Introduction à l'informatique CM8

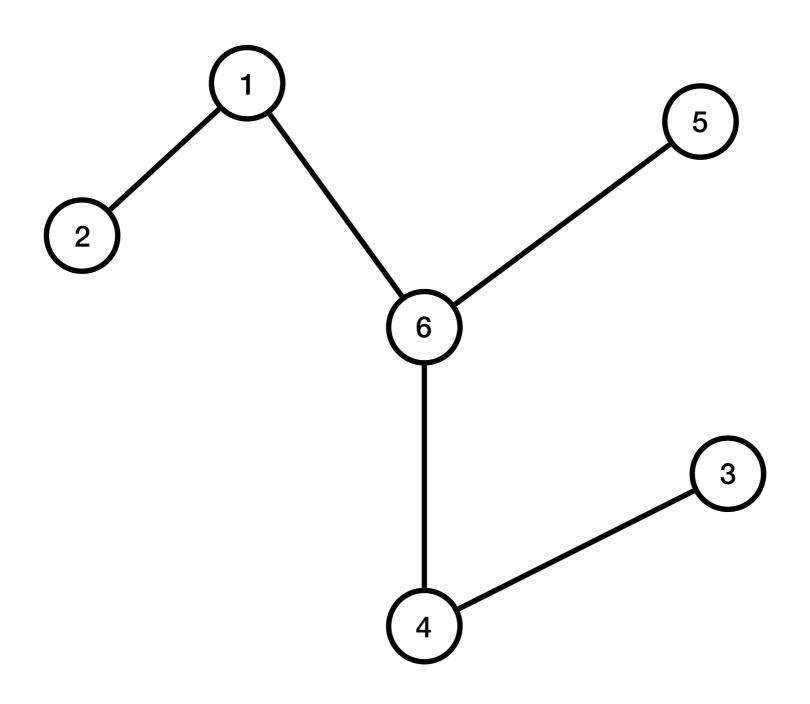
Antonio E. Porreca https://aeporreca.org/teaching

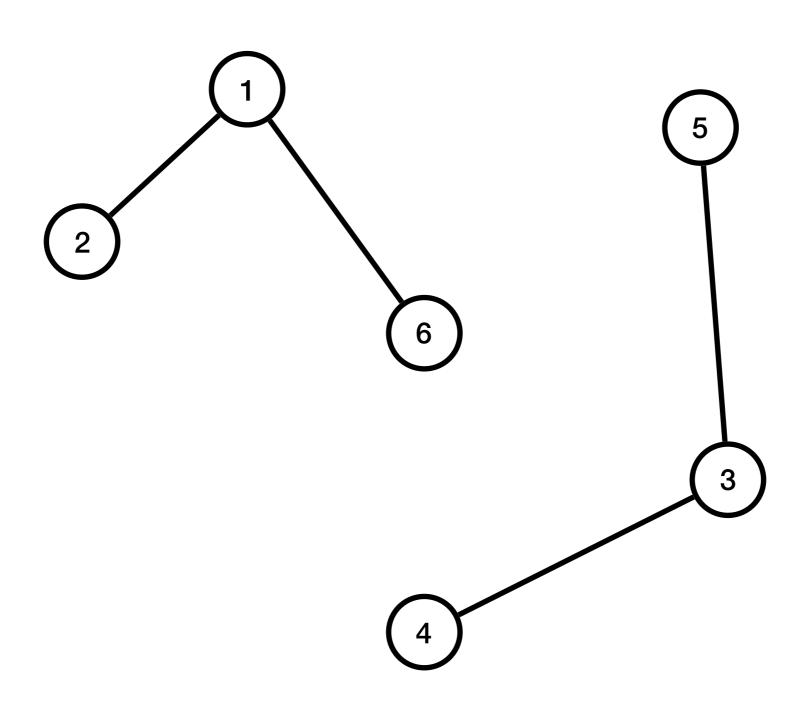
Arbres

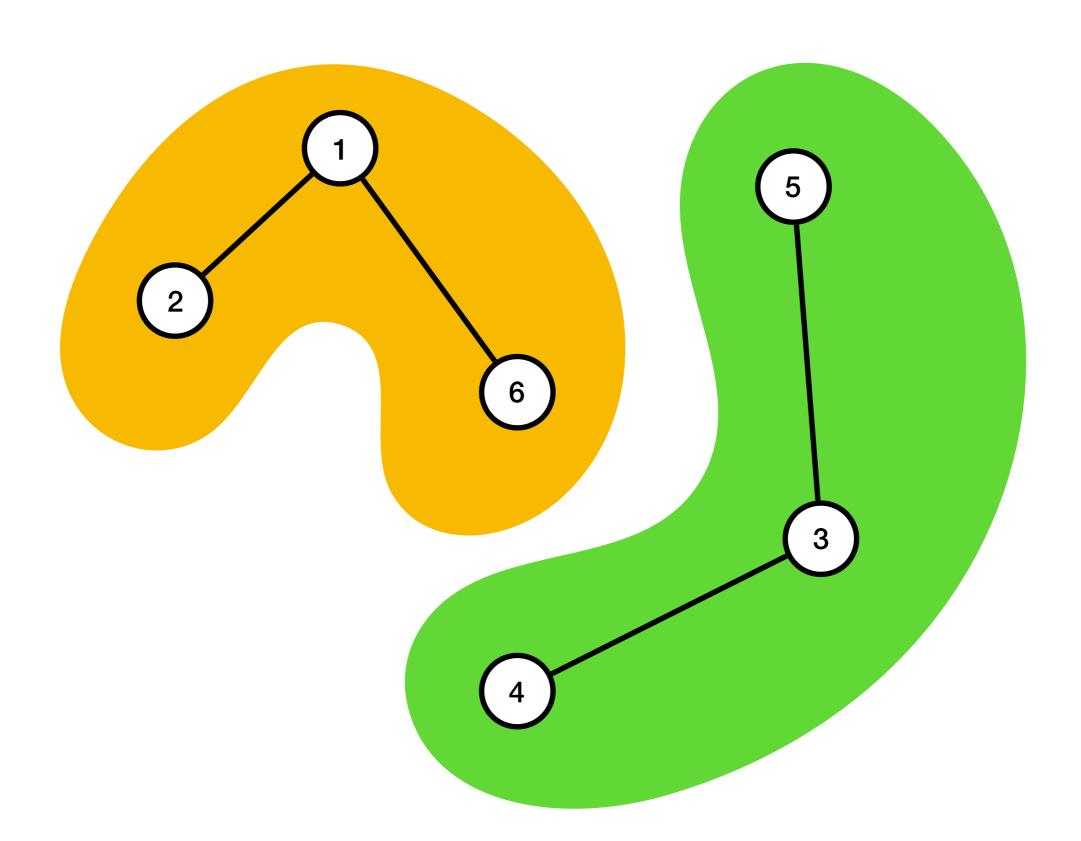


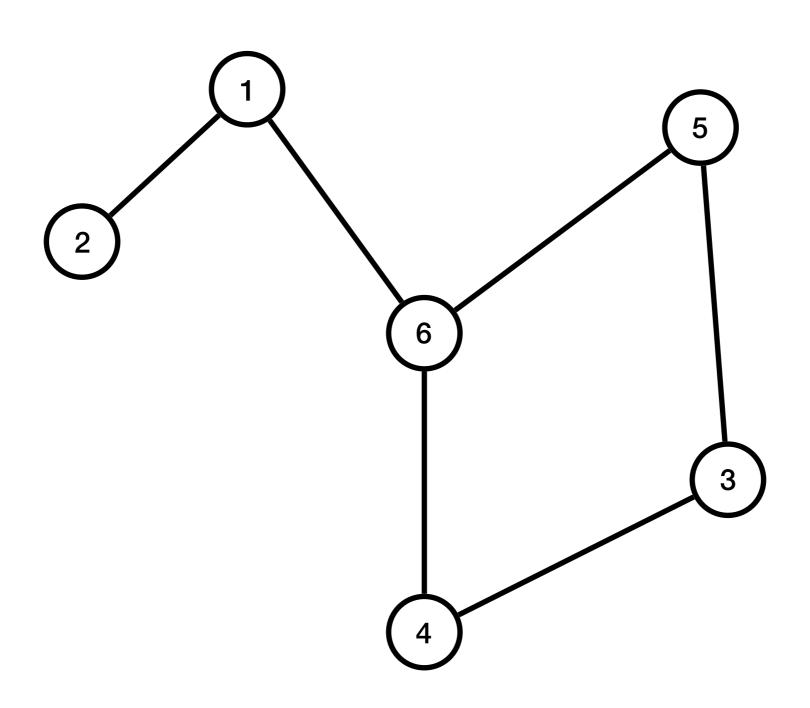
Un arbre est un graphe non orienté, connexe et acyclique

