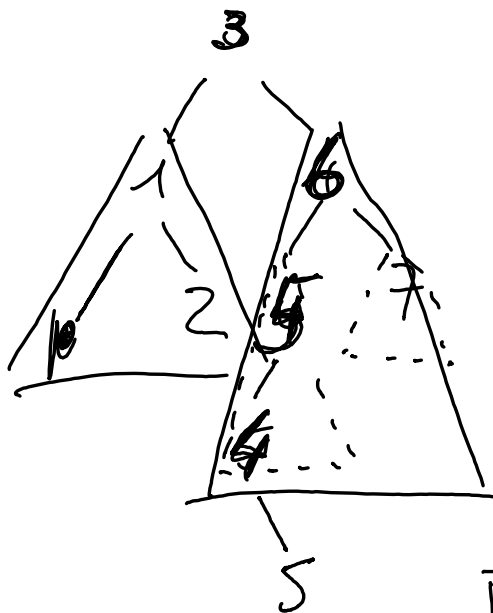
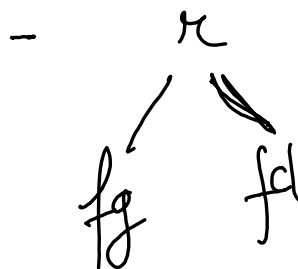


ch3 Arbres

1. Arbres binaires de recherche.



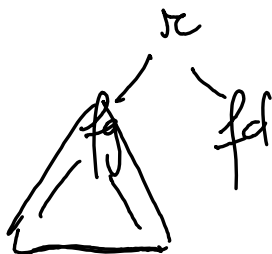
- binaire:
noeud \rightarrow au +
2 fils



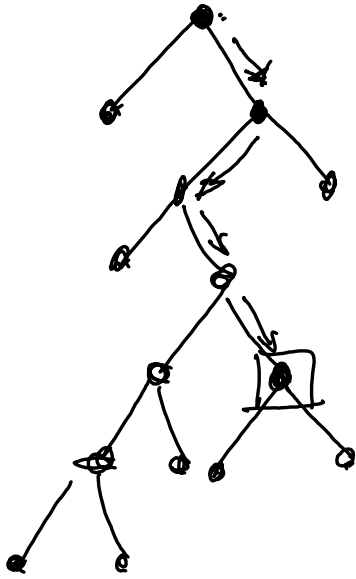
$$fg \leq r < fd$$

(5)

Structure de données

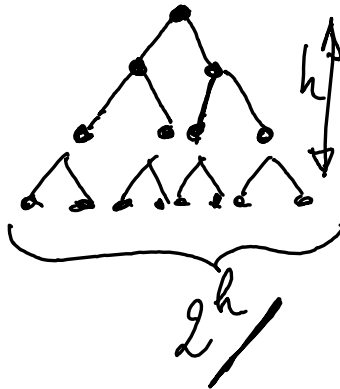
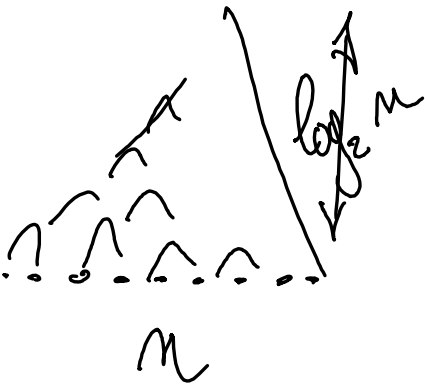


"dichotomique"

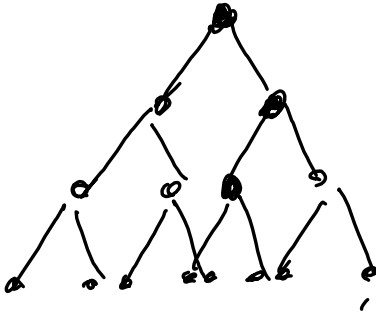


$$\log_2 n$$

n éléments



$$\begin{aligned} 1 &= 2^0 \\ 2 &= 2^1 \\ 4 &= 2^2 \\ 8 &= 2^3 \end{aligned}$$



$h=3$ $\rightarrow 2^h = 8$ leaves.

$\rightarrow 2^{(h+1)} - 1 = 15$ nodes.

$$2^3 + 2^2 + 2^1 + 2^0 = 2^4 - 1$$

$$2^n + 2^{n-1} + 2^{n-2} + \dots + 2^0 = 2^{(n+1)} - 1$$

```
def recherche (element, arbre) :  
    if element == racine (arbre) :  
        return True  
    if element < racine (arbre) :  
        return recherche (elr, fg (arbre))  
    else :  
        return recherche (elr, fd (arbre))
```

arbre \rightarrow racine()
fg()
fd()

convention : arbre = tuple (r, fg, fd)