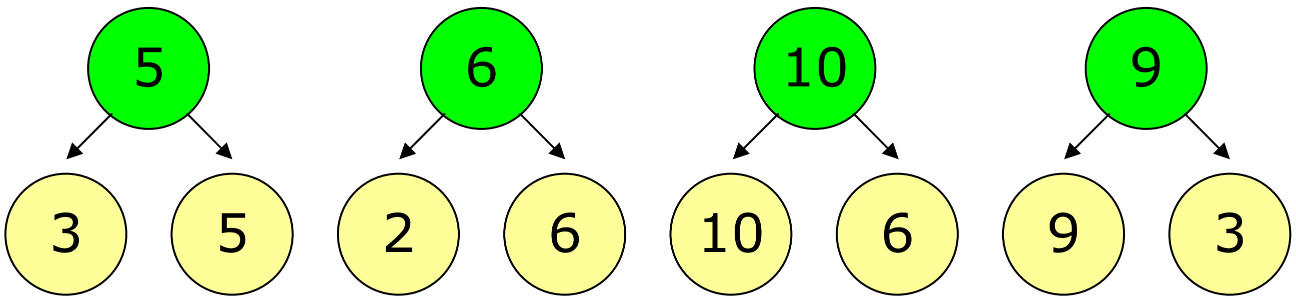
Le problème c'est que c'est justement cette opération qui consiste à changer la valeur du maximum que l'on souhaite effectuer à chaque fois ! Il faut donc que l'on trouve une méthode qui traite efficacement tous les cas.

Pour faire plus efficace

Idée 1 : on va regrouper les valeurs deux par deux.

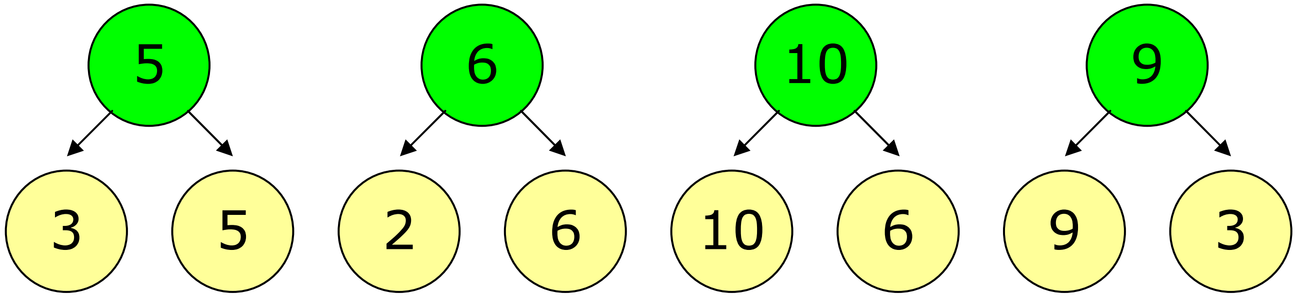


Idée 2 : pour chaque paire, on retient le maximum.

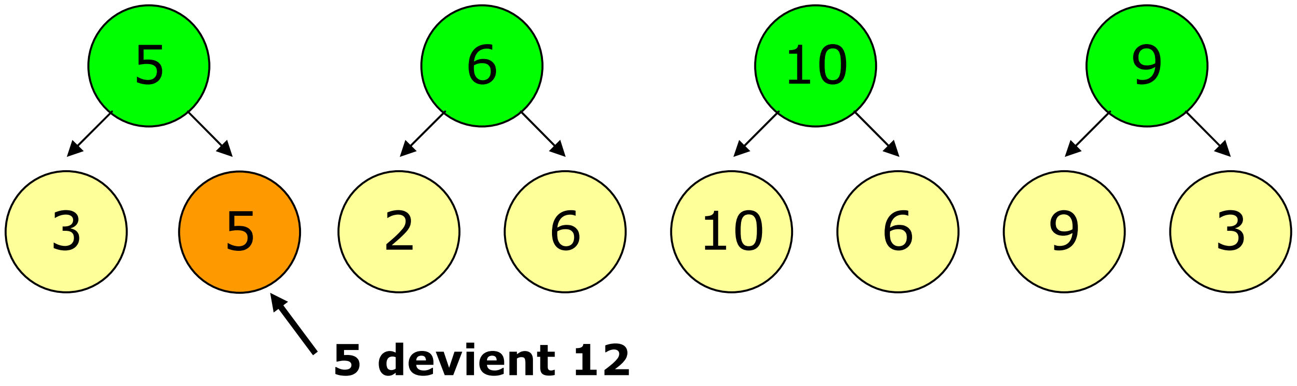


Question : comment cela peut-il nous aider ?

Questions auxquels il faut répondre



max : combien d'opérations pour trouver le max ?

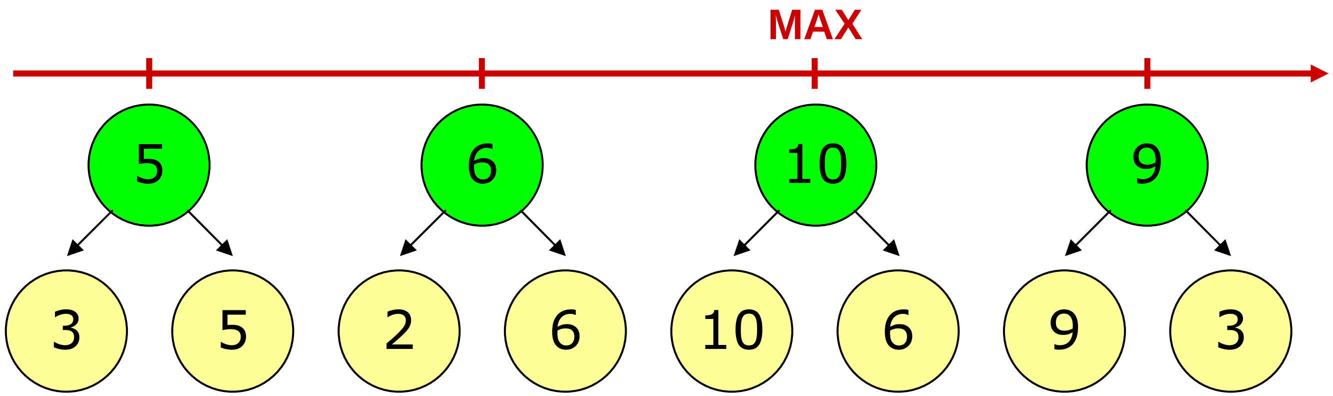


modif : combien d'opérations pour maintenir la structure lorsqu'on change une valeur ?

Trouver le maximum

Question : comment faire pour trouver le max ?

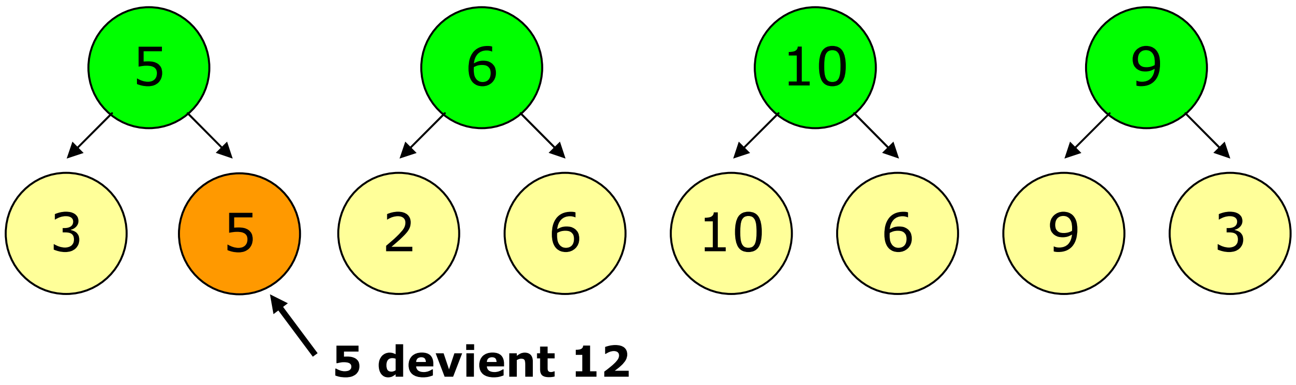
Solution : parcourir les maxima des paires.