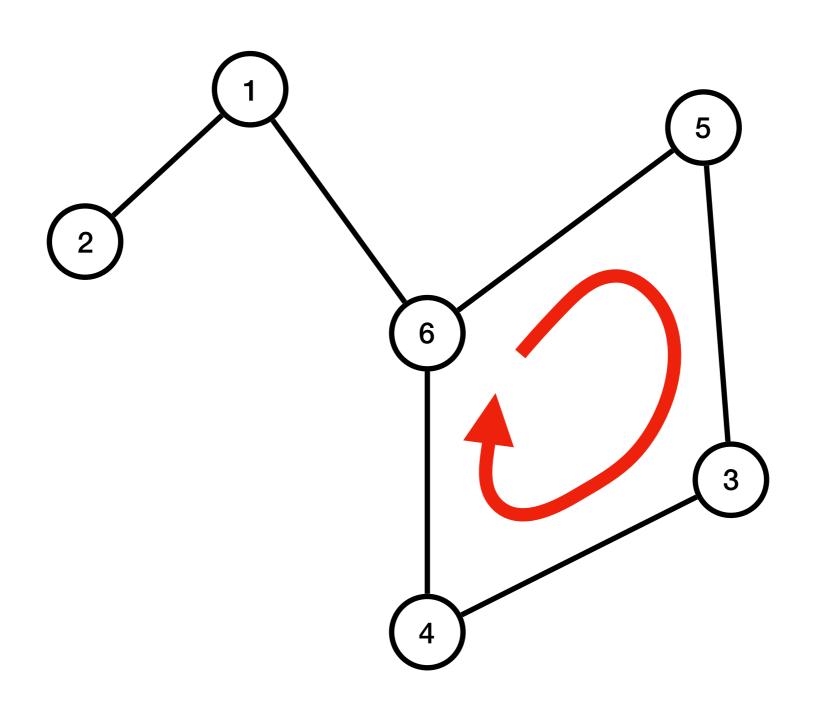
Un arbre

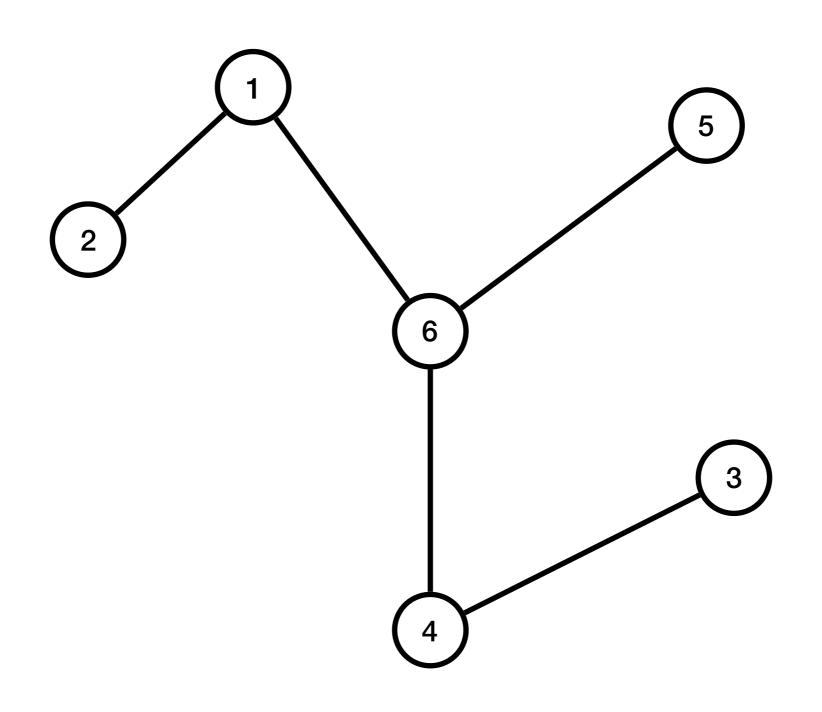


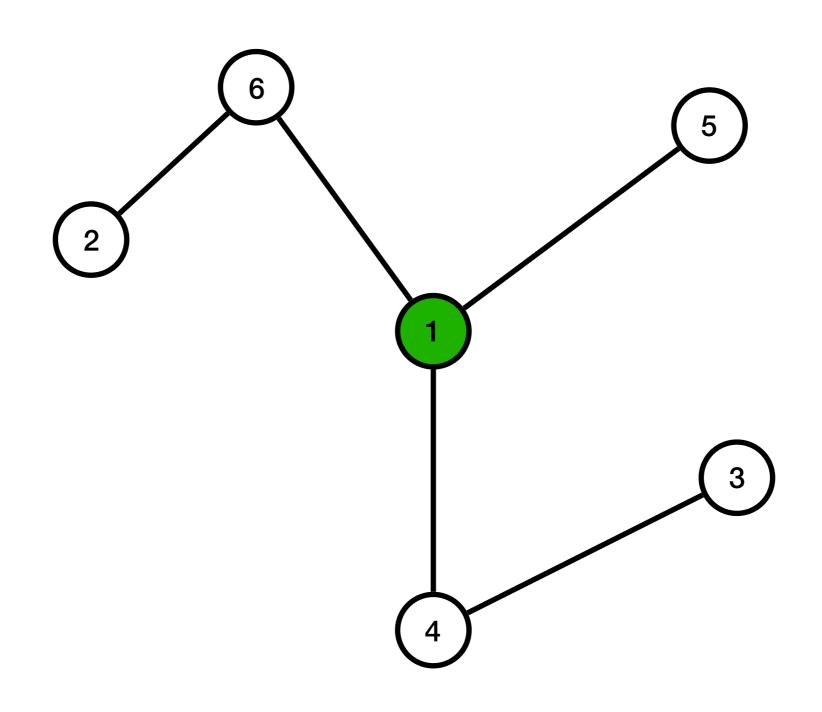


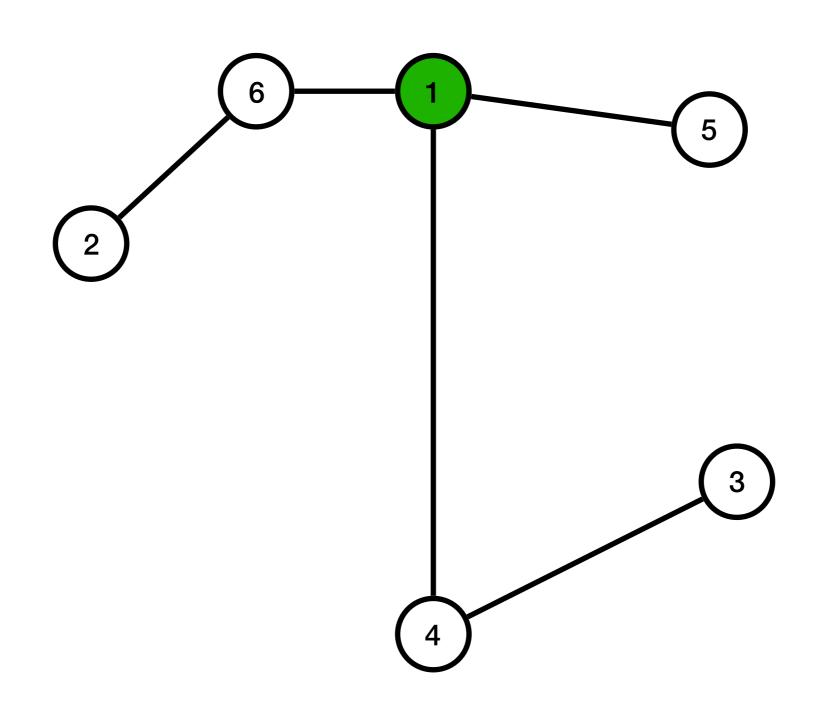


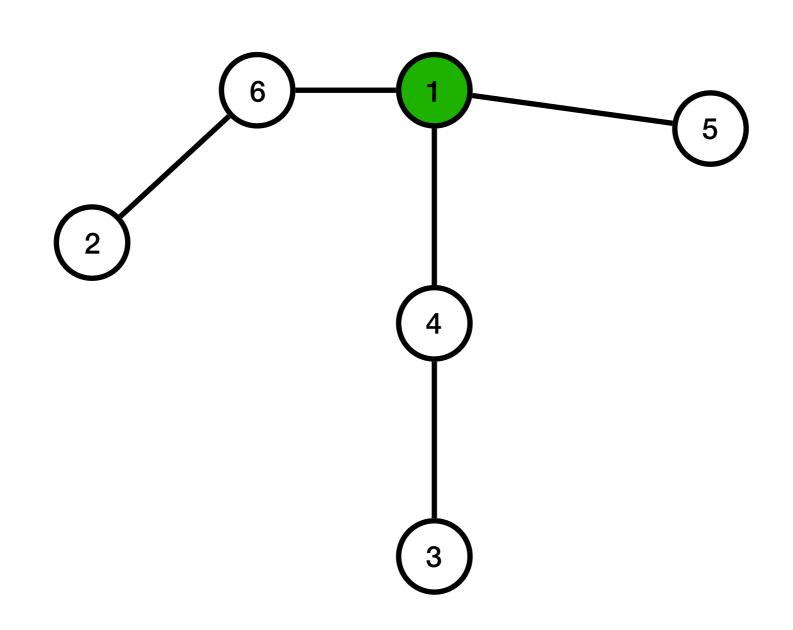


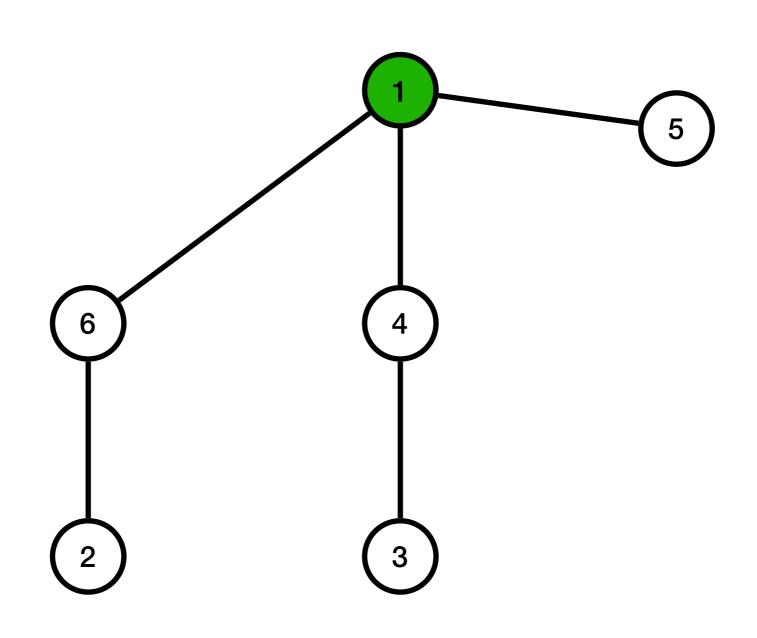


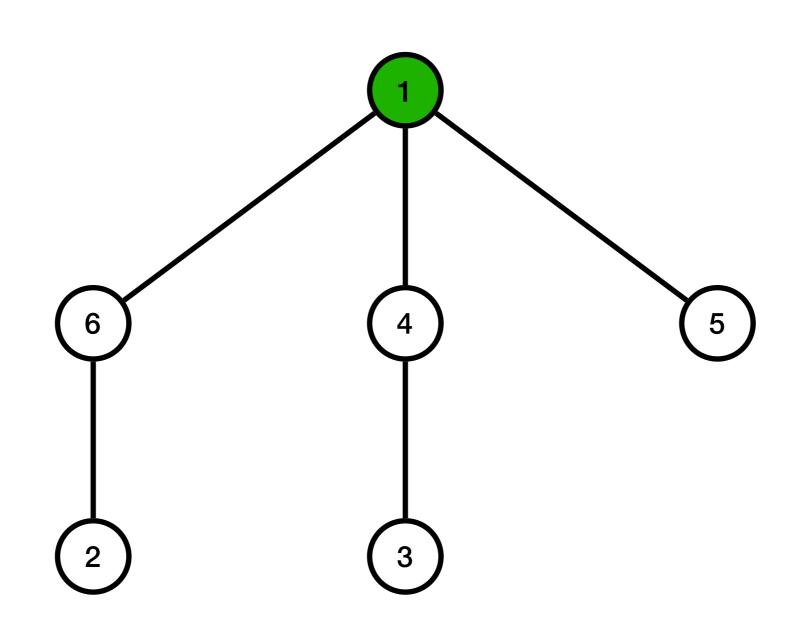




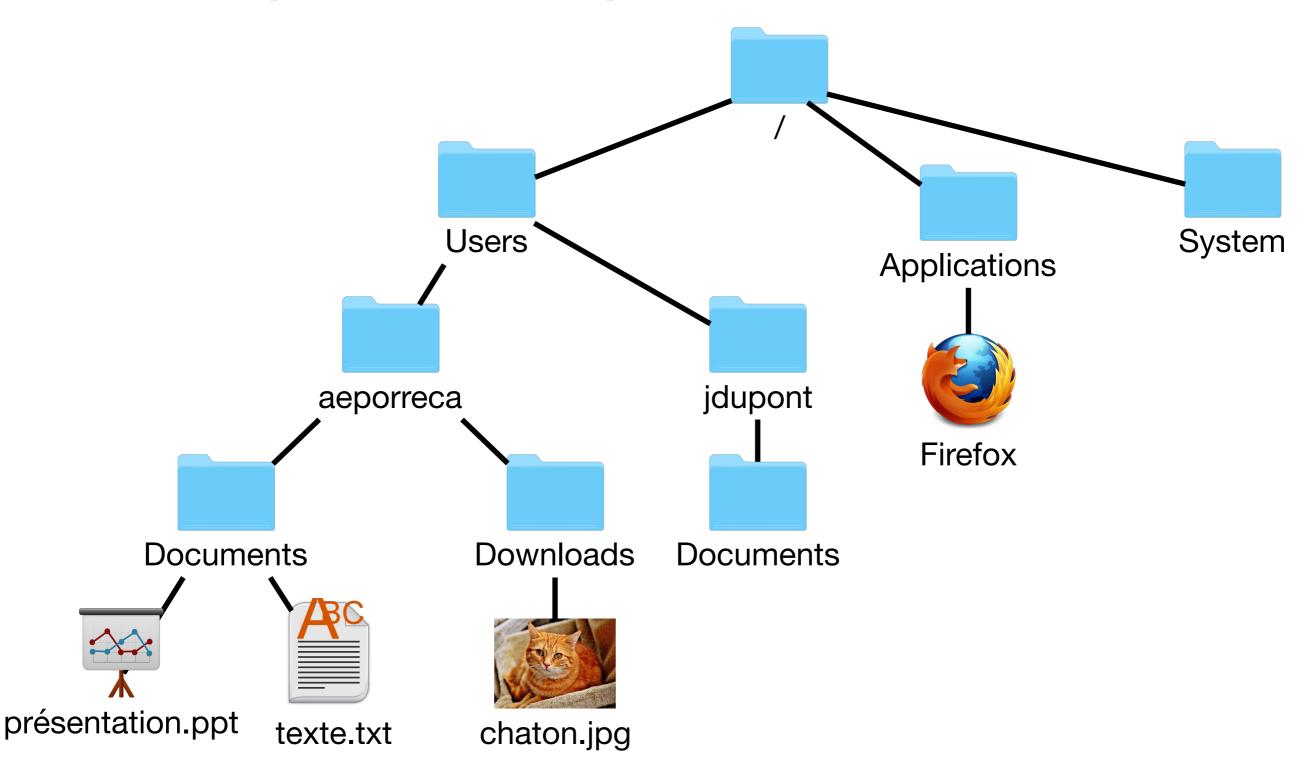








Arborescence des répertoires (dossiers) et fichiers



lettre	codage fixe
a	00000
b	00001
c	00010
d	00011
e	00100
f	00101
g	00110
h	00111
i	01000
j	01001
k	01010
l	01011
m	01100
n	01101
О	01110
p	01111
q	10000
r	10001
s	10010
t	10011
u	10100
v	10101
w	10110
x	10111
y	11000
z	11001
espace	11010

lettre	codage variable
a	1010
b	0010011
c	01001
d	01110
e	110
f	0111100
g	0111110
h	0010010
i	1000
j	011111110
k	011111111001
l	0001
m	00101
n	1001
0	0000
p	01000
q	0111101
r	0101
S	1011
t	0110
u	0011
v	001000
w	0111111111000
x	01111110
y	0111111111
z	01111111101
espace	111

encore

lettre	codage fixe
a	00000
b	00001
c	00010
d	00011
e	00100
f	00101
g	00110
h	00111
i	01000
j	01001
k	01010
l	01011
m	01100
n	01101
О	01110
p	01111
q	10000
r	10001
S	10010
$oxed{t}$	10011
u	10100
v	10101
$oxed{w}$	10110
x	10111
y	11000
z	11001
espace	11010

lettre	codage variable
a	1010
b	0010011
c	01001
d	01110
\overline{e}	110
f	0111100
g	0111110
h	0010010
i	1000
j	011111110
\overline{k}	011111111001
l	0001
\overline{m}	00101
\overline{n}	1001
0	0000
\overline{p}	01000
\overline{q}	0111101
r	0101
s	1011
t	0110
\overline{u}	0011
\overline{v}	001000
\overline{w}	011111111000
\overline{x}	01111110
\overline{y}	0111111111
\overline{z}	01111111101
espace	111
	<u> </u>

en	C	O	r	e
			_	

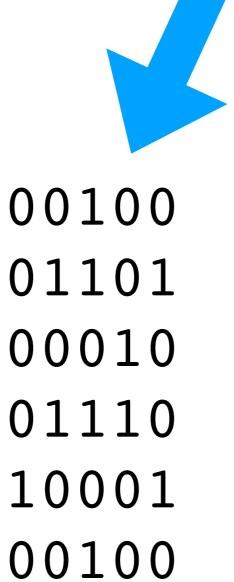


lettre	codage variable
$\frac{a}{a}$	1010
$\frac{a}{b}$	0010011
$\frac{\sigma}{c}$	01001
$\frac{c}{d}$	01110
$\frac{a}{e}$	110
	0111100
$\frac{f}{g}$	
$\frac{g}{l}$	0111110
h	0010010
$\frac{i}{\cdot}$	1000
j	011111110
k	011111111001
l	0001
m	00101
n	1001
0	0000
p	01000
q	0111101
r	0101
s	1011
t	0110
u	0011
\overline{v}	001000
\overline{w}	011111111000
\overline{x}	01111110
\overline{y}	0111111111
$\frac{z}{z}$	01111111101
espace	111

lettre	codage fixe
a	00000
b	00001
c	00010
d	00011
e	00100
f	00101
g	00110
h	00111
i	01000
j	01001
k	01010
l	01011
\overline{m}	01100
n	01101
0	01110
p	01111
q	10000
r	10001
s	10010
$oxed{t}$	10011
u	10100
v	10101
w	10110
x	10111
y	11000
z	11001
espace	11010

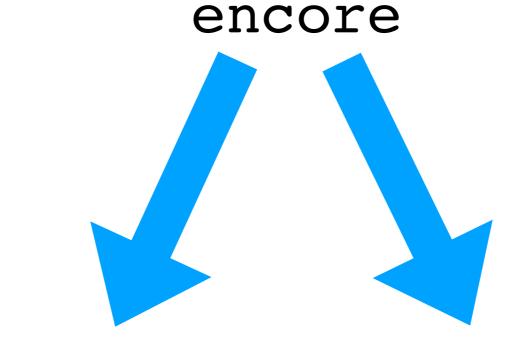
lettre	codage fixe
\overline{a}	00000
b	00001
c	00010
d	00011
e	00100
f	00101
g	00110
h	00111
i	01000
j	01001
k	01010
l	01011
$ \boxed{ m } $	01100
n	01101
О	01110
p	01111
q	10000
$oxed{r}$	10001
s	10010
$oxed{t}$	10011
u	10100
$oxed{v}$	10101
w	10110
x	10111
y	11000
\overline{z}	11001
espace	11010

n		r	
ГТ,	U	_	



lettre	codage variable
a	1010
b	0010011
c	01001
d	01110
e	110
f	0111100
g	0111110
h	0010010
i	1000
j	011111110
k	011111111001
l	0001
\overline{m}	00101
n	1001
0	0000
p	01000
\overline{q}	0111101
r	0101
s	1011
t	0110
u	0011
v	001000
\overline{w}	0111111111000
x	01111110
y	0111111111
\overline{z}	01111111101
espace	111

	Т
lettre	codage fixe
a	00000
b	00001
c	00010
d	00011
e	00100
f	00101
g	00110
h	00111
i	01000
j	01001
k	01010
l	01011
\overline{m}	01100
n	01101
0	01110
p	01111
q	10000
r	10001
s	10010
t	10011
u	10100
v	10101
w	10110
x	10111
y	11000
\overline{z}	11001
espace	11010



lettre	codage variable
a	1010
b	0010011
c	01001
d	01110
e	110
f	0111100
g	0111110
h	0010010
i	1000
j	011111110
k	011111111001
\overline{l}	0001
\overline{m}	00101
n	1001
0	0000
p	01000
q	0111101
r	0101
s	1011
t	0110
u	0011
v	001000
w	0111111111000
x	01111110
y	0111111111
z	01111111101
espace	111

_	
lettre	codage fixe
a	00000
b	00001
c	00010
d	00011
e	00100
f	00101
g	00110
h	00111
\overline{i}	01000
j	01001
k	01010
l	01011
\overline{m}	01100
n	01101
О	01110
p	01111
q	10000
r	10001
s	10010
t	10011
u	10100
\overline{v}	10101
\overline{w}	10110
x	10111
y	11000
z	11001
espace	11010

en	core
00100	110
01101	1001
00010	01001
01110	0000
10001	0101
00100	110

lettre	codage variable
\overline{a}	1010
b	0010011
\overline{c}	01001
d	01110
\overline{e}	110
f	0111100
g	0111110
h	0010010
i	1000
$\frac{j}{k}$	011111110
k	011111111001
l	0001
m	00101
n	1001
0	0000
p	01000
q	0111101
r	0101
s	1011
t	0110
u	0011
v	001000
\overline{w}	011111111000
x	01111110
y	0111111111
z	01111111101
espace	111

lettre	codage fixe
a	00000
b	00001
c	00010
d	00011
e	00100
f	00101
g	00110
h	00111
i	01000
j	01001
k	01010
l	01011
\overline{m}	01100
n	01101
О	01110
p	01111
q	10000
r	10001
s	10010
t	10011
u	10100
v	10101
\overline{w}	10110
x	10111
y	11000
z	11001
espace	11010

en	core
00100	110
01101	1001
00010	01001
01110	0000
10001	0101
00100	110
30 bits	23 bits

lettre	codage variable
a	1010
b	0010011
\overline{c}	01001
d	01110
e	110
f	0111100
g	0111110
h	0010010
i	1000
j	011111110
k	011111111001
l	0001
\overline{m}	00101
n	1001
0	0000
p	01000
\overline{q}	0111101
r	0101
s	1011
t	0110
u	0011
v	001000
\overline{w}	0111111111000
\overline{x}	01111110
\overline{y}	0111111111
z	01111111101
espace	111

lettre	codage variable
\overline{a}	1010
b	0010011
c	01001
d	01110
e	110
f	0111100
g	0111110
h	0010010
i	1000
j	011111110
k	011111111001
l	0001
m	00101
n	1001
0	0000
p	01000
q	0111101
r	0101
s	1011
t	0110
u	0011
v	001000
w	011111111000
\overline{x}	01111110
y	0111111111
z	01111111101
espace	111

<u> </u>	
lettre	codage variable
a	1010
b	0010011
c	01001
d	01110
e	110
f	0111100
g	0111110
h	0010010
i	1000
$\frac{j}{k}$	011111110
k	011111111001
l	0001
m	00101
n	1001
0	0000
p	01000
q	0111101
r	0101
s	1011
t	0110
u	0011
v	001000
\overline{w}	0111111111000
x	01111110
y	0111111111
z	01111111101
espace	111

lettre	codage variable
\overline{a}	1010
b	0010011
c	01001
d	01110
e	110
f	0111100
g	0111110
h	0010010
i	1000
$\frac{j}{k}$	011111110
k	011111111001
l	0001
m	00101
n	1001
0	0000
p	01000
q	0111101
r	0101
s	1011
t	0110
u	0011
v	001000
\overline{w}	011111111000
\overline{x}	01111110
y	0111111111
z	01111111101
espace	111

lettre	codage variable
\overline{a}	1010
b	0010011
c	01001
d	01110
e	110
f	0111100
g	0111110
h	0010010
i	1000
j	011111110
k	011111111001
l	0001
\overline{m}	00101
n	1001
0	0000
p	01000
q	0111101
r	0101
S	1011
t	0110
u	0011
v	001000
\overline{w}	011111111000
x	01111110
y	0111111111
z	01111111101
espace	111

lettre	codage variable
\overline{a}	1010
b	0010011
c	01001
d	01110
e	110
f	0111100
g	0111110
h	0010010
i	1000
j	011111110
k	011111111001
l	0001
m	00101
n	1001
0	0000
p	01000
q	0111101
r	0101
s	1011
t	0110
u	0011
v	001000
\overline{w}	011111111000
x	01111110
y	0111111111
z	01111111101
espace	111

lettre	codage variable
\overline{a}	1010
b	0010011
c	01001
d	01110
e	110
f	0111100
g	0111110
h	0010010
i	1000
$\frac{j}{k}$	011111110
k	011111111001
l	0001
m	00101
n	1001
0	0000
p	01000
q	0111101
r	0101
s	1011
t	0110
u	0011
v	001000
\overline{w}	011111111000
\overline{x}	01111110
y	0111111111
z	01111111101
espace	111

<u> </u>	
lettre	codage variable
\overline{a}	1010
b	0010011
c	01001
d	01110
e	110
f	0111100
g	0111110
h	0010010
i	1000
$\frac{j}{k}$	011111110
k	011111111001
l	0001
m	00101
n	1001
0	0000
p	01000
\overline{q}	0111101
r	0101
s	1011
t	0110
u	0011
\overline{v}	001000
\overline{w}	011111111000
\overline{x}	01111110
y	0111111111
z	01111111101
espace	111

lettre	codage variable
\overline{a}	1010
b	0010011
c	01001
d	01110
e	110
f	0111100
g	0111110
h	0010010
i	1000
j	011111110
k	011111111001
l	0001
\overline{m}	00101
n	1001
0	0000
p	01000
\overline{q}	0111101
r	0101
S	1011
t	0110
u	0011
v	001000
\overline{w}	011111111000
\overline{x}	01111110
y	0111111111
z	01111111101
espace	111

lettre	codage variable
\overline{a}	1010
b	0010011
c	01001
d	01110
e	110
f	0111100
$\mid g \mid$	0111110
h	0010010
i	1000
$\frac{j}{k}$	011111110
k	011111111001
l	0001
m	00101
n	1001
0	0000
p	01000
q	0111101
r	0101
s	1011
t	0110
u	0011
v	001000
\overline{w}	011111111000
x	01111110
y	0111111111
z	01111111101
espace	111

lettre	codage variable
a	1010
b	0010011
c	01001
d	01110
e	110
f	0111100
g	0111110
h	0010010
\overline{i}	1000
$\frac{j}{k}$	011111110
k	011111111001
l	0001
\overline{m}	00101
n	1001
0	0000
p	01000
q	0111101
r	0101
s	1011
t	0110
u	0011
v	001000
w	0111111111000
x	01111110
y	0111111111
\overline{z}	01111111101
espace	111

