



# TD 3

typedef struct {

void \*\* array;

int n, mmax;

int (\*p) (const void \*, const void \*);

} heap;

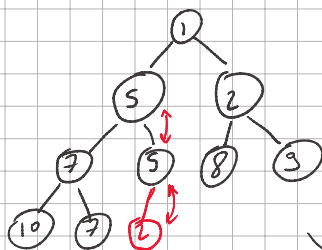
## Question 1

Un tas binaire est un arbre quasi-parfait (ou parfait).

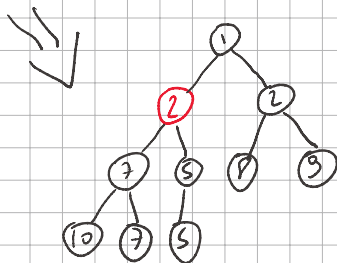
De plus, le minimum est à la racine.  
Les fils doivent être supérieurs à leurs pères.



## Question 2



insertion de l'élément 2



Tas binaire de hauteur  $h$  a  
au moins  $1+2+4+\dots+2^{h-1}+1$  nœuds  
 $= 2^h$

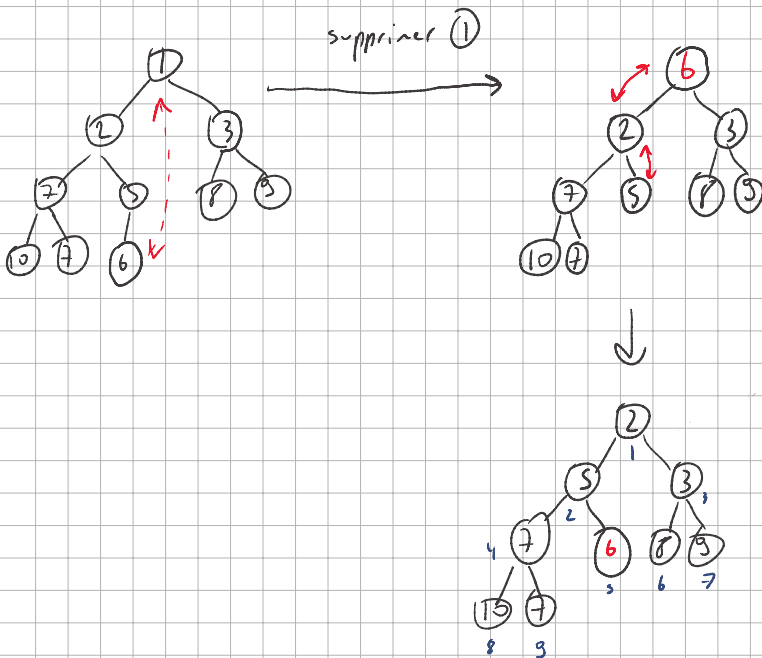
$n = nb \text{ nœuds}$

donc  $n \geq 2^h$

$$\log_2(n) \geq \log_2(2^h) = h$$

$$h \leq \log_2(n)$$

### Question 3



Implementation du tableau :

0	1	2	3	4	5	6	7	8	9	10	...
1	2	3	4	5	6	7	8	9	10	11	...

### Question 4

$$\frac{1-2^{n+1}}{1-2} = 2^{n+1}-1 \Rightarrow$$

accès valeur

Implementation du tableau :

0	1	2	3	4	5	6	7	8	9	10	...
///	2	5	3	7	6	8	9	10	7	///	...

$$2^h = 2^0 = 1$$

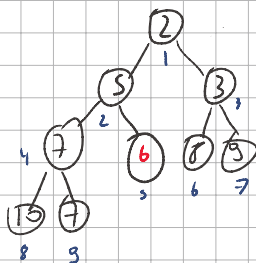
$$2^h = 2^1 = 2$$

$$2^{h+1} = 2^2 = 4$$

$$2^{h+1} = 2^3 = 8$$

$$2^{h+1} = 2^4 = 16$$

fil gauche	fil droit
$2^h$	$2^{h+1}$



à partir de 6 remonter à 2

$$6 = \text{index } 5 = 2^2 + 1$$

$$5 = \text{index } 2 = 2^1$$

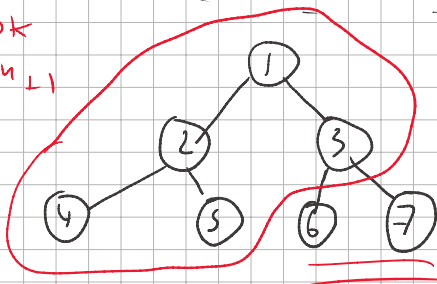
$$2 = \text{index } 1 = 2^0$$

à partir de 8 remonter à 2

$$8 = \text{index } 6 = 2^3$$

PROBLEME

OK  
 $2^h / 2^{h+1} = 1$



$$h = 0$$

$$h = 1$$

$$h = 2$$

$$1 = 2^0 = 2^0 = 1$$

$$2 = 2^1 = 2^1 = 2$$

$$3 = 2^1 + 1 = 2^1 + 1 = 3$$

$$4 = 2^2 = 2^2 = 4$$

$$5 = 2^2 + 1 = 2^2 + 1 = 5$$

$$6 = 2^2 + 1 - 1 = 8 - 1 - 1 = 6$$

$$7 = 2^2 + 1 - 1 = 8 - 1 - 1 = 7$$

///	1	2	3	4	5	6	7	///	///	///	///	///	///	///	///
-----	---	---	---	---	---	---	---	-----	-----	-----	-----	-----	-----	-----	-----



$$h=2 \quad h=1$$

$$4 \rightarrow 2$$

$$h=2 \quad h=1$$

$$5 \rightarrow 2$$

$$h \perp 2$$

$$h \perp 1$$

$$h=1 \quad h=2$$

$$8 \rightarrow 4$$