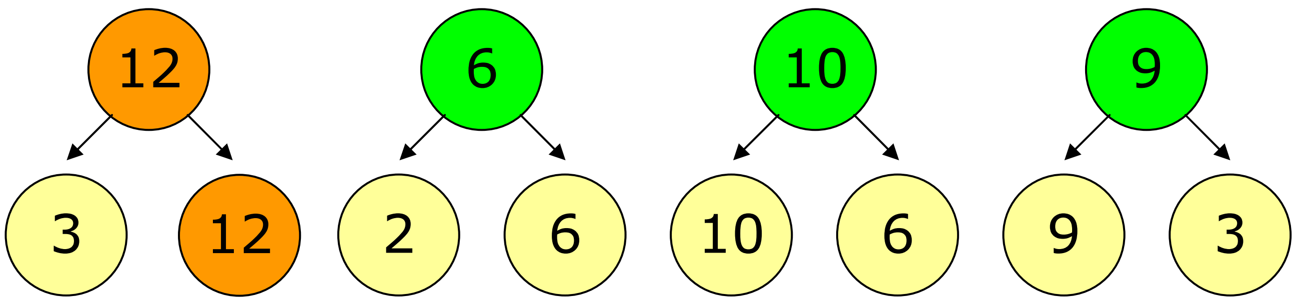
Coût : environ N/2 calculs suffisent.

Maintenir la structure

Question : une valeur change, que faire ?



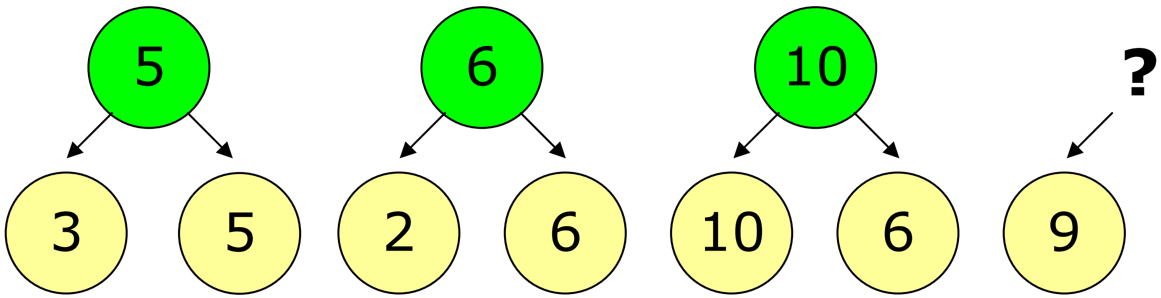
Solution : mettre à jour le maximum de la paire.



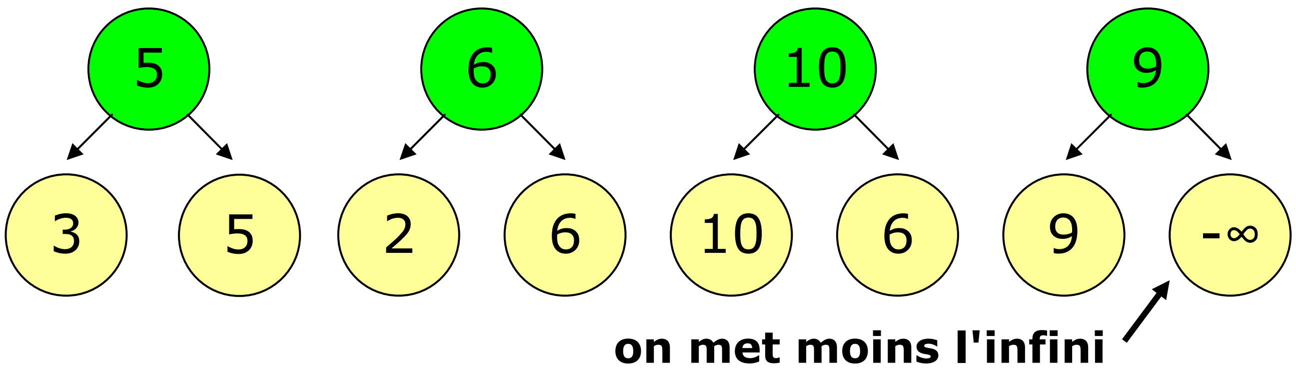
Coût : on doit effectuer une opération en plus.

Cas N impair

Question : si N est impair, comment peut-on faire ?



Solution : on ajoute une valeur pour rendre N pair.



Coût : coûte une unité en plus, c'est négligeable.

Récapitulation

Maintient du maximum de N valeurs qui changent

- on stocke N/2 valeurs supplémentaires : les maxima de chacune des paires.

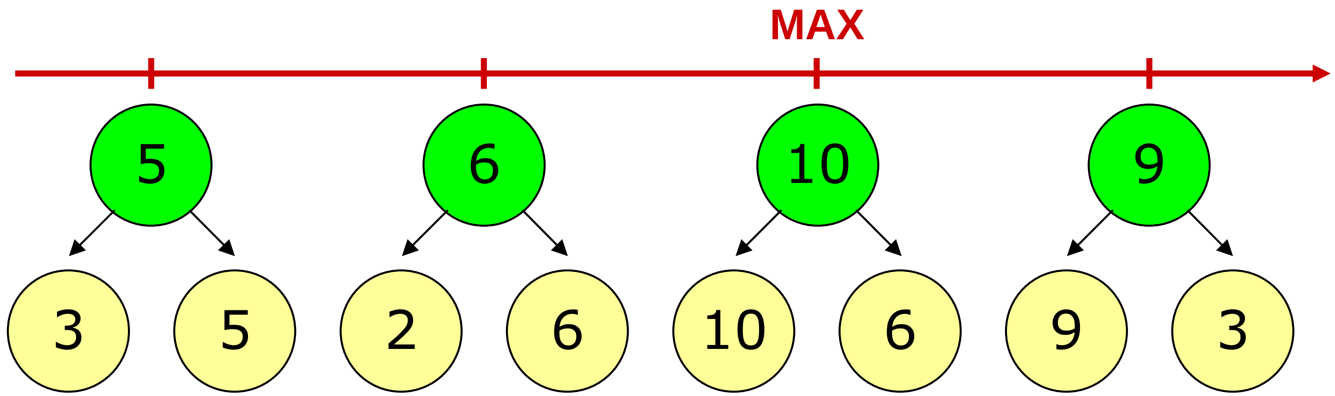
- maintenir les maxima des paires lorsqu'on change une valeur ne demande qu'une opération en plus.

- trouver le maximum se fait 2 fois plus vite.

Question : on peut donc maintenir le maximum deux fois plus vite; peut-on faire mieux ?