10001001011110000000101
i n f o r

lettre	codage variable
a	1010
b	0010011
c	01001
d	01110
e	110
f	0111100
g	0111110
h	0010010
i	1000
j	011111110
$\frac{s}{k}$	011111111001
l	0001
m	00101
n	1001
0	0000
p	01000
q	0111101
r	0101
s	1011
t	0110
u	0011
v	001000
\overline{w}	011111111000
\overline{x}	01111110
y	0111111111
z	01111111101
espace	111

lettre	codage variable
a	1010
b	0010011
c	01001
d	01110
e	110
f	0111100
g	0111110
h	0010010
i	1000
j	011111110
$\frac{s}{k}$	011111111001
l	0001
m	00101
n	1001
0	0000
p	01000
q	0111101
r	0101
s	1011
t	0110
u	0011
v	001000
\overline{w}	011111111000
\overline{x}	01111110
y	0111111111
z	01111111101
espace	111

lettre	codage variable
a	1010
b	0010011
c	01001
d	01110
e	110
f	0111100
g	0111110
h	0010010
i	1000
$\frac{j}{k}$	011111110
k	011111111001
l	0001
m	00101
n	1001
0	0000
p	01000
q	0111101
r	0101
s	1011
t	0110
u	0011
v	001000
w	011111111000
x	01111110
y	0111111111
z	01111111101
espace	111

10001001011110000000101001011010 i n f o r m

lettre	codage variable
a	1010
b	0010011
c	01001
d	01110
e	110
f	0111100
g	0111110
h	0010010
i	1000
$\frac{j}{k}$	011111110
k	011111111001
l	0001
m	00101
n	1001
0	0000
p	01000
q	0111101
r	0101
s	1011
t	0110
u	0011
v	001000
\overline{w}	011111111000
\overline{x}	01111110
y	0111111111
z	01111111101
espace	111

lettre	codage variable
\overline{a}	1010
b	0010011
c	01001
d	01110
e	110
f	0111100
$\mid g \mid$	0111110
h	0010010
i	1000
$\frac{j}{k}$	011111110
k	011111111001
l	0001
m	00101
n	1001
0	0000
p	01000
q	0111101
r	0101
s	1011
t	0110
u	0011
v	001000
\overline{w}	011111111000
x	01111110
y	0111111111
z	01111111101
espace	111

100010010111100000001010010110100110 i n f orma

<u> </u>	
lettre	codage variable
\overline{a}	1010
b	0010011
c	01001
d	01110
e	110
f	0111100
g	0111110
h	0010010
i	1000
$\frac{j}{k}$	011111110
k	011111111001
l	0001
m	00101
\overline{n}	1001
0	0000
p	01000
\overline{q}	0111101
r	0101
s	1011
t	0110
u	0011
\overline{v}	001000
\overline{w}	011111111000
\overline{x}	01111110
y	0111111111
z	01111111101
espace	111

100010010111100000001010010110100110 i n f o r m a t

lettre	codage variable
\overline{a}	1010
b	0010011
c	01001
d	01110
e	110
f	0111100
g	0111110
h	0010010
i	1000
$\frac{j}{k}$	011111110
k	011111111001
l	0001
\overline{m}	00101
n	1001
0	0000
p	01000
q	0111101
r	0101
s	1011
t	0110
\overline{u}	0011
v	001000
\overline{w}	011111111000
\overline{x}	01111110
\overline{y}	0111111111
z	01111111101
espace	111

lettre	codage variable
a	1010
b	0010011
c	01001
d	01110
e	110
f	0111100
g	0111110
h	0010010
i	1000
$\frac{j}{k}$	011111110
k	011111111001
l	0001
\overline{m}	00101
n	1001
0	0000
p	01000
q	0111101
r	0101
S	1011
t	0110
u	0011
v	001000
\overline{w}	011111111000
x	01111110
y	0111111111
z	01111111101
espace	111

lettre	codage variable
\overline{a}	1010
b	0010011
c	01001
d	01110
e	110
f	0111100
g	0111110
h	0010010
i	1000
j	011111110
k	011111111001
l	0001
m	00101
n	1001
0	0000
p	01000
q	0111101
r	0101
s	1011
t	0110
u	0011
v	001000
w	011111111000
\overline{x}	01111110
\overline{y}	0111111111
z	01111111101
espace	111

lettre	codage variable
a	1010
b	0010011
c	01001
d	01110
e	110
f	0111100
g	0111110
h	0010010
i	1000
j	011111110
k	011111111001
l	0001
m	00101
n	1001
0	0000
p	01000
q	0111101
r	0101
s	1011
t	0110
u	0011
v	001000
w	0111111111000
x	01111110
y	0111111111
z	01111111101
espace	111

lettre	codage variable
a	1010
b	0010011
c	01001
d	01110
e	110
f	0111100
g	0111110
h	0010010
i	1000
$\frac{j}{k}$	011111110
k	011111111001
l	0001
m	00101
n	1001
0	0000
p	01000
q	0111101
r	0101
s	1011
t	0110
u	0011
v	001000
\overline{w}	011111111000
x	01111110
y	0111111111
z	01111111101
espace	111

lettre	codage variable
\overline{a}	1010
b	0010011
c	01001
d	01110
e	110
f	0111100
g	0111110
h	0010010
i	1000
$\frac{j}{k}$	011111110
k	011111111001
l	0001
\overline{m}	00101
n	1001
0	0000
p	01000
q	0111101
r	0101
s	1011
t	0110
u	0011
v	001000
\overline{w}	011111111000
\overline{x}	01111110
\overline{y}	0111111111
z	01111111101
espace	111

lettre	codage variable
\overline{a}	1010
b	0010011
c	01001
d	01110
e	110
f	0111100
g	0111110
h	0010010
i	1000
$\frac{j}{k}$	011111110
k	011111111001
l	0001
m	00101
n	1001
0	0000
p	01000
q	0111101
r	0101
s	1011
t	0110
u	0011
\overline{v}	001000
\overline{w}	011111111000
x	01111110
\overline{y}	0111111111
\overline{z}	01111111101
espace	111

Pourquoi n'a-t-on jamais le choix du décodage?

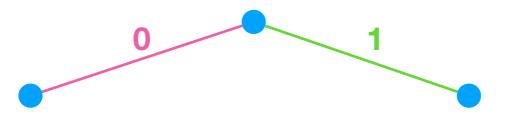
lettre	codage variable
\overline{a}	1010
b	0010011
c	01001
d	01110
e	110
f	0111100
g	0111110
h	0010010
i	1000
$\frac{j}{k}$	011111110
k	011111111001
l	0001
m	00101
\overline{n}	1001
0	0000
p	01000
\overline{q}	0111101
r	0101
s	1011
t	0110
u	0011
v	001000
\overline{w}	0111111111000
x	01111110
y	0111111111
z	01111111101
espace	111

Pourquoi n'a-t-on jamais le choix du décodage?

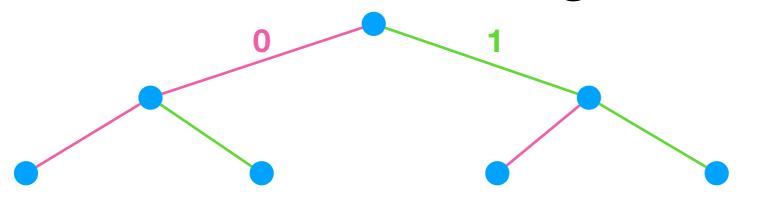
Aucun code n'est le **préfixe** d'un autre...

lettre	codage variable
\overline{a}	1010
b	0010011
c	01001
d	01110
e	110
f	0111100
g	0111110
h	0010010
i	1000
$\frac{j}{k}$	011111110
k	011111111001
l	0001
m	00101
n	1001
0	0000
p	01000
\overline{q}	0111101
r	0101
s	1011
t	0110
u	0011
v	001000
\overline{w}	0111111111000
x	01111110
y	0111111111
z	01111111101
espace	111

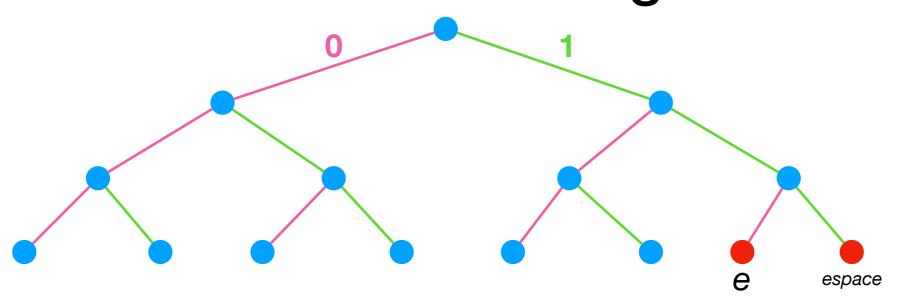
lettre	codage variable
a	1010
b	0010011
c	01001
d	01110
e	110
f	0111100
g	0111110
h	0010010
i	1000
j	011111110
$\frac{v}{k}$	011111111001
l	0001
m	00101
n	1001
0	0000
p	01000
q	0111101
r	0101
s	1011
t	0110
u	0011
v	001000
\overline{w}	011111111000
x	01111110
\overline{y}	0111111111
\overline{z}	01111111101
espace	111



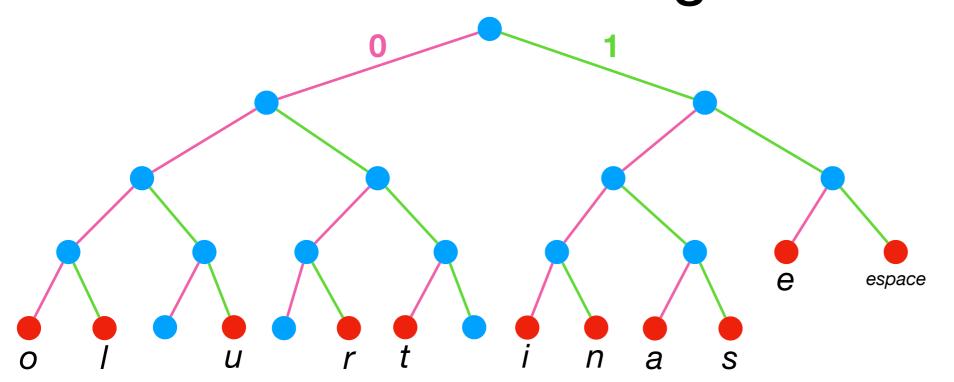
lettre	codage variable
a	1010
b	0010011
c	01001
d	01110
e	110
f	0111100
g	0111110
h	0010010
i	1000
j	011111110
k	011111111001
l	0001
\overline{m}	00101
n	1001
О	0000
p	01000
q	0111101
r	0101
s	1011
t	0110
u	0011
v	001000
\overline{w}	011111111000
x	01111110
y	0111111111
z	01111111101
espace	111



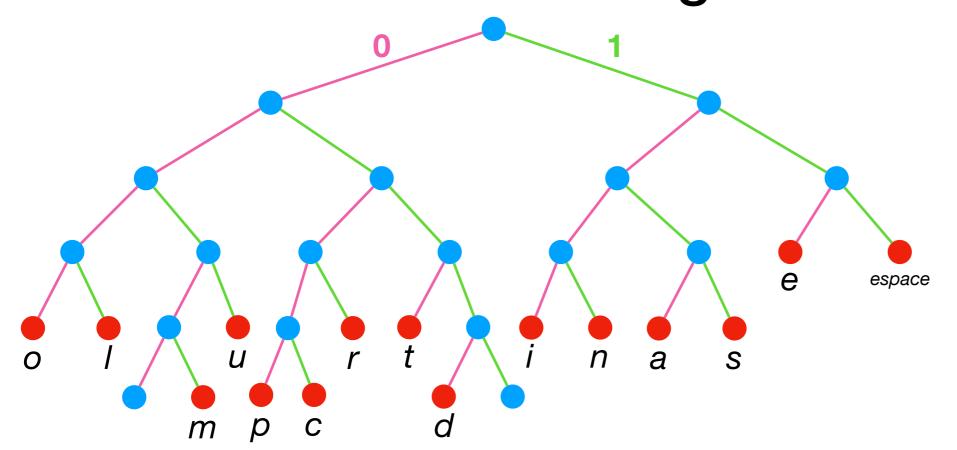
lettre	codage variable
a	1010
b	0010011
c	01001
d	01110
e	110
f	0111100
g	0111110
h	0010010
i	1000
$\frac{j}{k}$	011111110
k	011111111001
l	0001
m	00101
n	1001
0	0000
p	01000
q	0111101
r	0101
s	1011
t	0110
u	0011
v	001000
\overline{w}	0111111111000
\overline{x}	01111110
y	0111111111
z	01111111101
espace	111



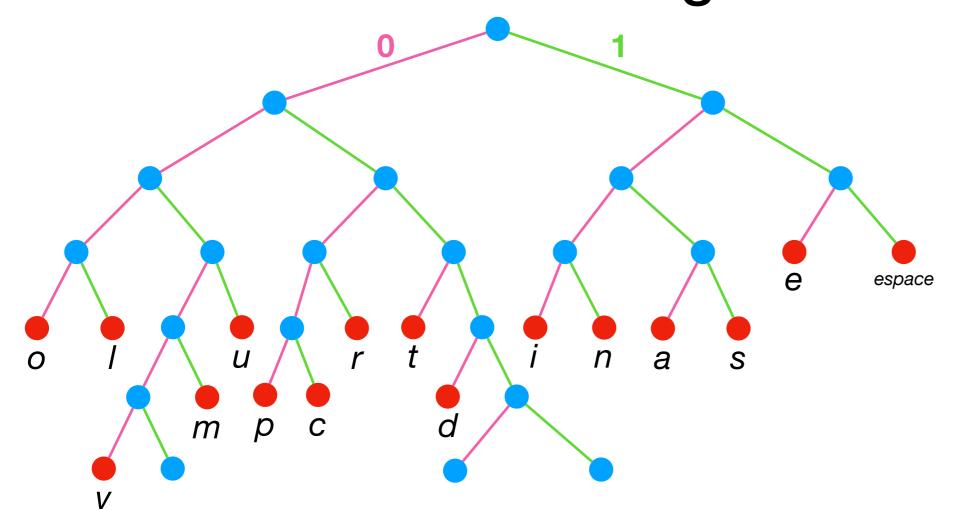
7	
lettre	codage variable
a	1010
b	0010011
c	01001
d	01110
e	110
f	0111100
g	0111110
h	0010010
i	1000
j	011111110
$\frac{s}{k}$	011111111001
l	0001
m	00101
n	1001
0	0000
p	01000
q	0111101
r	0101
S	1011
t	0110
u	0011
v	001000
\overline{w}	011111111000
\overline{x}	01111110
y	0111111111
z	01111111101
espace	111



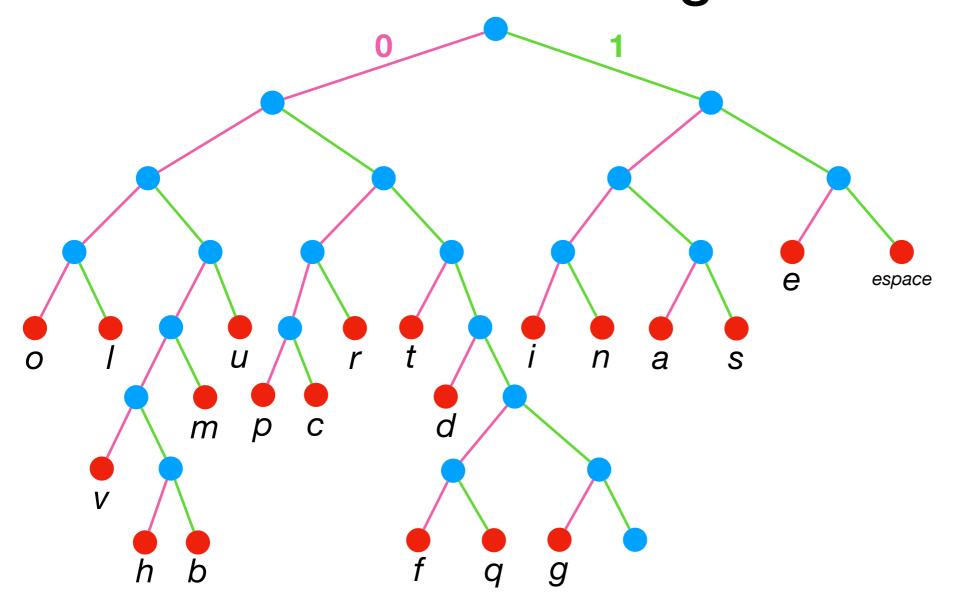
7	
lettre	codage variable
a	1010
b	0010011
c	01001
d	01110
e	110
f	0111100
g	0111110
h	0010010
i	1000
j	011111110
$\frac{s}{k}$	011111111001
l	0001
m	00101
n	1001
0	0000
p	01000
q	0111101
r	0101
S	1011
t	0110
u	0011
v	001000
\overline{w}	011111111000
\overline{x}	01111110
y	0111111111
z	01111111101
espace	111



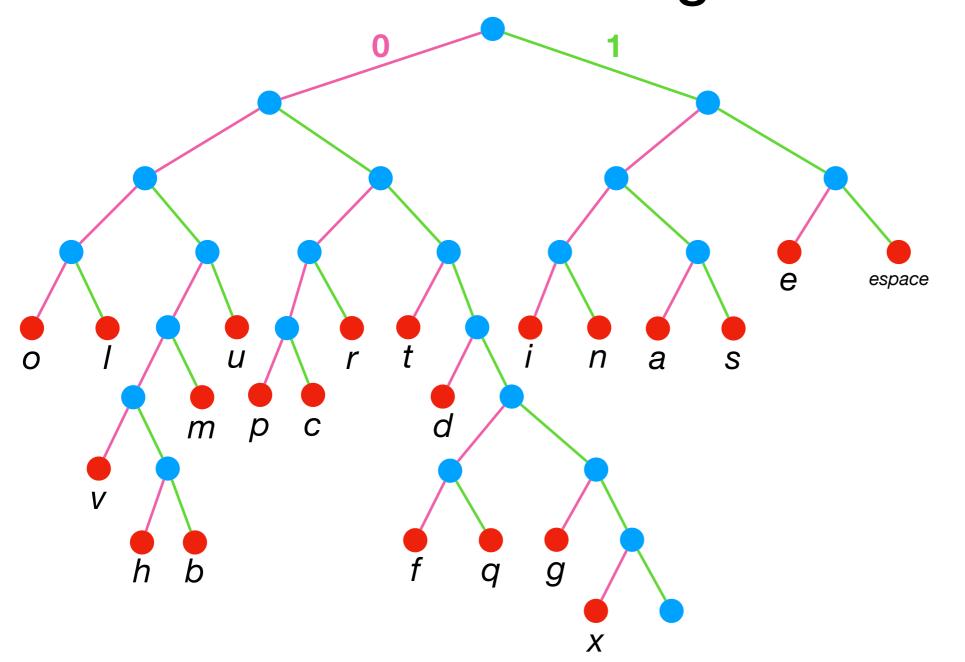
lettre	codage variable
\overline{a}	1010
b	0010011
c	01001
d	01110
e	110
f	0111100
g	0111110
h	0010010
i	1000
$\frac{j}{k}$	011111110
k	011111111001
l	0001
\overline{m}	00101
n	1001
0	0000
p	01000
\overline{q}	0111101
r	0101
s	1011
t	0110
u	0011
v	001000
\overline{w}	011111111000
\overline{x}	01111110
y	0111111111
z	01111111101
espace	111



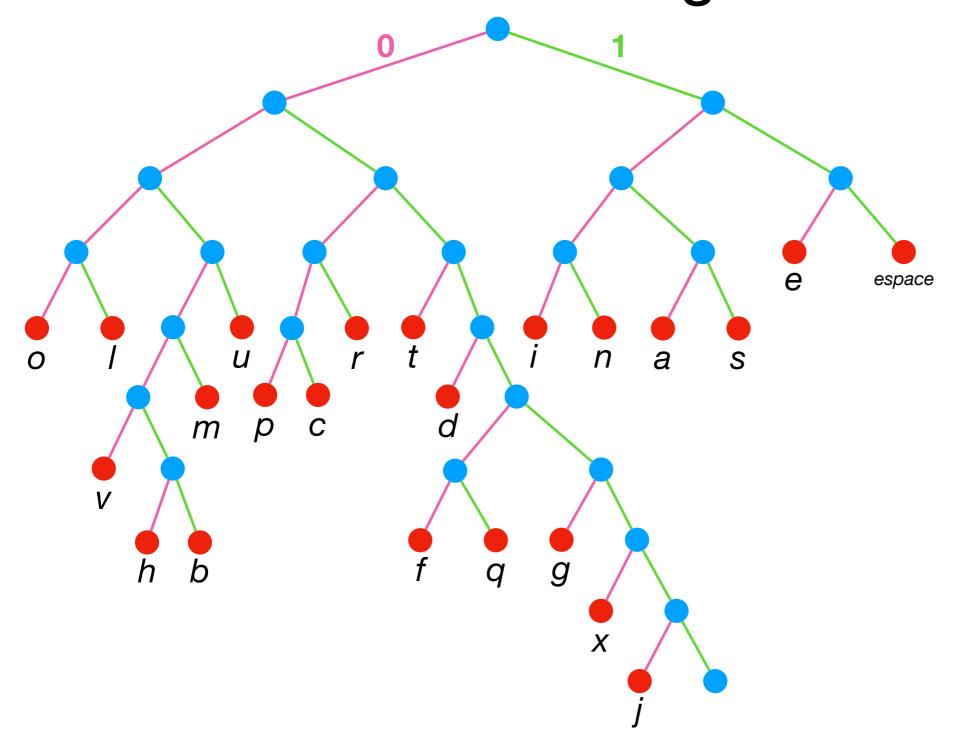
lettre	codage variable
\overline{a}	1010
b	0010011
c	01001
d	01110
e	110
f	0111100
g	0111110
h	0010010
i	1000
j	011111110
k	011111111001
l	0001
\overline{m}	00101
n	1001
0	0000
p	01000
\overline{q}	0111101
r	0101
s	1011
t	0110
u	0011
v	001000
\overline{w}	011111111000
x	01111110
y	0111111111
z	01111111101
espace	111



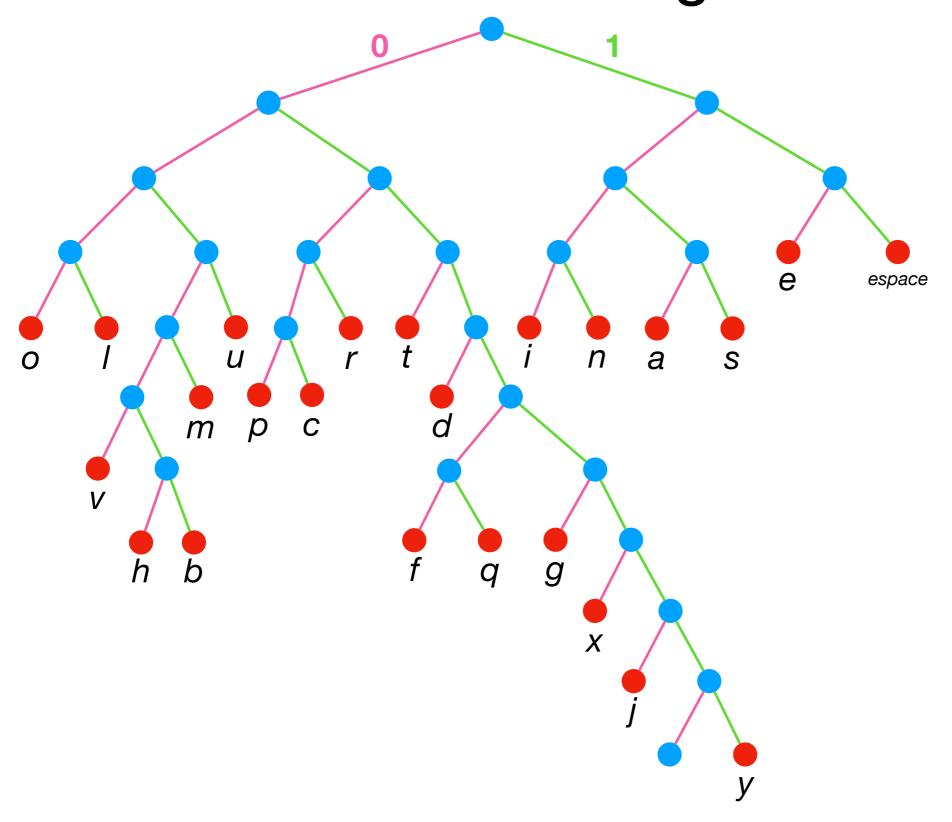
lettre	codage variable
\overline{a}	1010
b	0010011
c	01001
d	01110
e	110
f	0111100
g	0111110
h	0010010
i	1000
$\frac{j}{k}$	011111110
k	011111111001
l	0001
\overline{m}	00101
n	1001
0	0000
p	01000
\overline{q}	0111101
r	0101
s	1011
t	0110
u	0011
v	001000
\overline{w}	011111111000
\overline{x}	01111110
y	0111111111
z	01111111101
espace	111



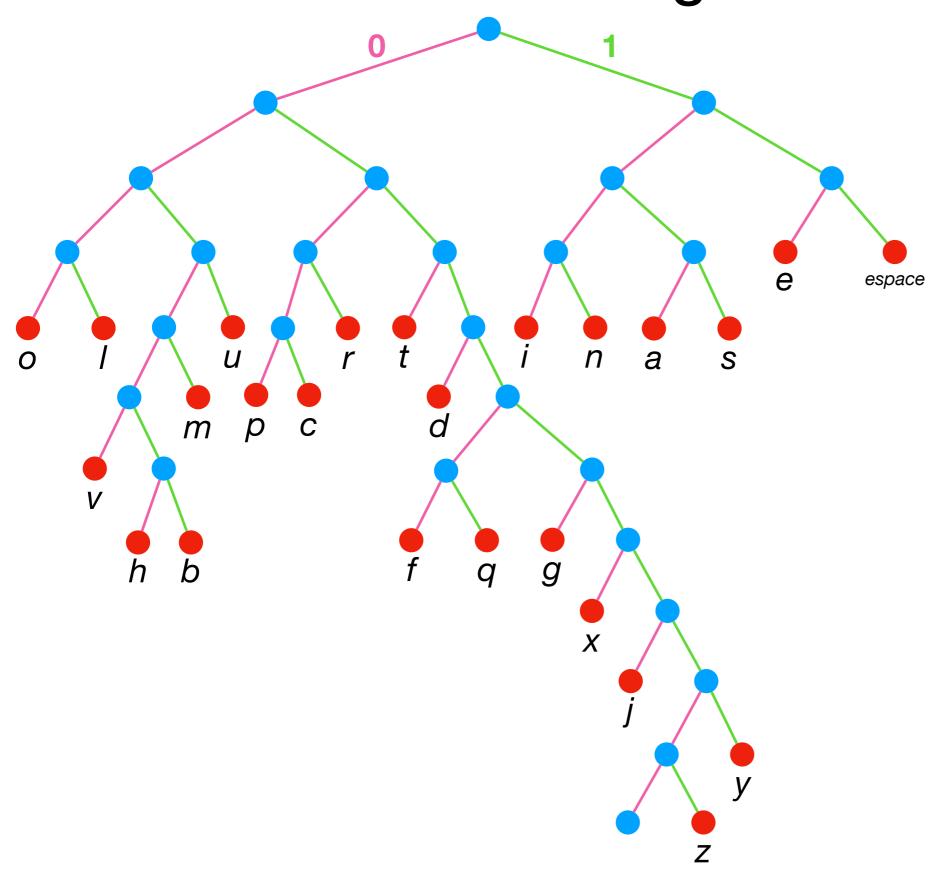
lettre	codage variable
\overline{a}	1010
b	0010011
c	01001
d	01110
e	110
f	0111100
g	0111110
h	0010010
i	1000
$\frac{j}{k}$	011111110
k	011111111001
l	0001
m	00101
\overline{n}	1001
0	0000
p	01000
\overline{q}	0111101
r	0101
S	1011
t	0110
u	0011
v	001000
\overline{w}	0111111111000
\overline{x}	01111110
y	0111111111
z	01111111101
\overline{espace}	111



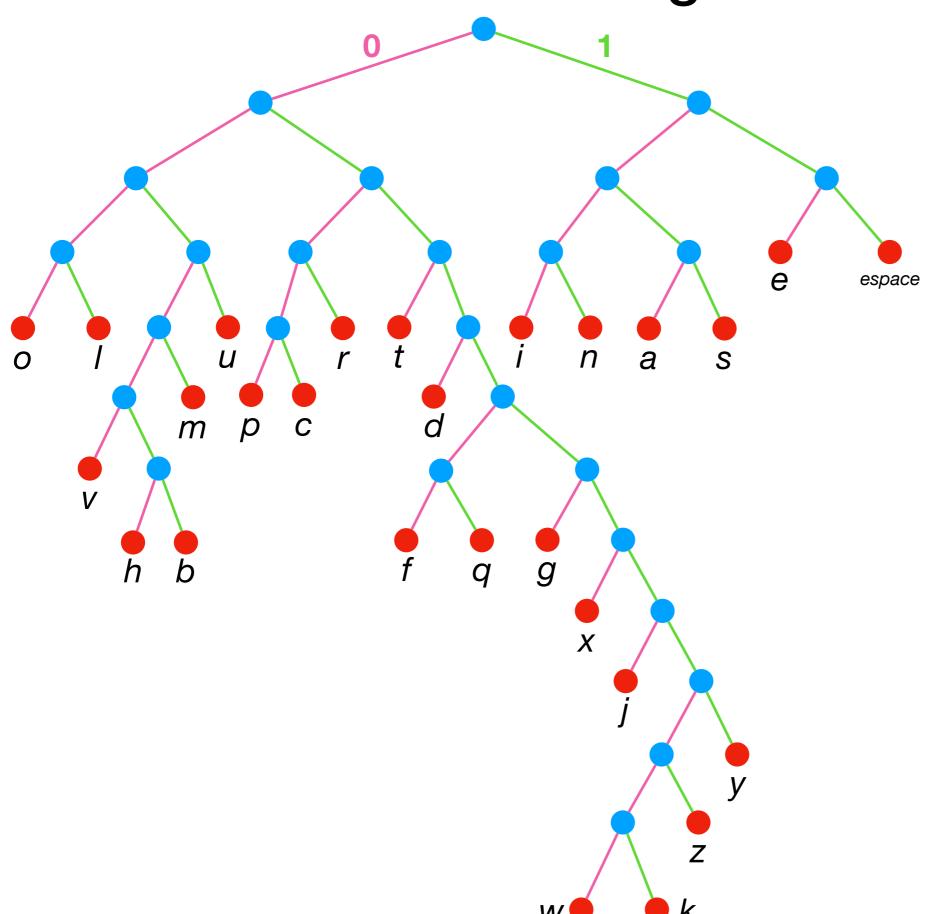
codage variable
1010
0010011
01001
01110
110
0111100
0111110
0010010
1000
011111110
011111111001
0001
00101
1001
0000
01000
0111101
0101
1011
0110
0011
001000
011111111000
01111110
0111111111
01111111101
111



lettre	codage variable
a	1010
b	0010011
c	01001
d	01110
e	110
f	0111100
g	0111110
h	0010010
i	1000
j	011111110
k	011111111001
l	0001
m	00101
n	1001
0	0000
p	01000
q	0111101
r	0101
s	1011
t	0110
u	0011
v	001000
\overline{w}	011111111000
x	01111110
y	0111111111
z	01111111101
espace	111

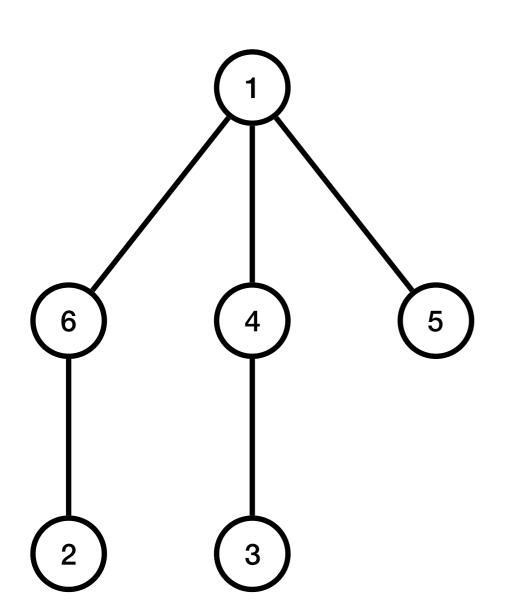


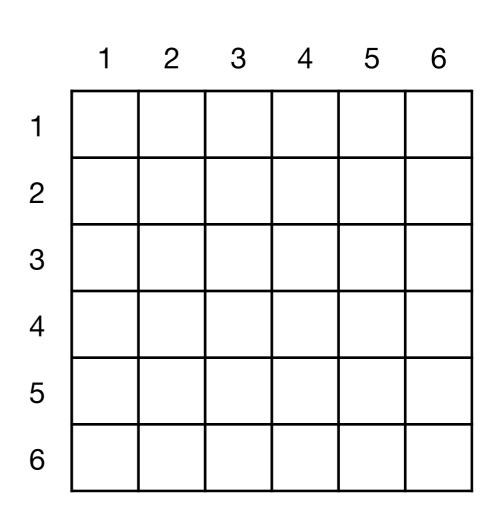
lettre	codage variable
a	1010
b	0010011
c	01001
d	01110
e	110
f	0111100
g	0111110
h	0010010
i	1000
j	011111110
k	011111111001
l	0001
m	00101
n	1001
0	0000
p	01000
q	0111101
r	0101
s	1011
t	0110
u	0011
v	001000
\overline{w}	011111111000
x	01111110
y	0111111111
z	01111111101
espace	111



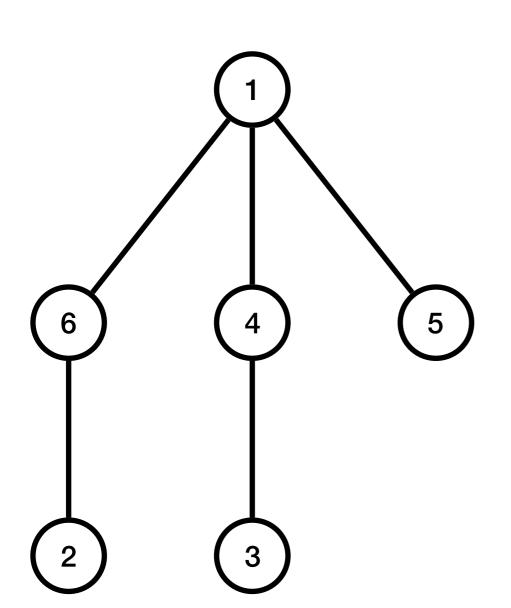
lettre	codage variable
a	1010
b	0010011
\overline{c}	01001
d	01110
e	110
f	0111100
g	0111110
h	0010010
i	1000
\overline{j}	011111110
k	011111111001
l	0001
m	00101
\overline{n}	1001
0	0000
p	01000
\overline{q}	0111101
r	0101
s	1011
t	0110
\overline{u}	0011
v	001000
\overline{w}	0111111111000
\overline{x}	01111110
y	0111111111
z	01111111101
\overline{espace}	111