Trouver le maximum (1)

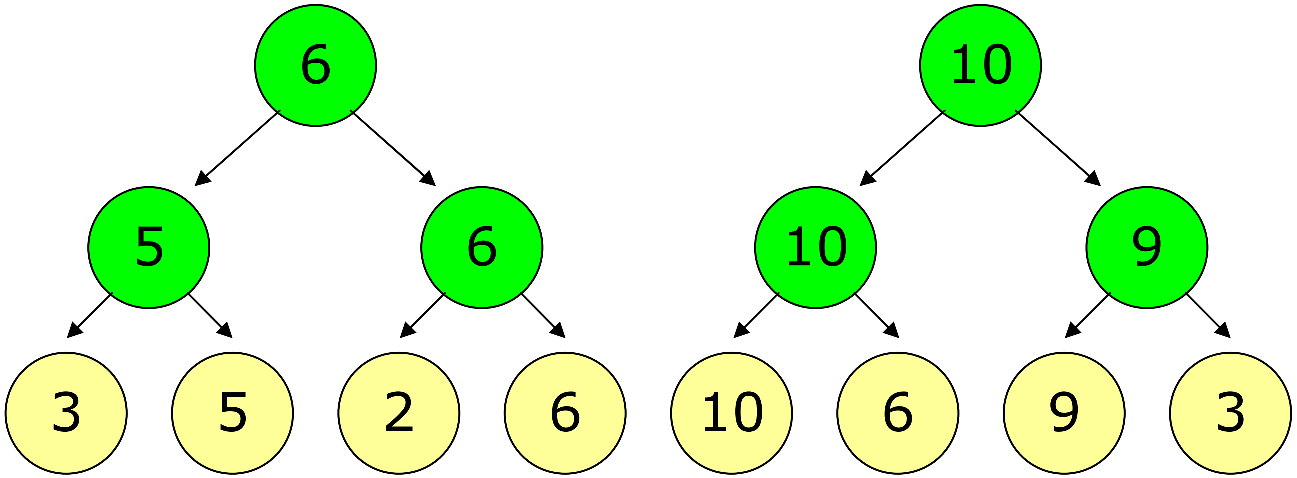
Question : comment faire pour trouver le max des N/2 maxima de paires plus rapidement ?



Indication : réfléchissez bien, vous savez déjà faire !

Trouver le maximum (2)

Solution : on n'a qu'à former des paires à nouveau.

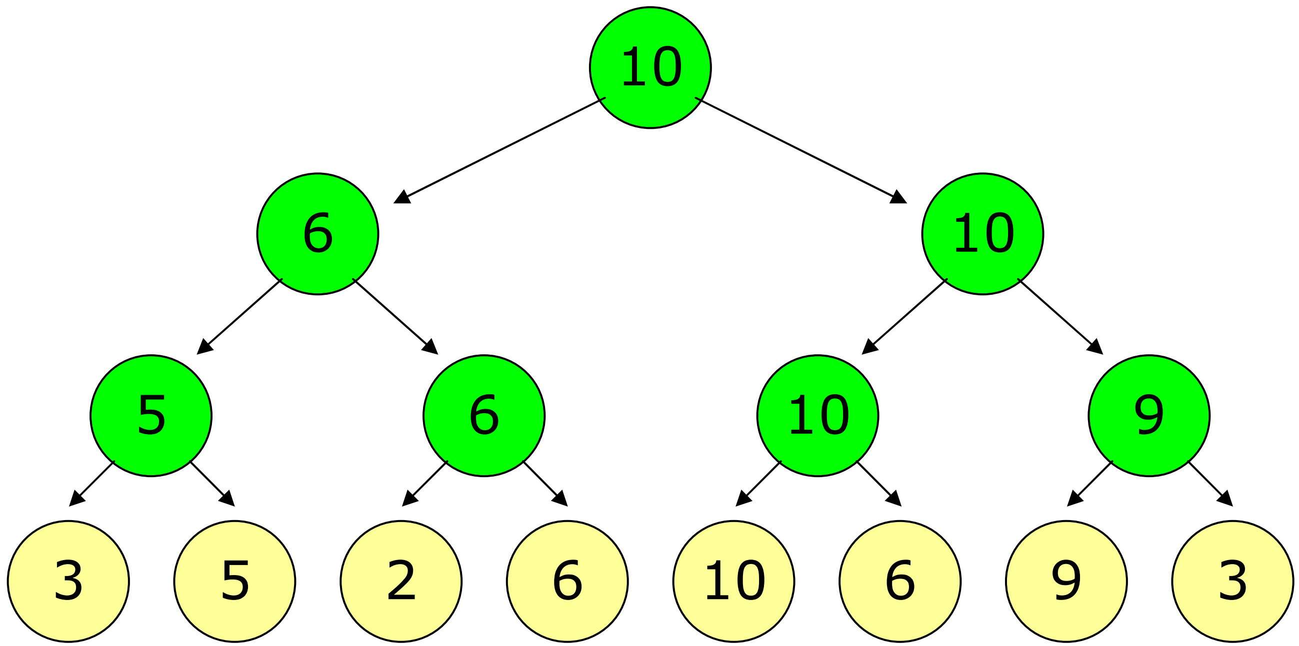


Remarque : les valeurs ajoutées correspondent à des maxima de groupe de 4 valeurs.

Question : et maintenant il nous reste à trouver le maximum de N/4 valeurs; comment fait-on ?

Trouver le maximum (3)

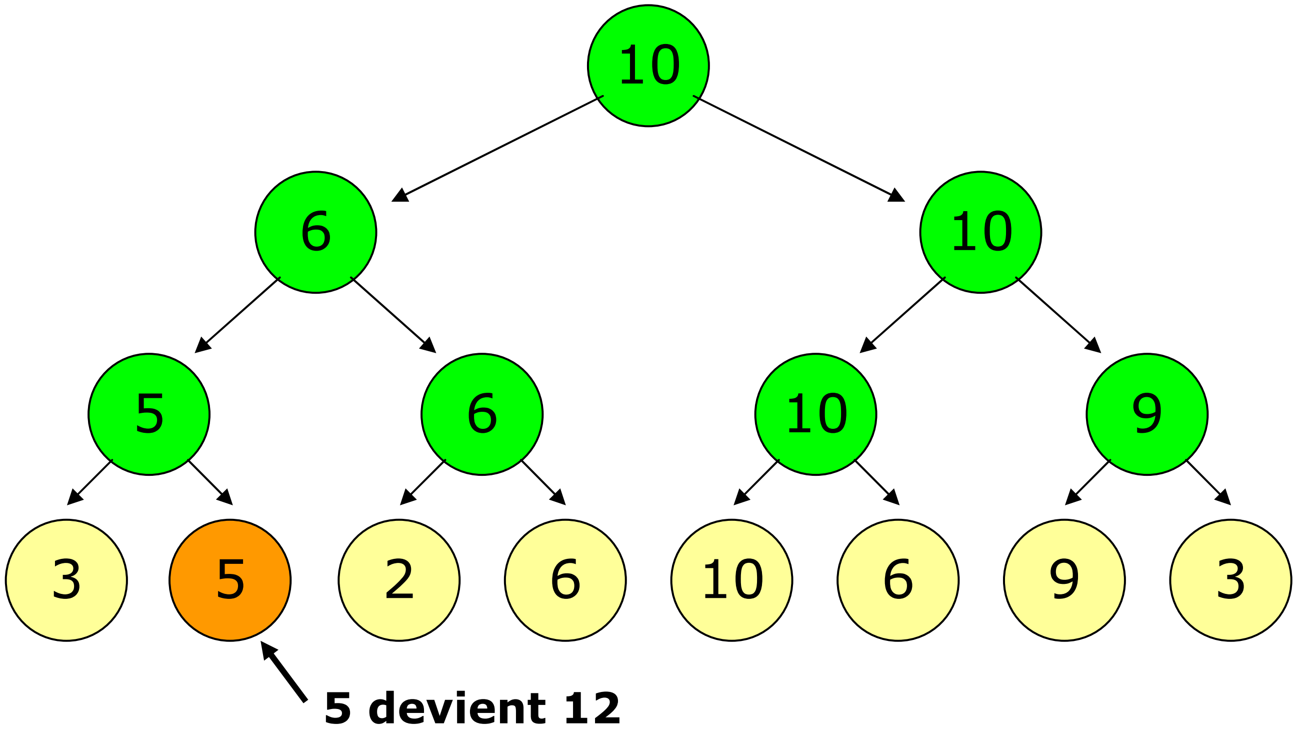
Solution : on n'a qu'à former des paires à nouveau.



Remarque : au bout d'un moment on n'a plus qu'une seule valeur égale au maximum, tout en haut de l'arbre. On l'appelle la racine de l'arbre.