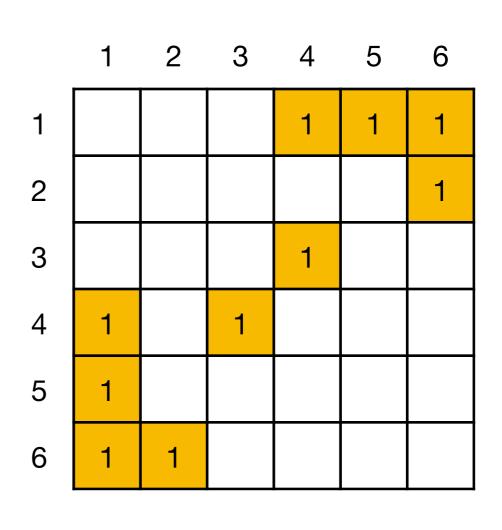
Représentation d'arbres en tant que graphes



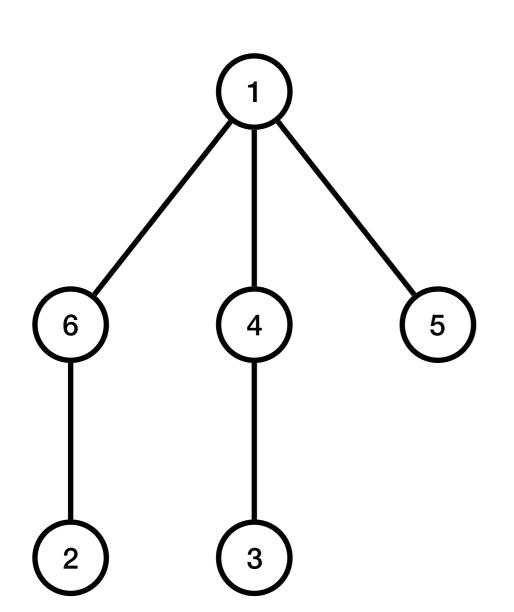


Représentation d'arbres en tant que graphes



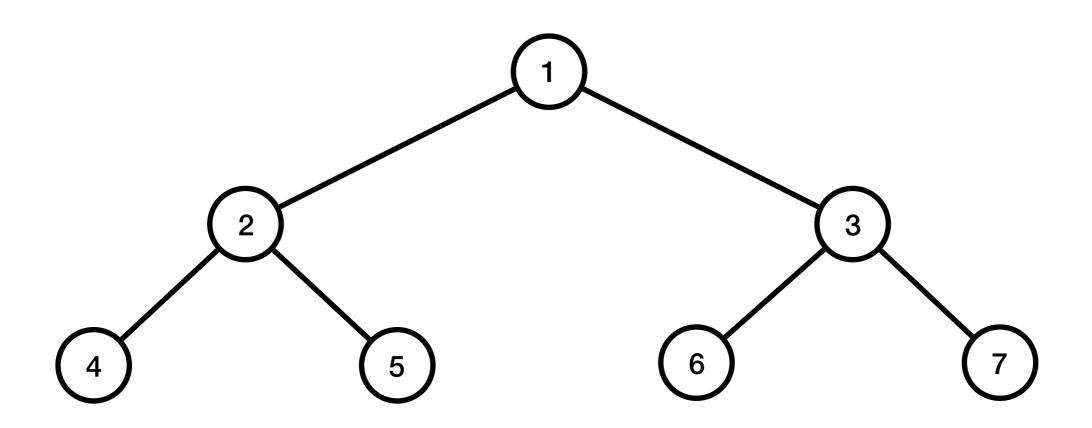


Représentation d'arbres en tant que graphes

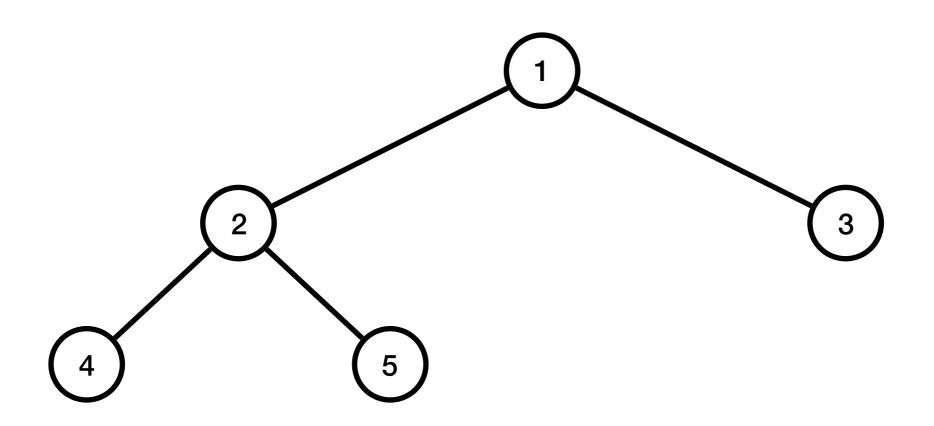


	1	2	3	4	5	6
1	0	0	0	1	1	1
2	0	0	0	0	0	1
3	0	0	0	1	0	0
4	1	0	1	0	0	0
5	1	0	0	0	0	0
6	1	1	0	0	0	0

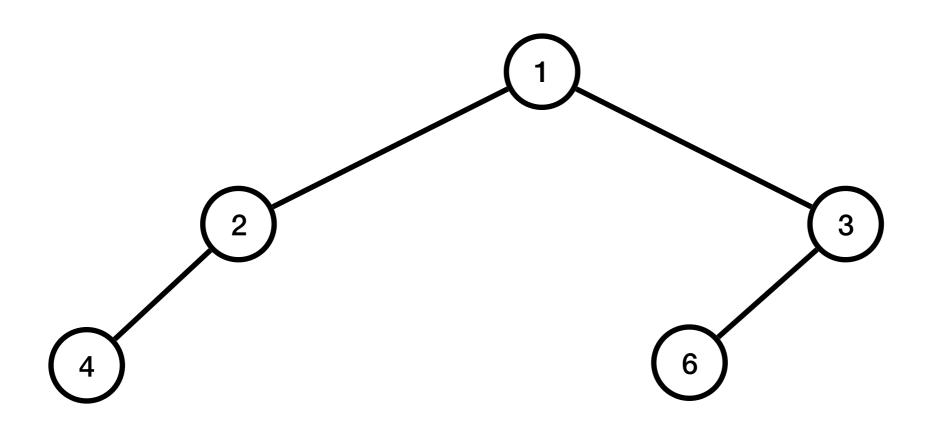
Arbres binaires



Arbres binaires



Arbres binaires



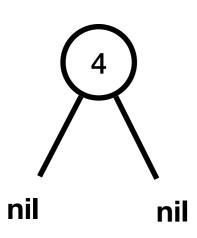
Arbres binaires : définition récursive

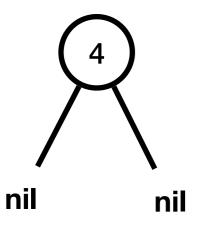
Un arbre binaire est:

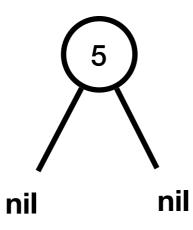
- soit l'arbre vide, qu'on dénote par nil
- soit un nœud avec une certaine valeur (par exemple, un nombre entier), un sous-arbre gauche et un sous-arbre droit; les sous-arbres sont, à leur tour, des arbres binaires

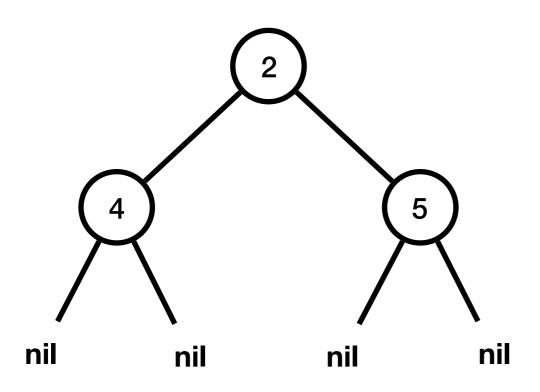
nil

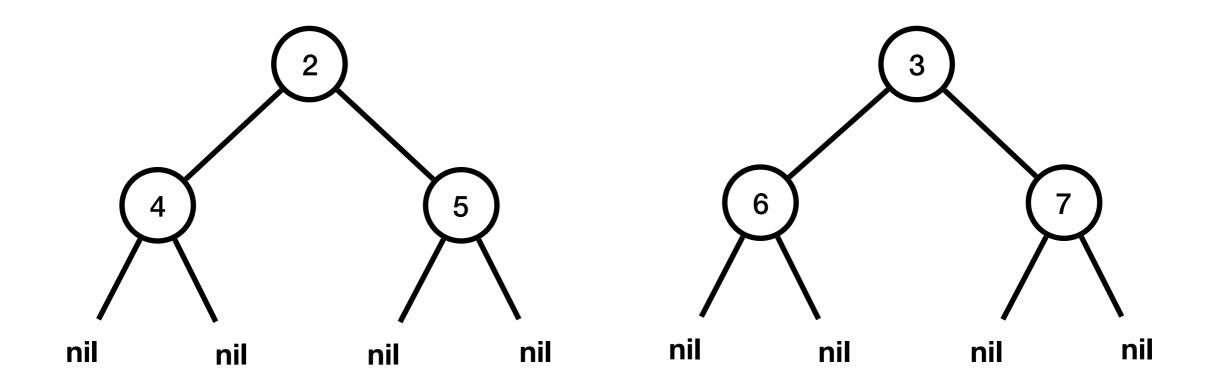
nil

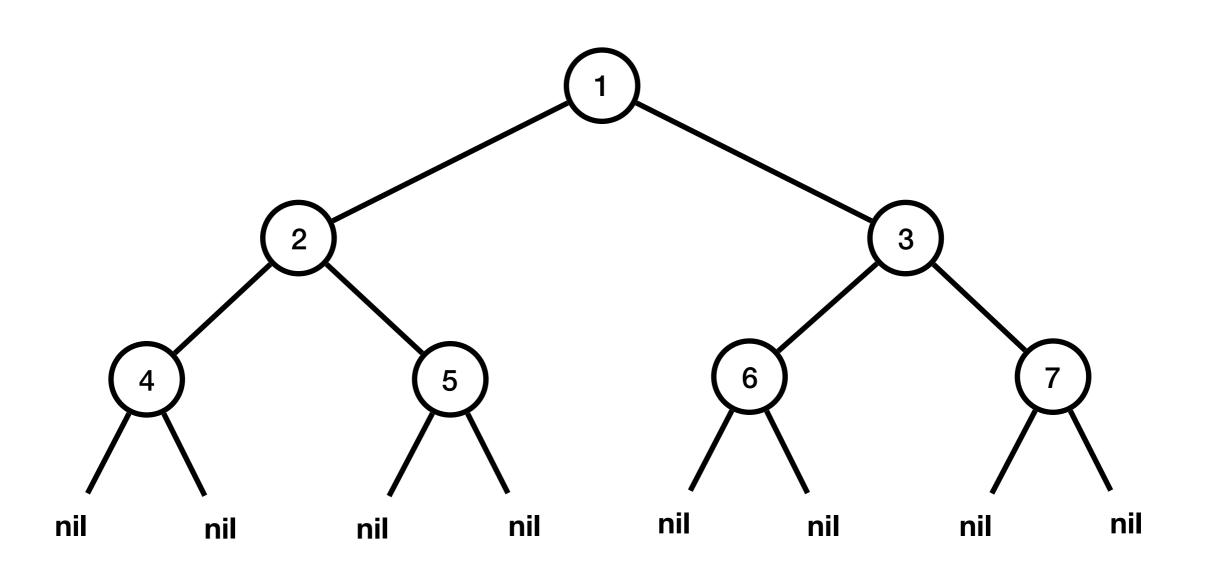


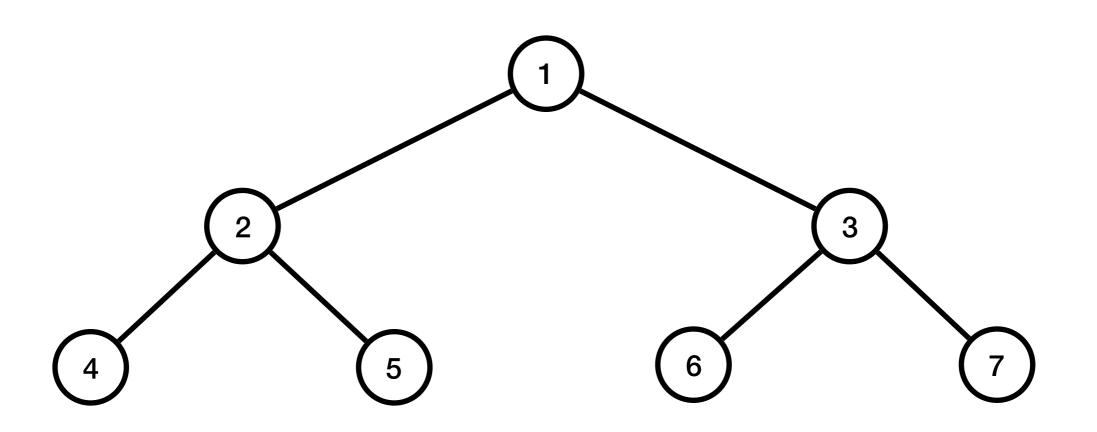


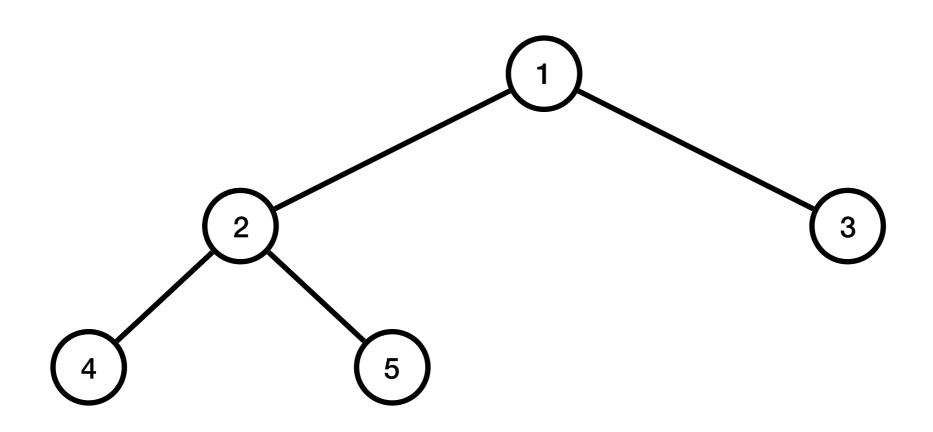


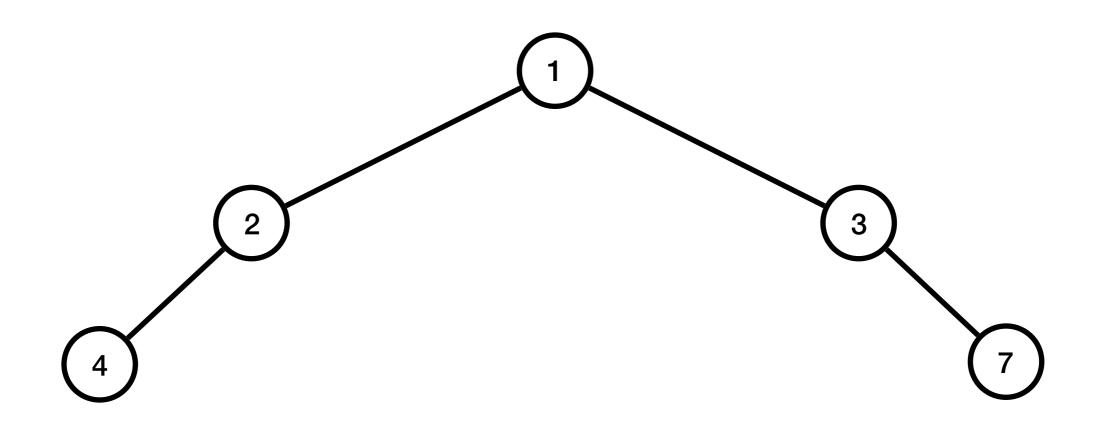


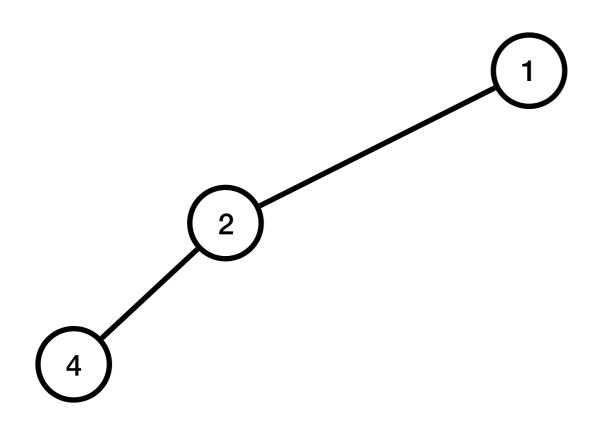


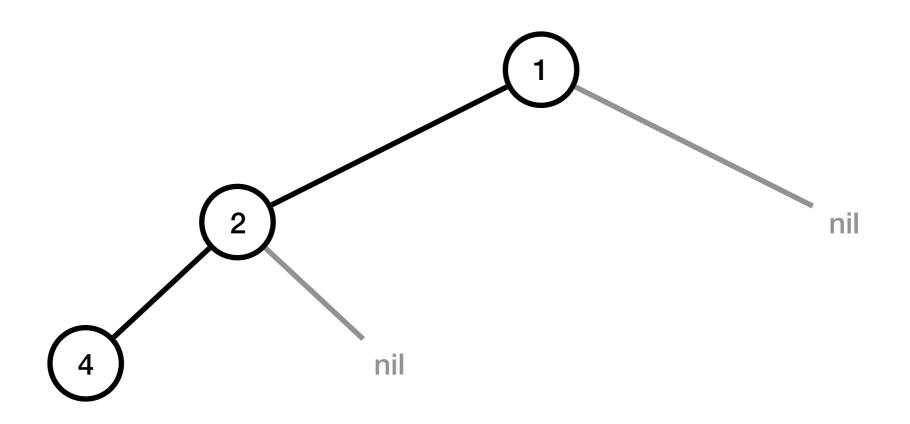




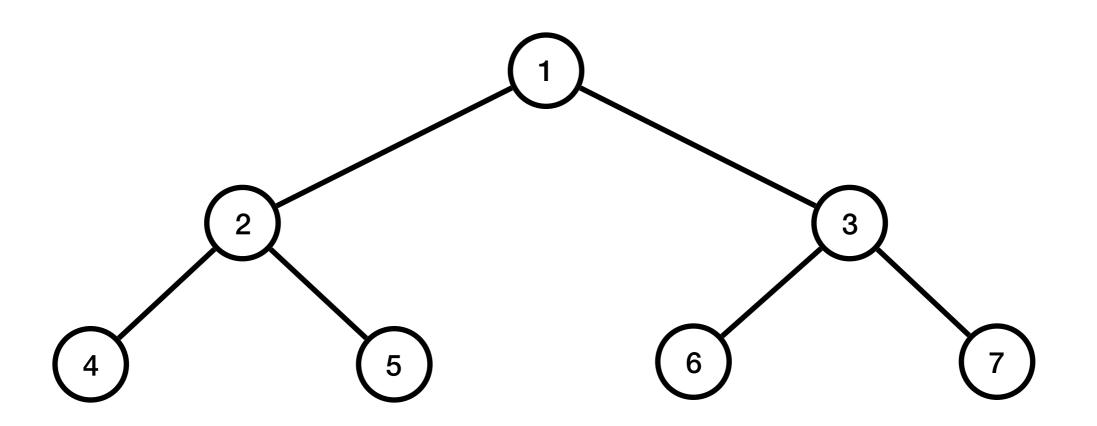




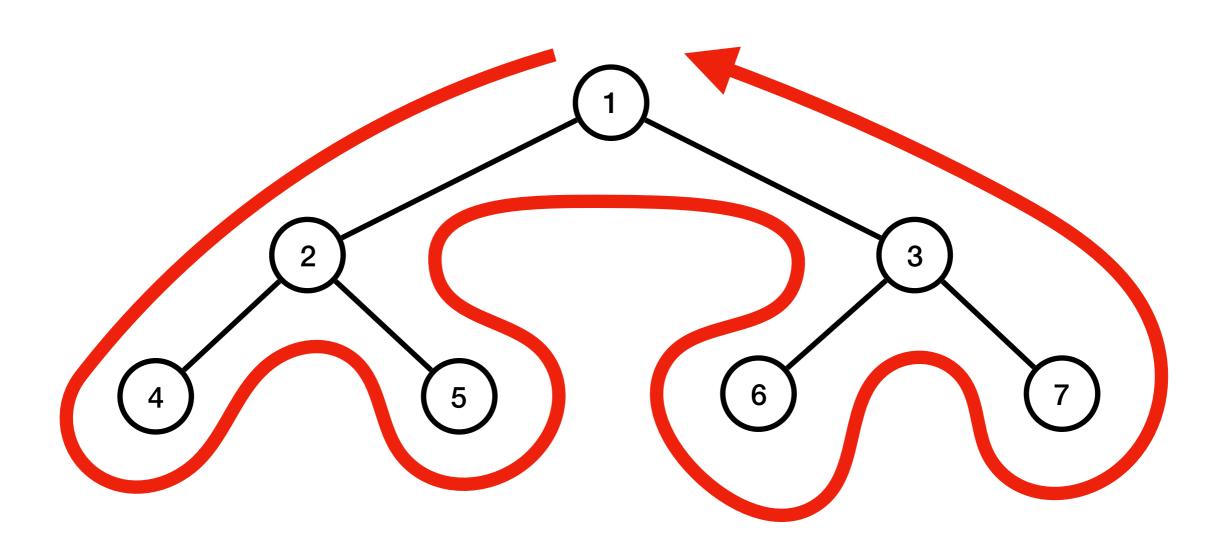




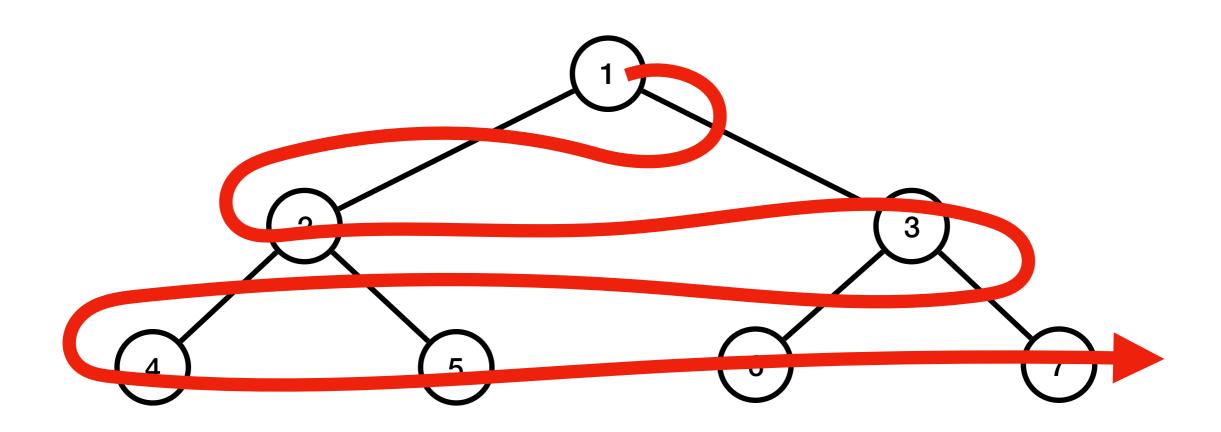
Parcours d'arbres binaires



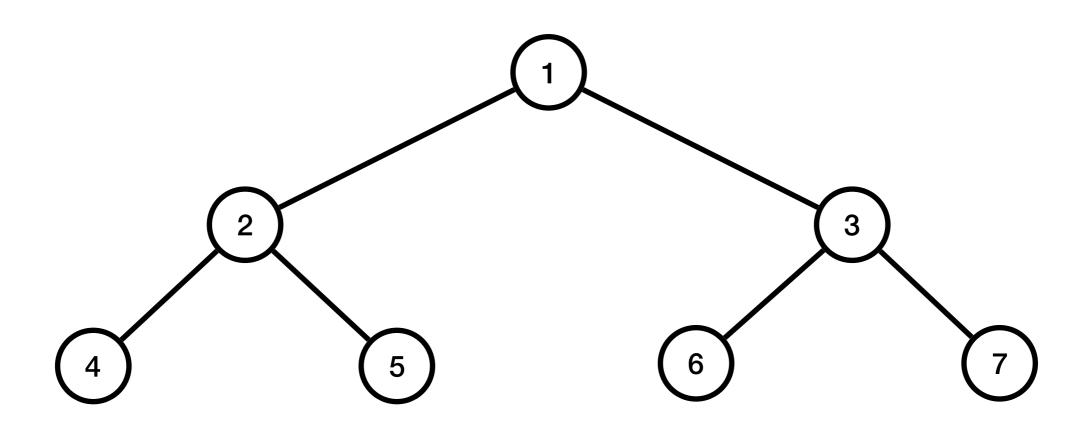
Parcours d'arbres binaires

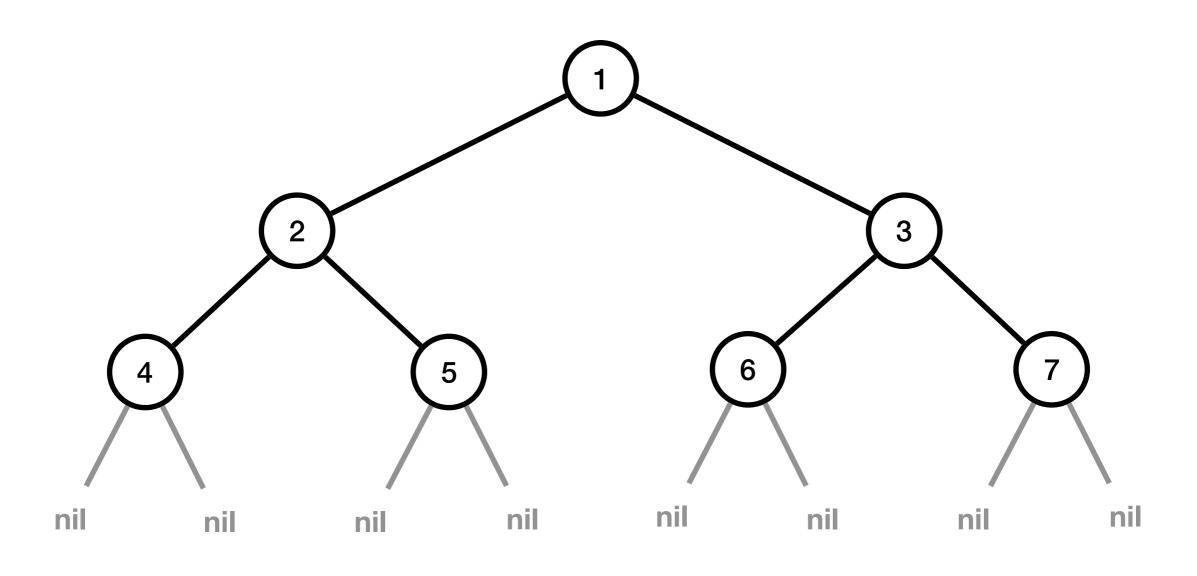


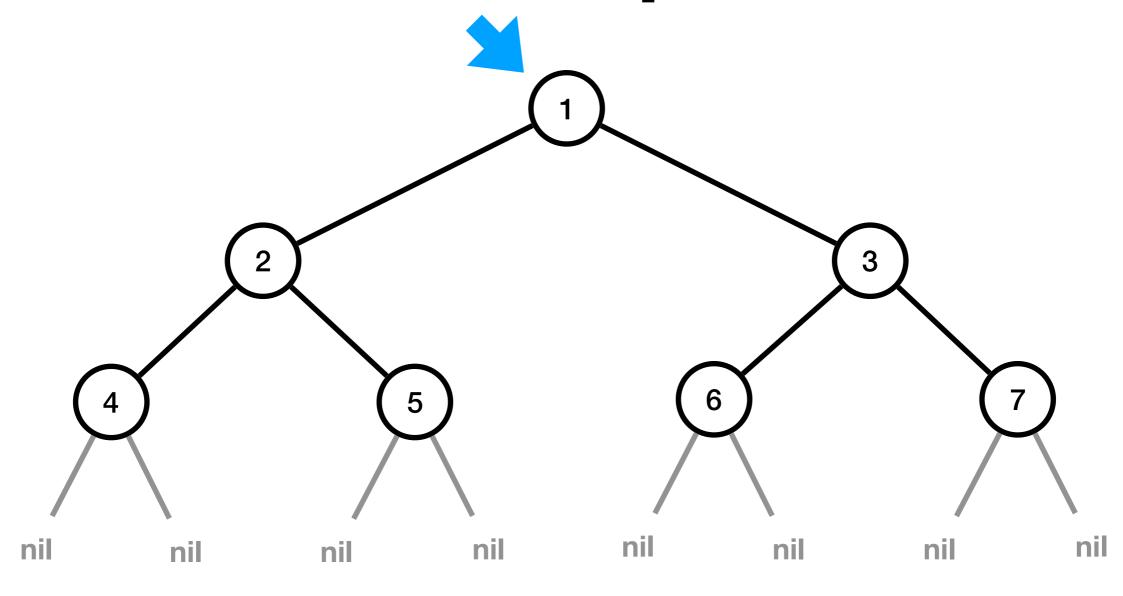
Parcours d'arbres binaires

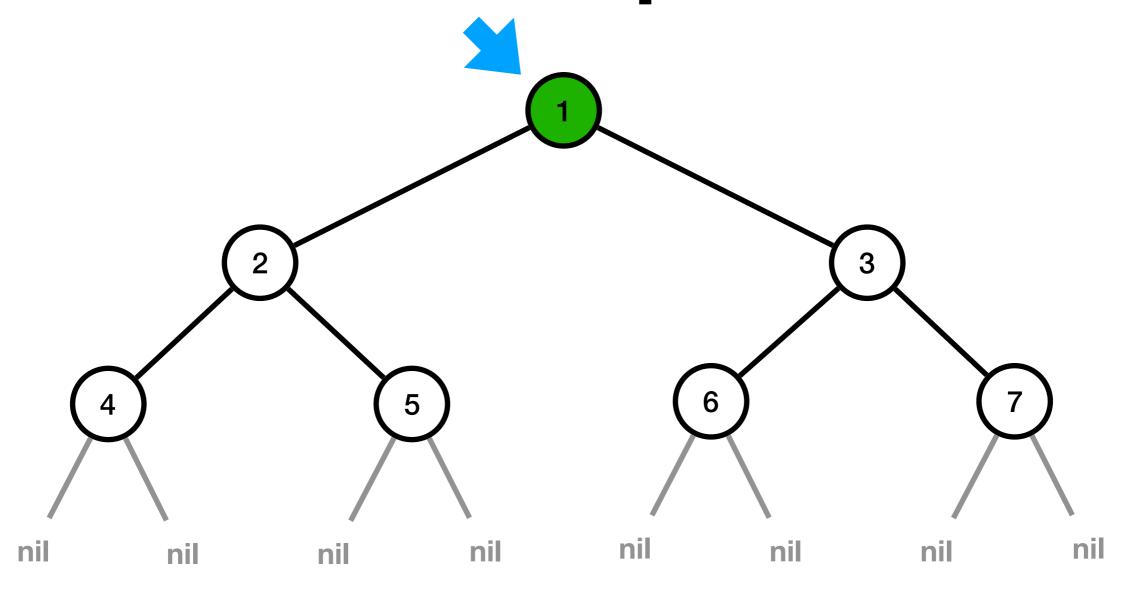


```
procédure visiter-préfixe(nœud)
    si nœud ≠ nil alors
    afficher(valeur[nœud])
    visiter-préfixe(gauche[nœud])
    visiter-préfixe(droite[nœud])
```