Modifier une valeur (1)

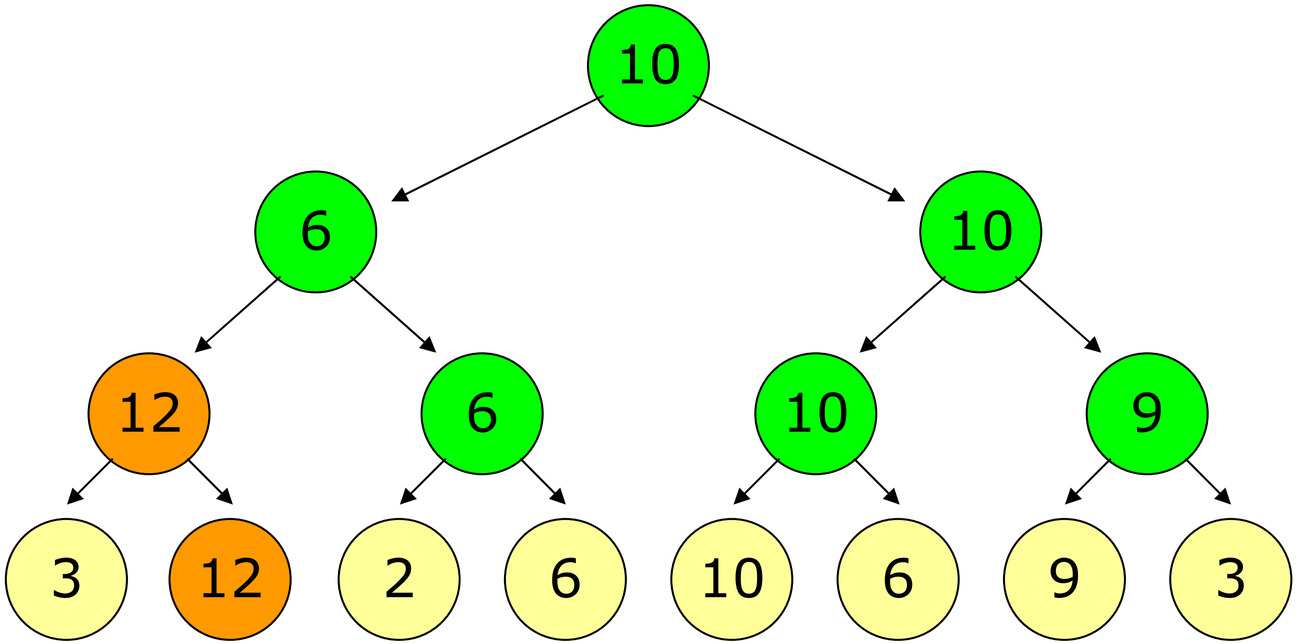
Question : une valeur change, que faire ?



Indication : il faut mettre à jour certains maxima.

Modifier une valeur (2)

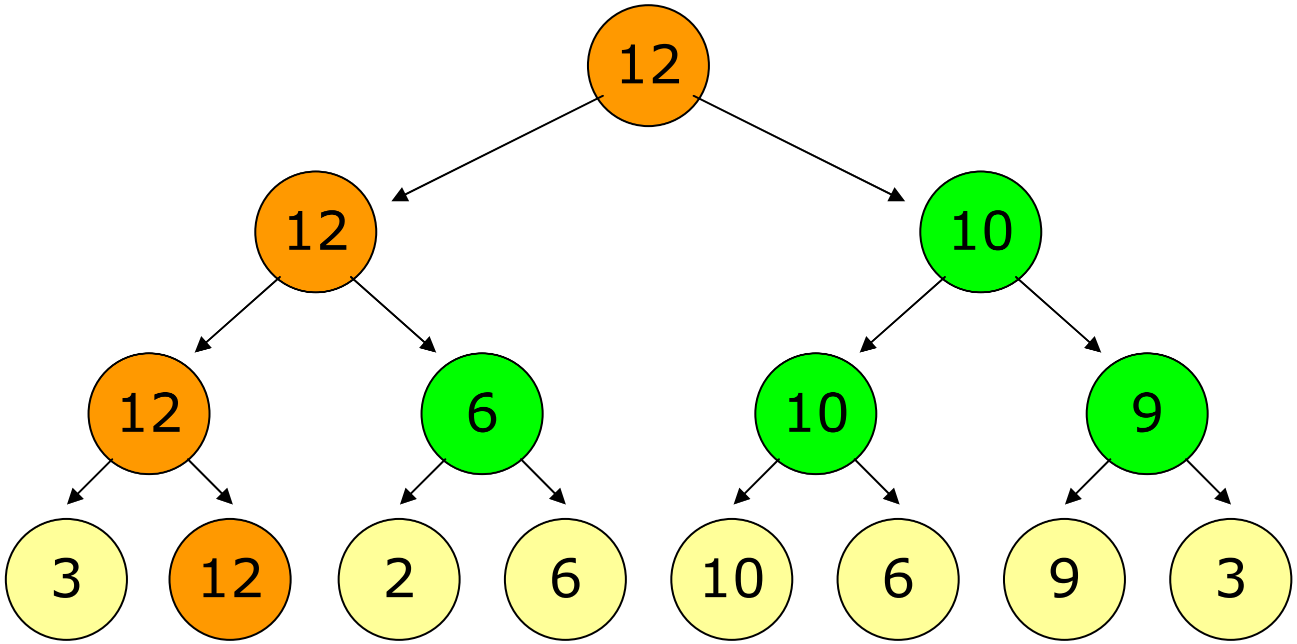
Solution : il faut déjà modifier la valeur au-dessus.



Solution : mais ce n'est pas tout, car maintenant le 6 n'est plus maximum de 6 et 12; il faut continuer à mettre à jour les valeurs jusqu'à la racine.

Modifier une valeur (3)

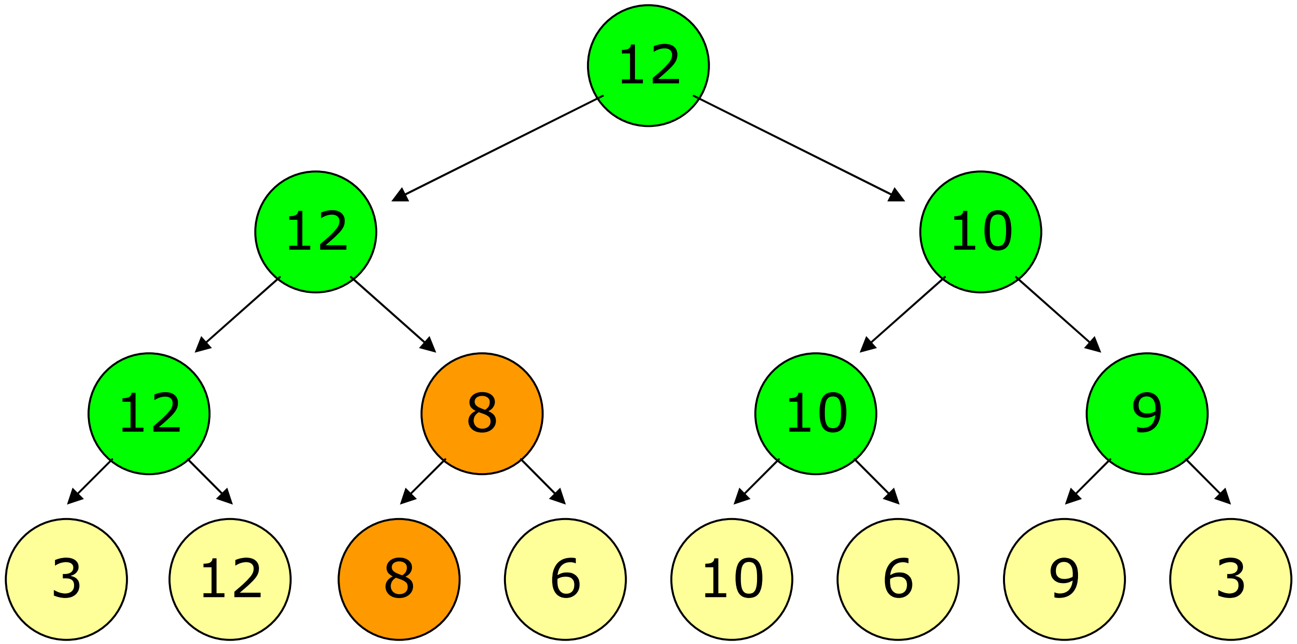
Résultat :



Question : essayez un autre exemple; par exemple, changez la valeur 2 en 8, et effectuez les mises à jour nécessaires.

Modifier une valeur (4)

Résultat :



Remarque : on modifie en tout deux valeurs; remarquez qu'on n'a pas eu besoin de propager des modifications jusqu'à la racine.

Questions auxquels il faut répondre

max : combien d'opérations pour trouver le max ?

modif : combien d'opérations pour maintenir la structure lorsqu'on change une valeur, au pire ?

déséquilibre : comment faire dans le cas où N n'est pas une puissance de 2 ?

initialisation : combien d'opérations sont nécessaires pour construire toute la structure au début ?

Essayez de répondre à toutes ces questions.

Réponses et nouvelles questions

max : il suffit de lire la valeur de la racine.

modif : au pire, il faut mettre à jour autant de valeurs qu'il y a d'étages dans la structure. Quelle est cette hauteur en fonction de N ?

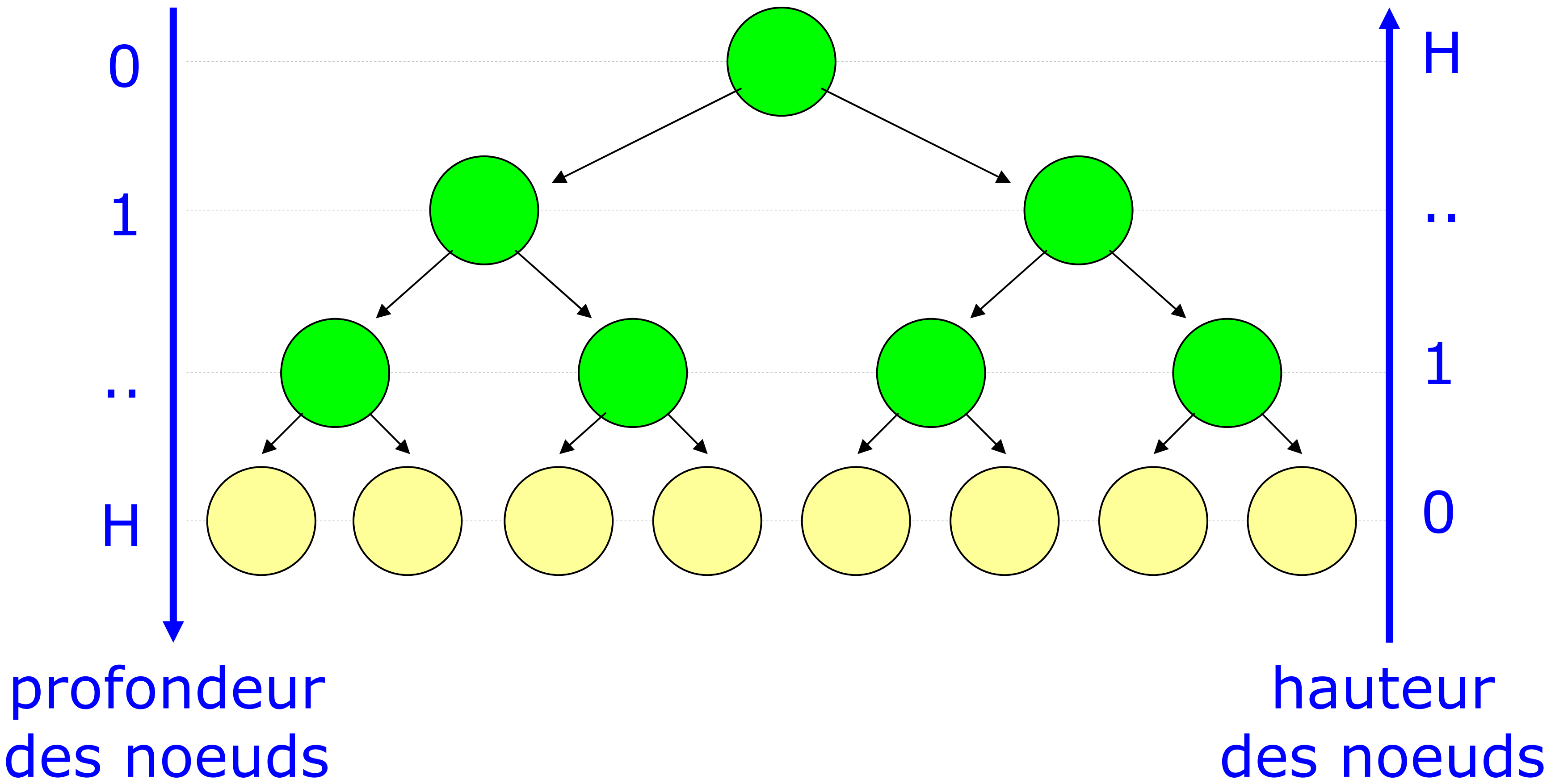
déséquilibre : si N n'est pas une puissance de 2, on se ramène à la première puissance de 2 juste au-dessus de N. Au pire, combien ajoute-t-on de valeurs en fonction de N ? Faites un dessin.

initialisation : pour construire tout l'arbre, il faut traiter chaque valeur de l'arbre. Combien y a-t-il de valeurs dans l'arbre en tout ?

Essayez de répondre à toutes les questions.

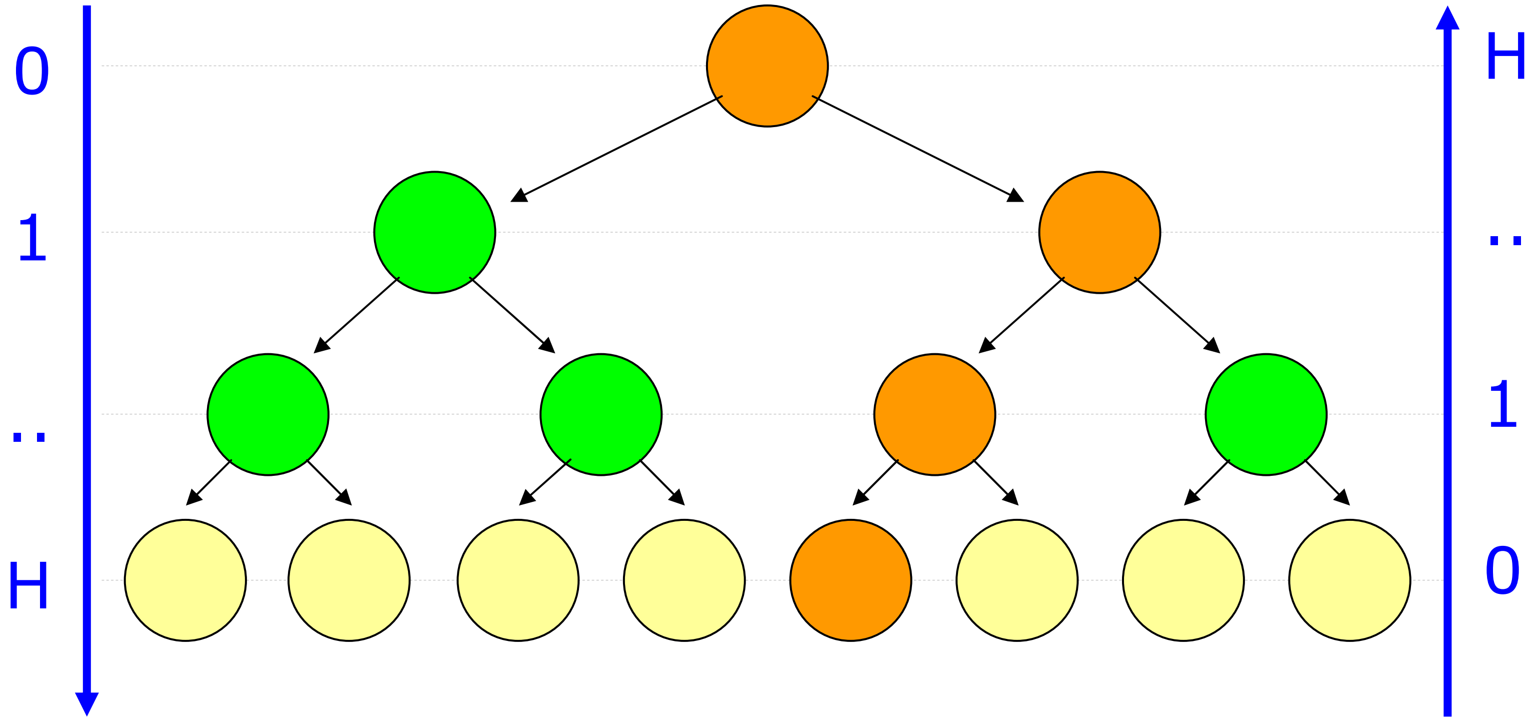
Hauteur de l'arbre

Idée : Commençons par le cas où $N = 2^H$.