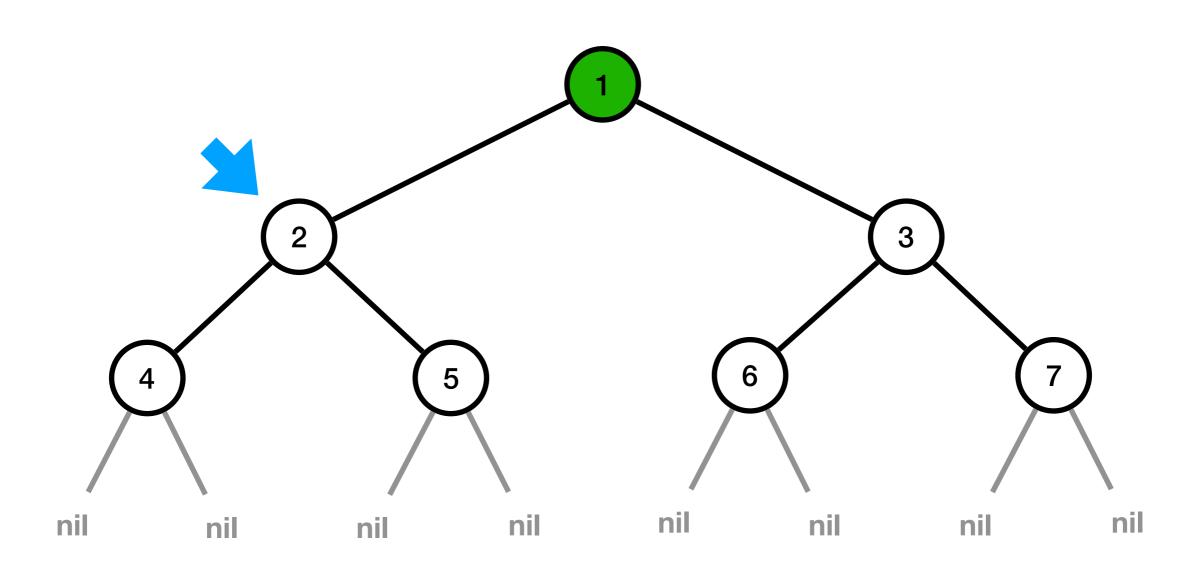




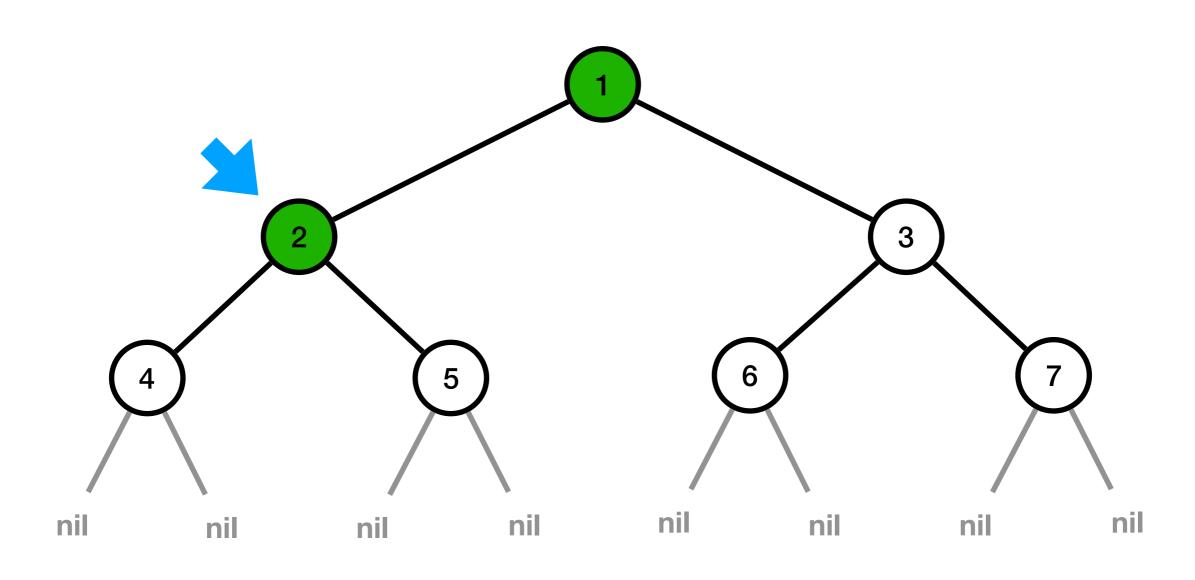




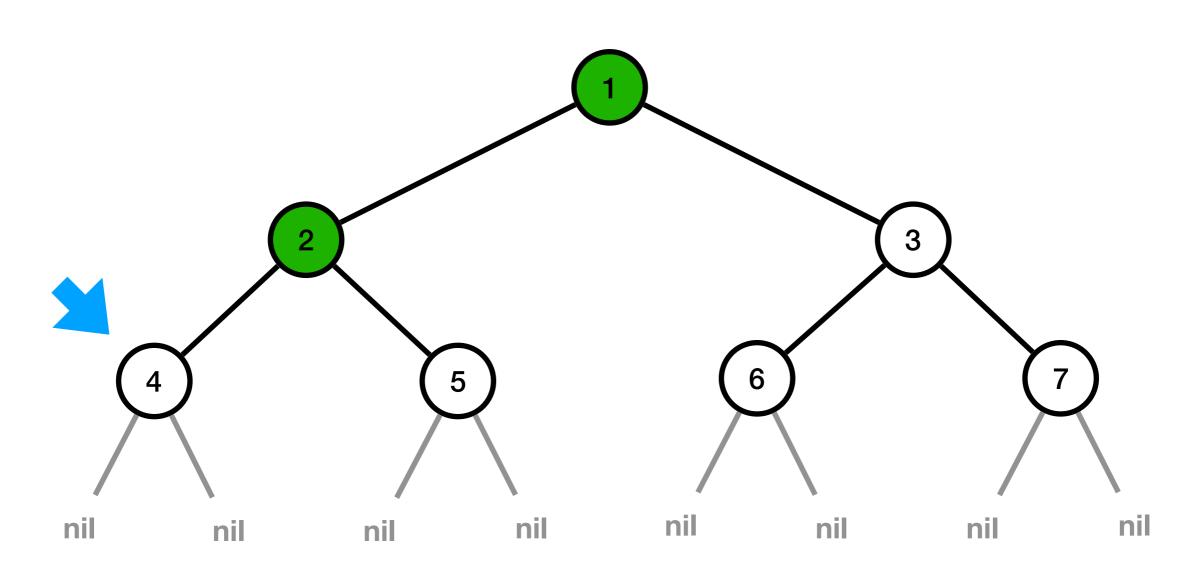




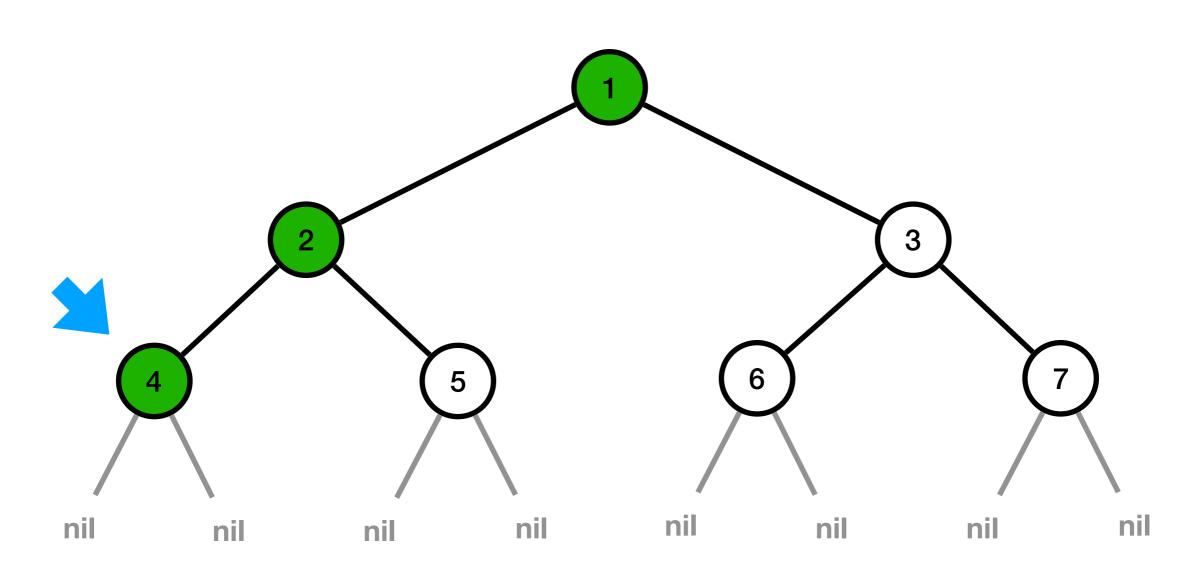








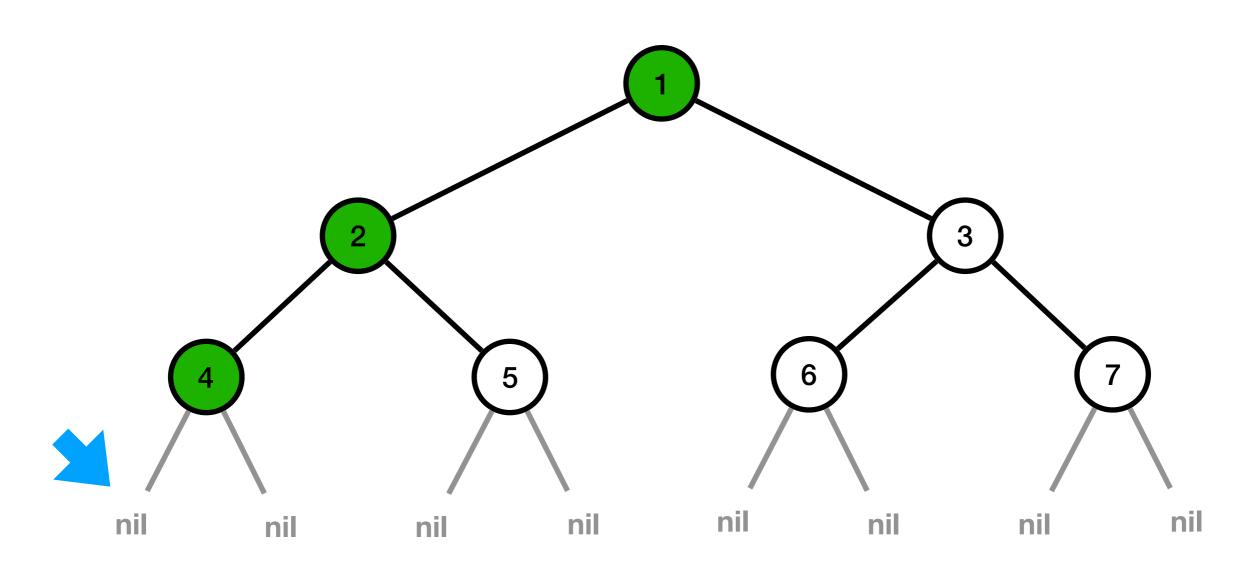


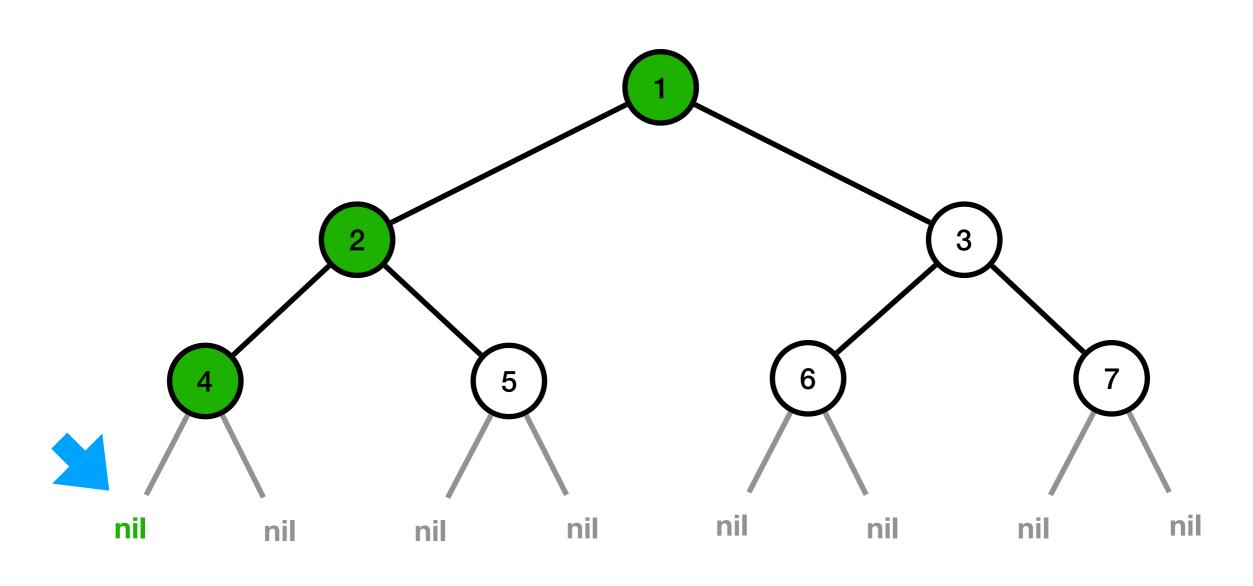


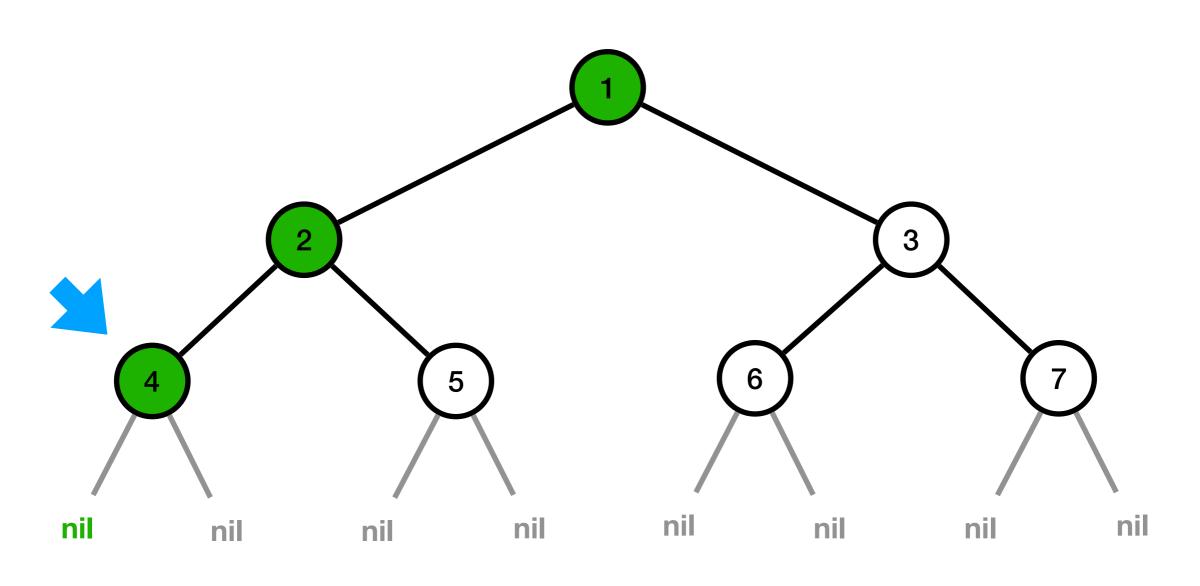








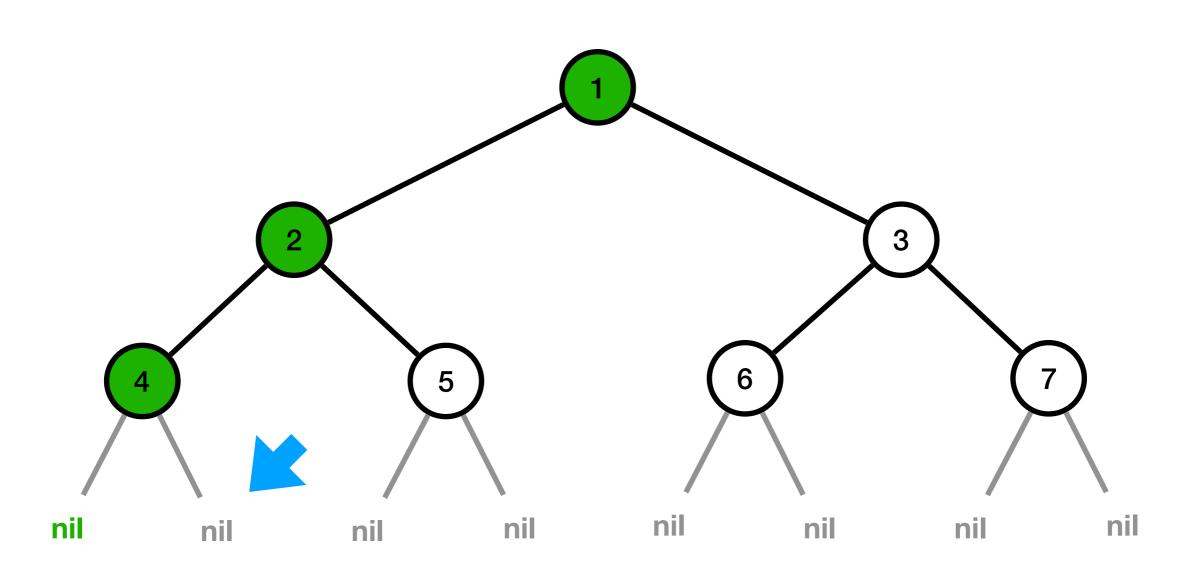




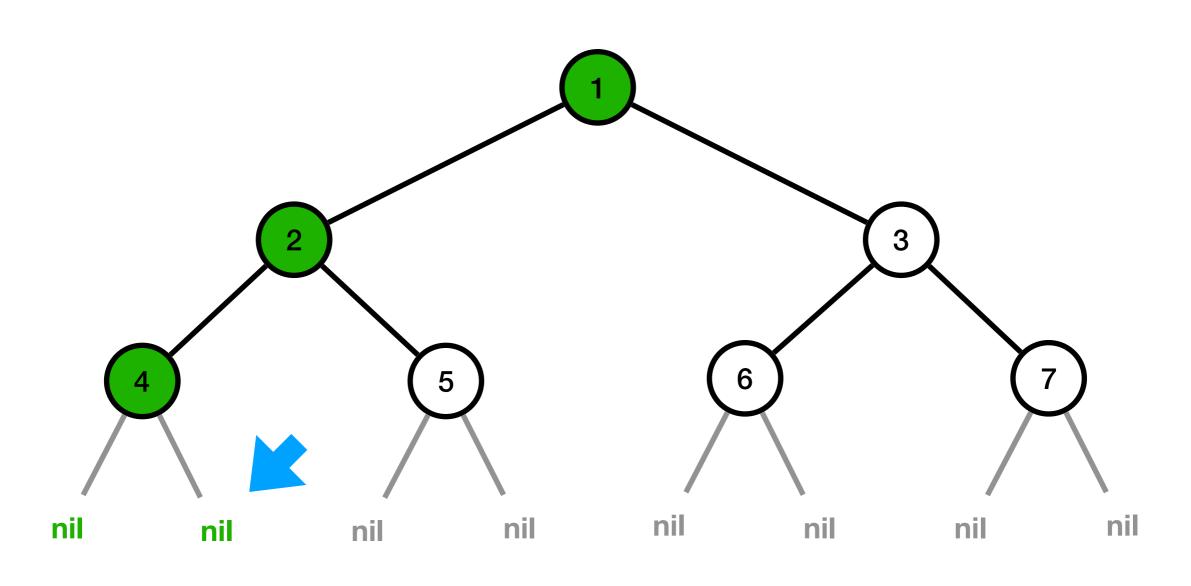