Travail pour une modification

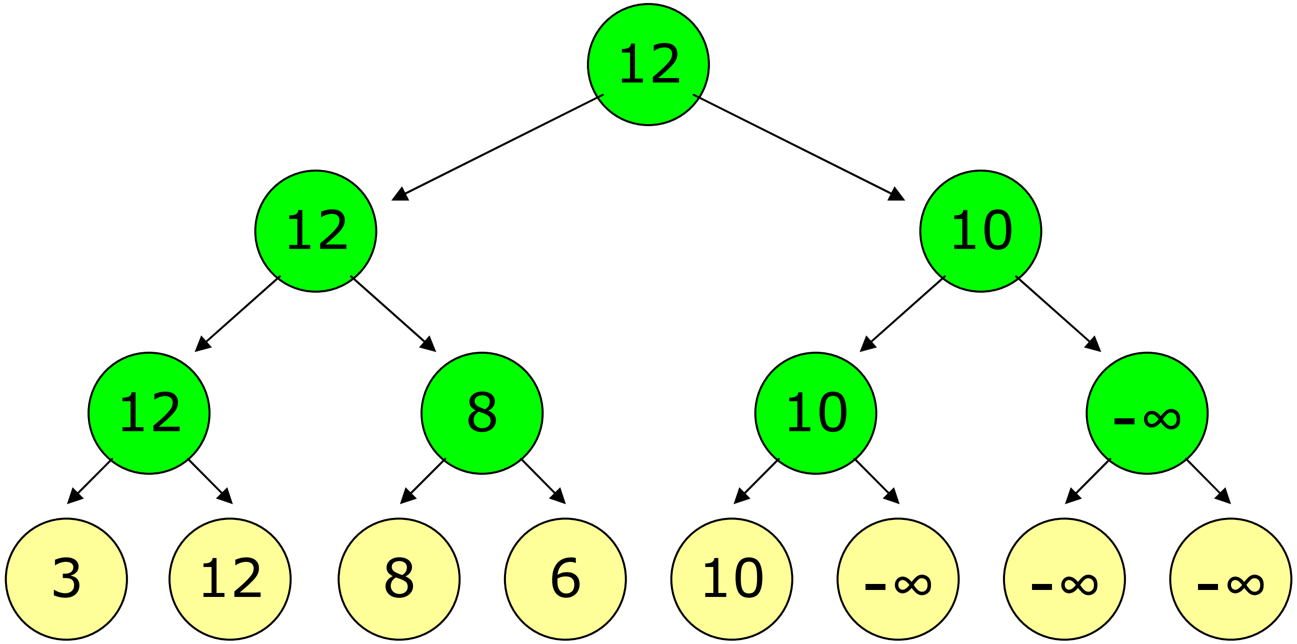
Solution : Si $N = 2^H$, un chemin d'une valeur en bas jusqu'à la racine tout en haut a H+1 noeuds.



Remarque : en orange, ensemble des valeurs modifiées lors d'un changement, au pire.

Cas où N n'est pas une puissance de 2

Idée : si N n'est pas une puissance de 2, on complète l'arbre; ici N = 5, on ajoute 3 valeurs.



Remarque : combien ajoute-t-on de valeurs dans le pire cas, en fonction de N ?

Un peu de maths (1)

Objectif : On veut exprimer le coût du changement d'une valeur, en fonction de N. Or on a vu que ce coût était au pire de l'ordre de grandeur de la hauteur de l'arbre H. On souhaite donc exprimer H en fonction de N.

Cas particulier : si N est une puissance de 2 :

$N = 2^H$ et donc $H = \log_2(N)$

(c'est la définition du logarithme en base 2)

Cas général : si N n'est pas une puissance de 2, on ajoute des valeurs jusqu'à en avoir $N' = 2^H$.

Montrons que $H \approx \log_2(N)$

Un peu de maths (2)

Preuve intuitive :

Comme N' = 2H, on a H = log2(N').

Maintenant N' n'est pas loin de N, et on peut dire que $\log_2(N')$ n'est pas loin de $\log_2(N)$.

Donc H est environ égal à $\log_2(N)$.

Preuve rigoureuse :

$2^{H-1} < N \leq N' = 2^H$ on passe tout au $\log_2$ et :

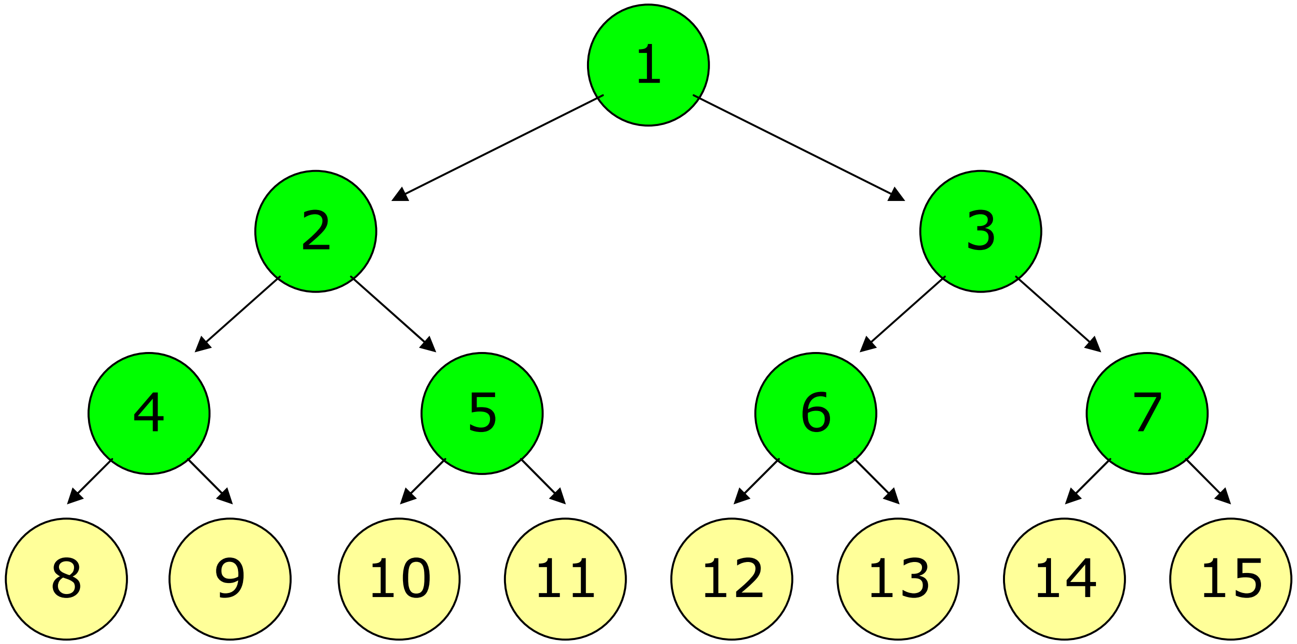
$H-1 < \log_2(N) \leq \log_2(N') = H$

D'où $H = \lceil \log 2(N) \rceil$

H est la partie entière supérieure de $\log_2(N)$

Nombre de noeuds dans l'arbre (1)

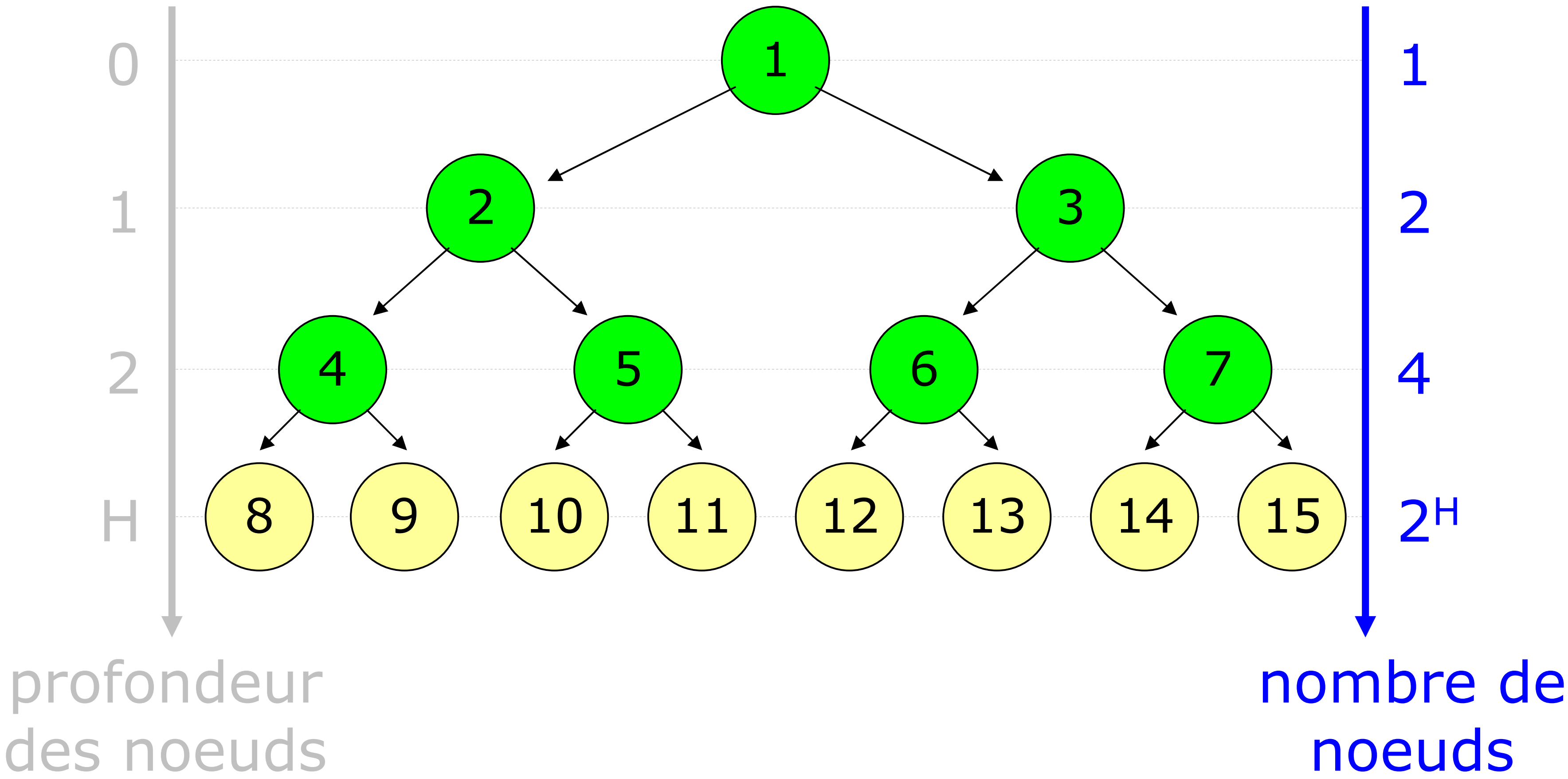
Idée : pour compter les noeuds, on les numérote.



Question : combien y a-t-il de noeuds dans l'arbre en fonction de N, lorsque N est une puissance de 2 ?

Nombre de noeuds dans l'arbre (2)

Idée : comptons les noeuds à chaque profondeur.



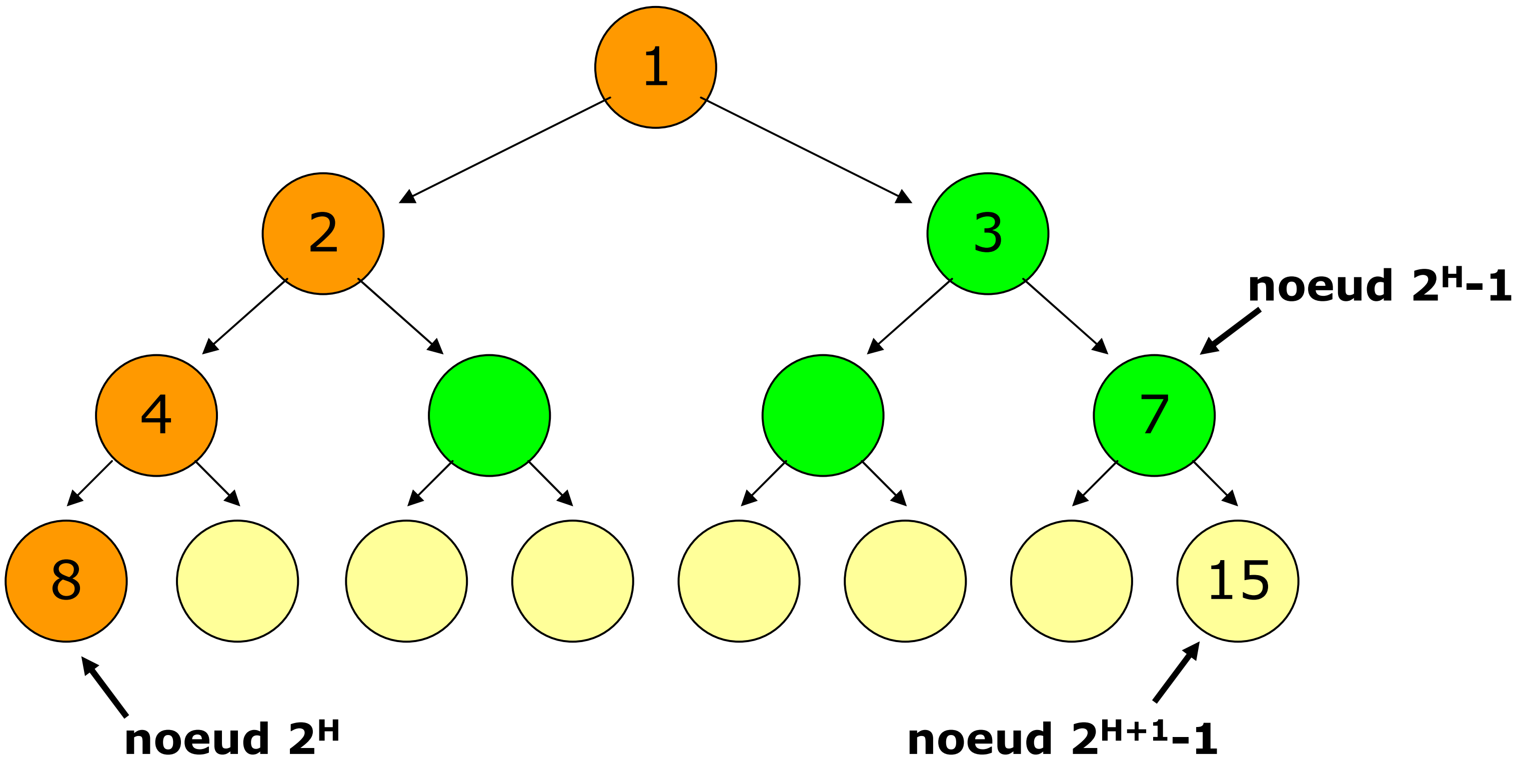
Résultat : il y a 2^p noeuds à la profondeur p.

Nombre de noeuds dans l'arbre (3)

Résultat : nombre de noeuds = $2^{H+1}-1$ = $2 \cdot N - 1$

Preuve 1 : $2^0 + 2^1 + .. + 2^{H-1} + 2^H = 2^{H+1}-1$ (maths)

Preuve 2 : en orange, les numéros puissance de 2

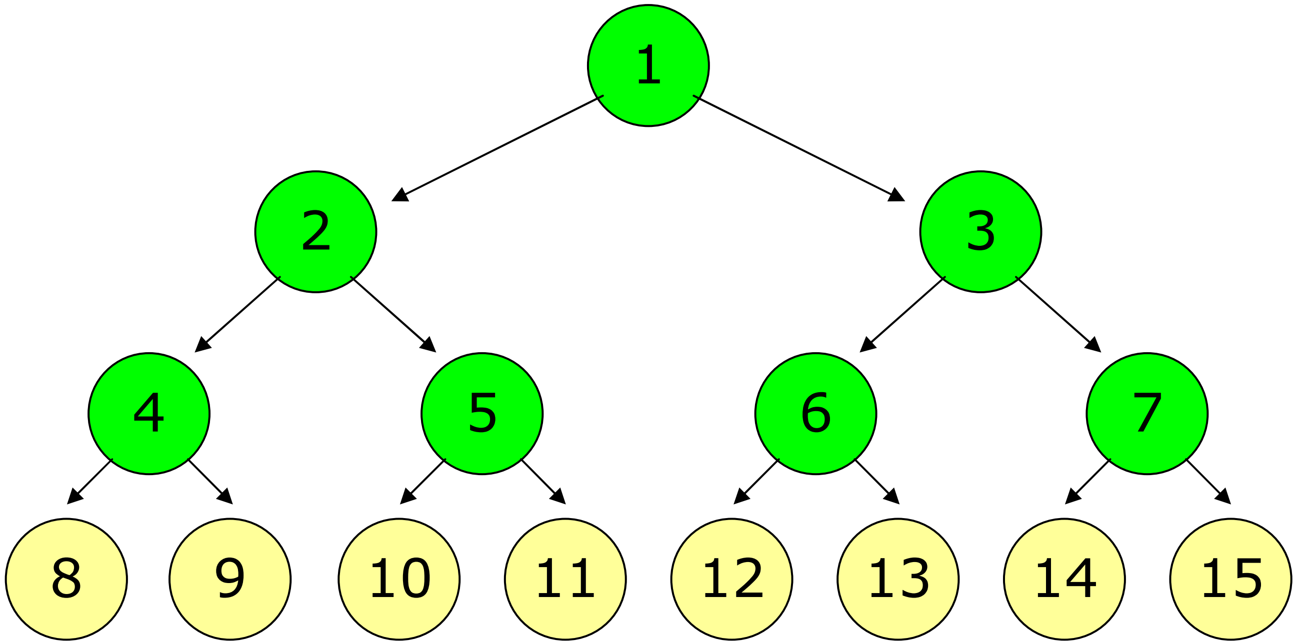


Astuce pour coder (1)

Question : quelles relations y a-t-il entre le numéro d'un noeud et les numéros :

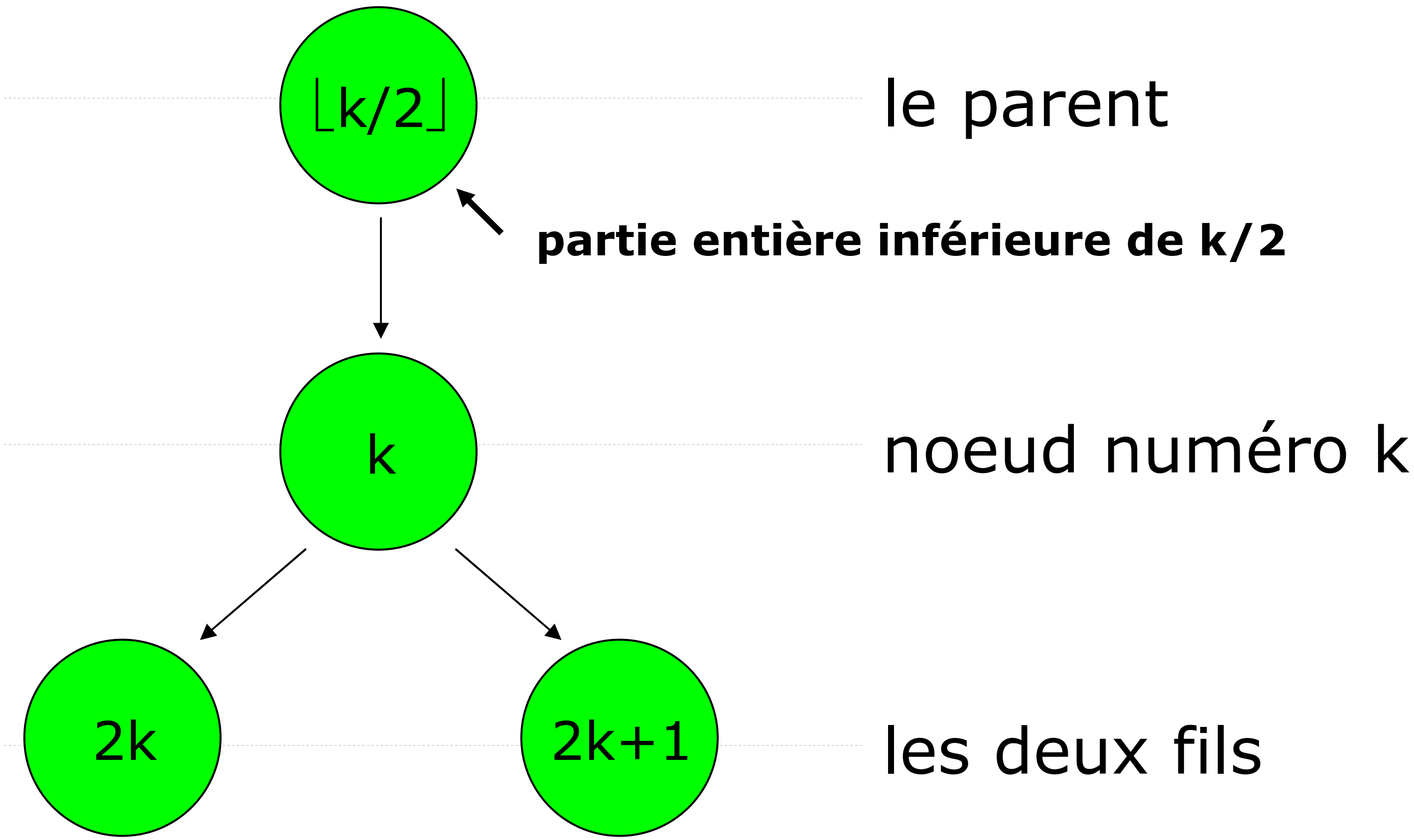
- du noeud juste au-dessus de lui (son parent)

- des noeuds juste en-dessous de lui (ses fils)



Astuce pour coder (2)

Réponse :



Récapitulation

Maintient du maximum de N valeurs qui changent

- on ajoute suffisement de valeurs pour que N devienne puissance de 2.

- on stocke N-1 valeurs supplémentaires; ces noeuds contiennent le maximum de leurs 2 fils.

- le maximum de toutes les valeurs est la valeur contenu dans le noeud à la racine de l'arbre.

- maintenir la structure lorsqu'une valeur change requiert un nombre d'opérations en $O(\log_2(N))$.

La structure obtenue s'appelle un arbre binaire.