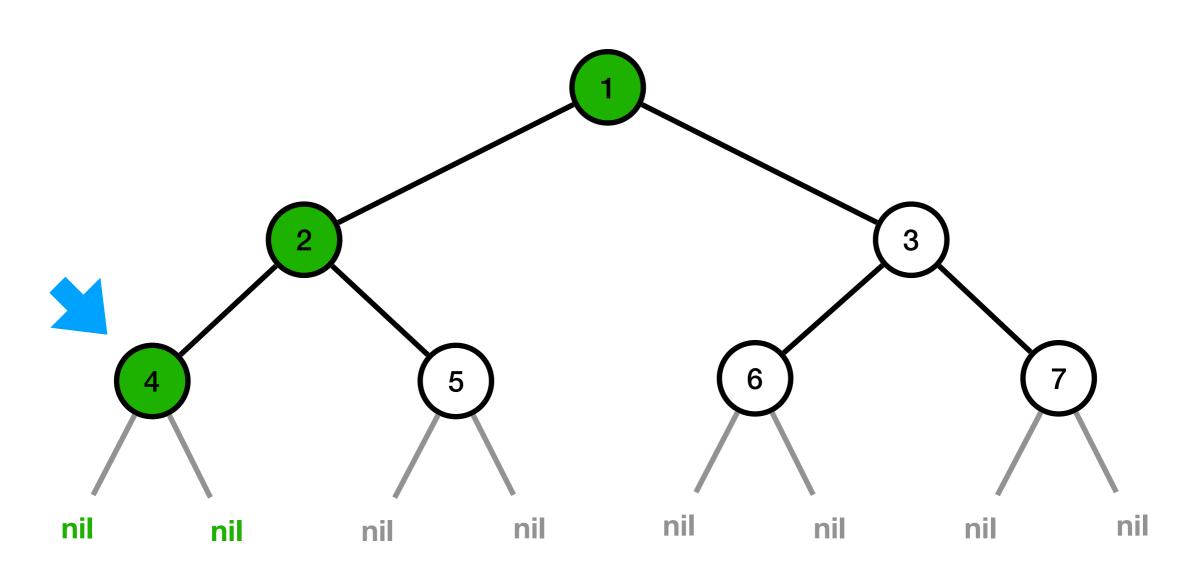






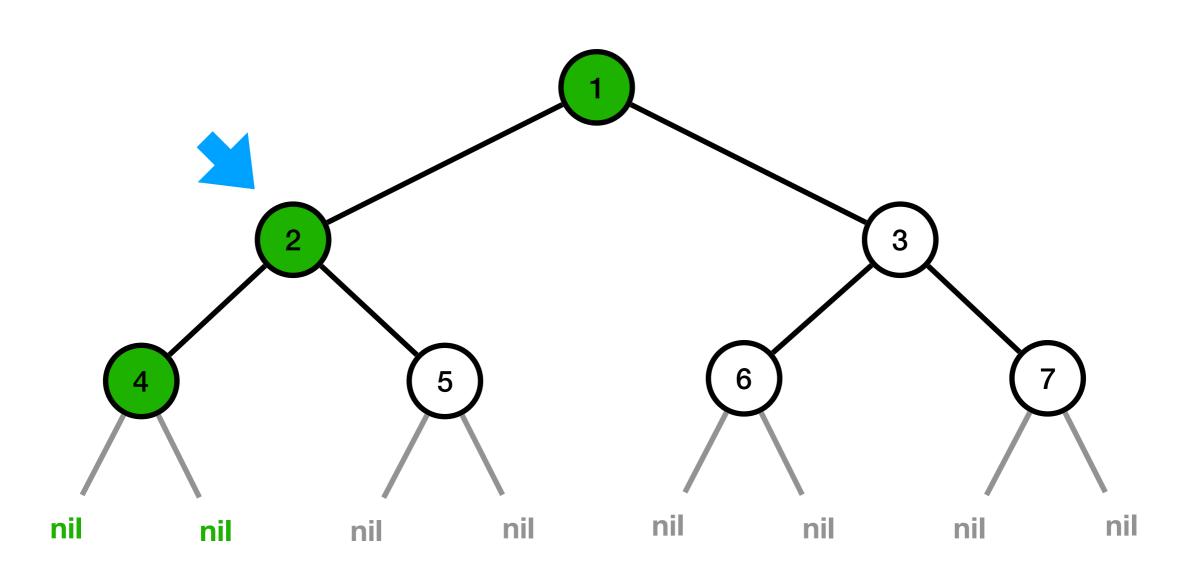




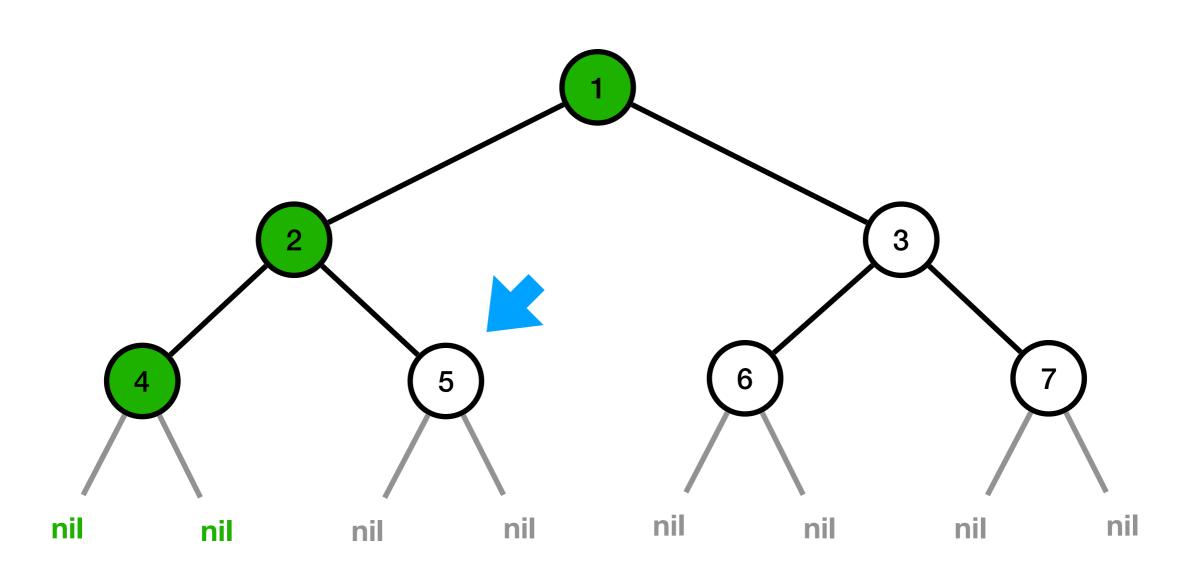




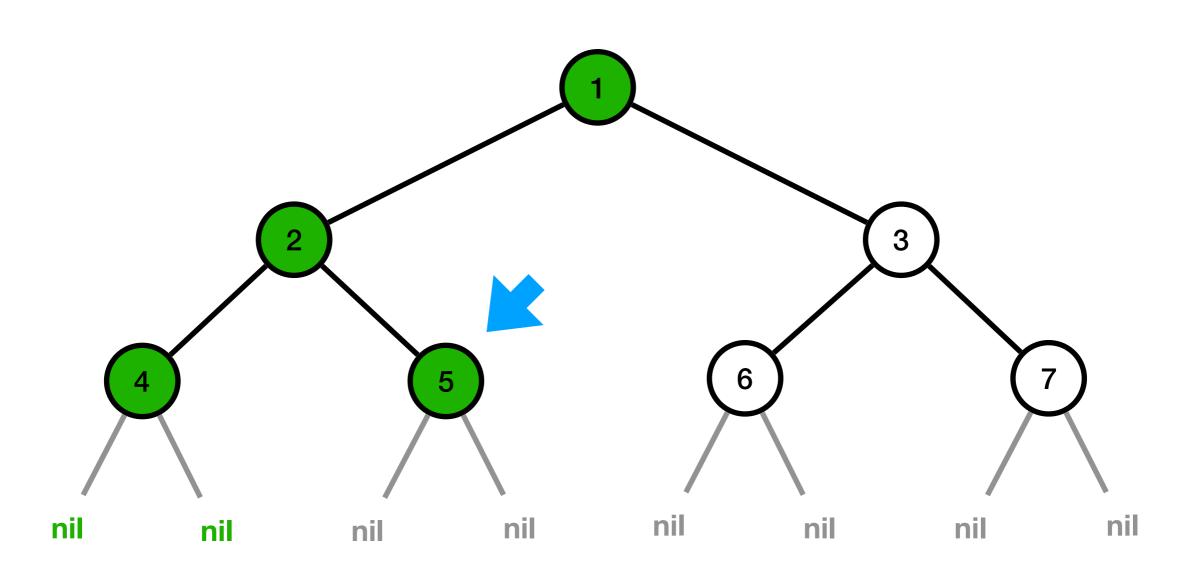


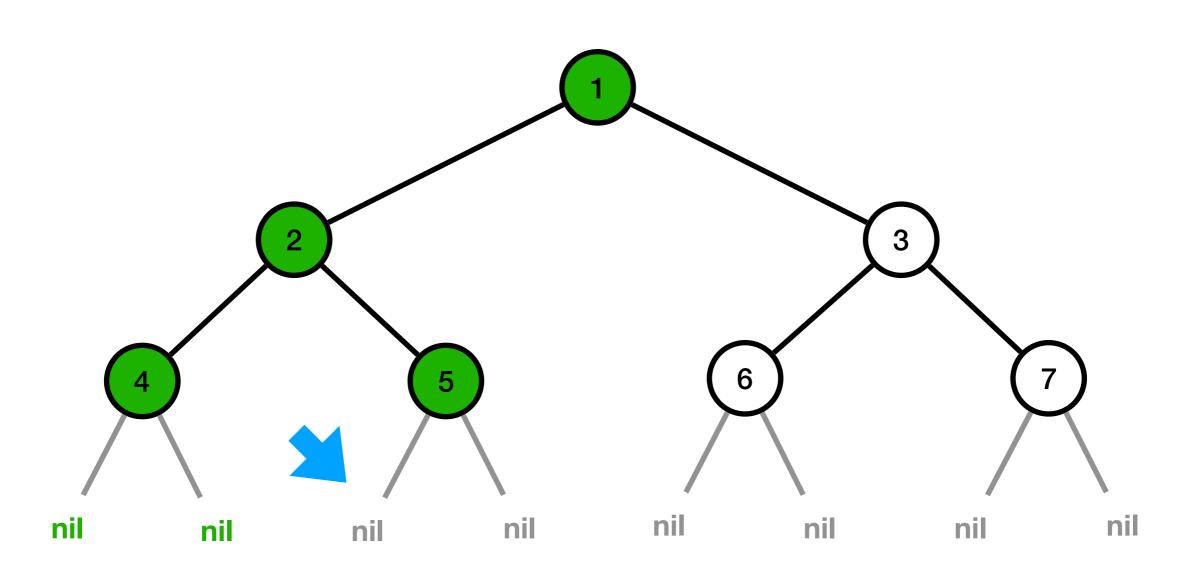


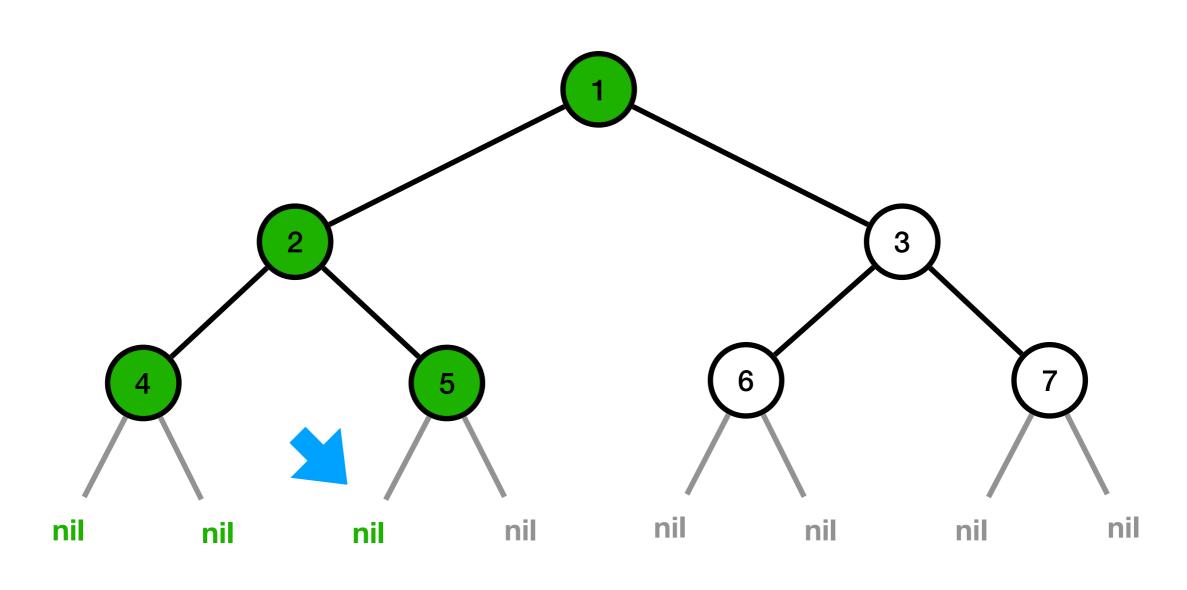


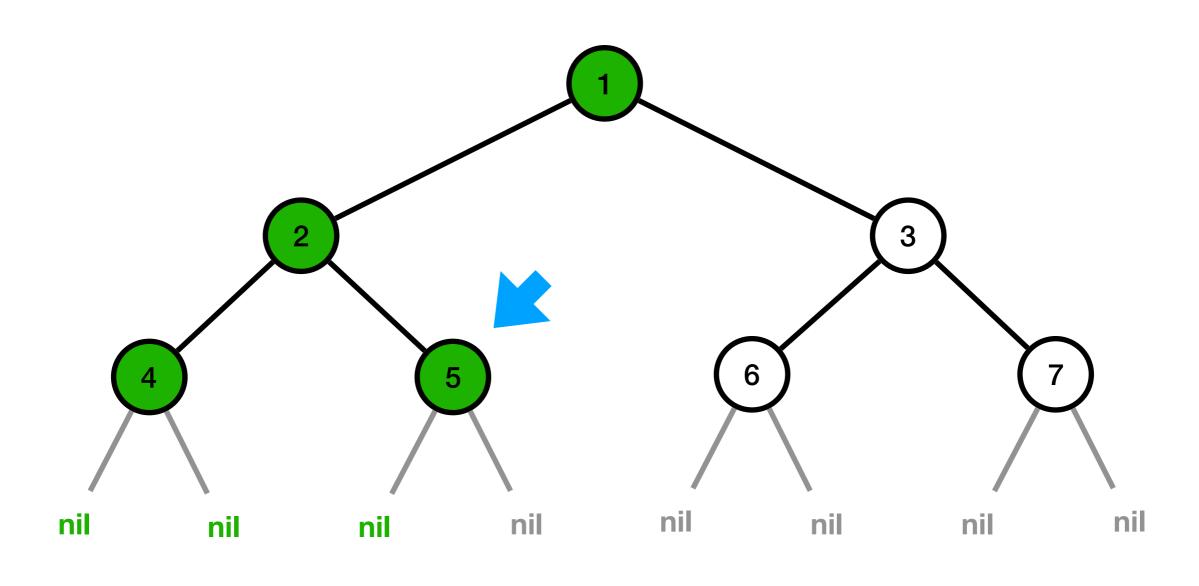


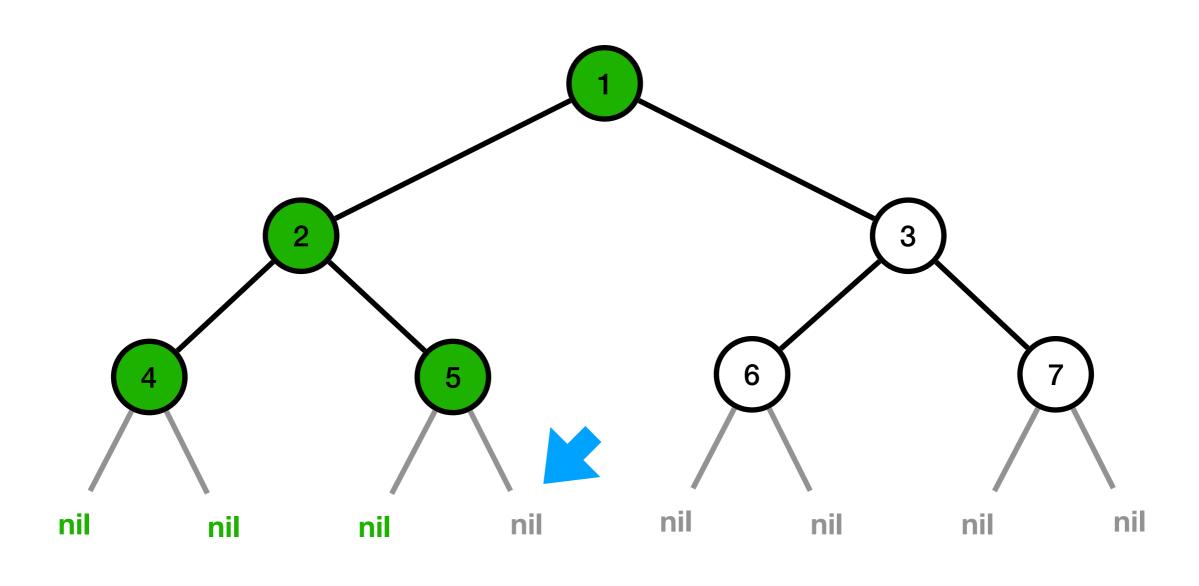


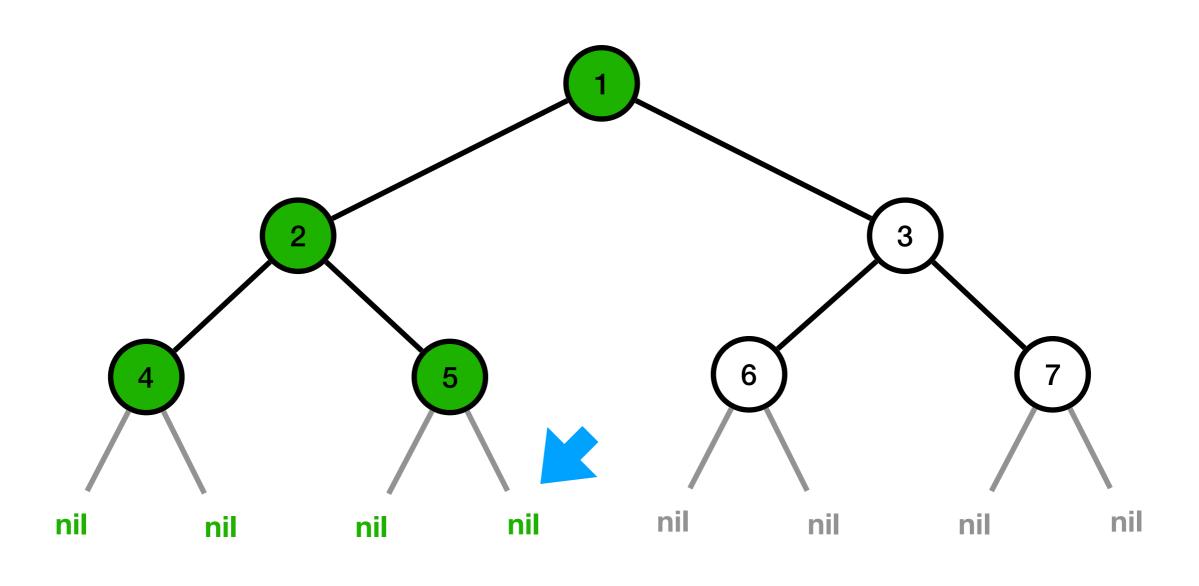


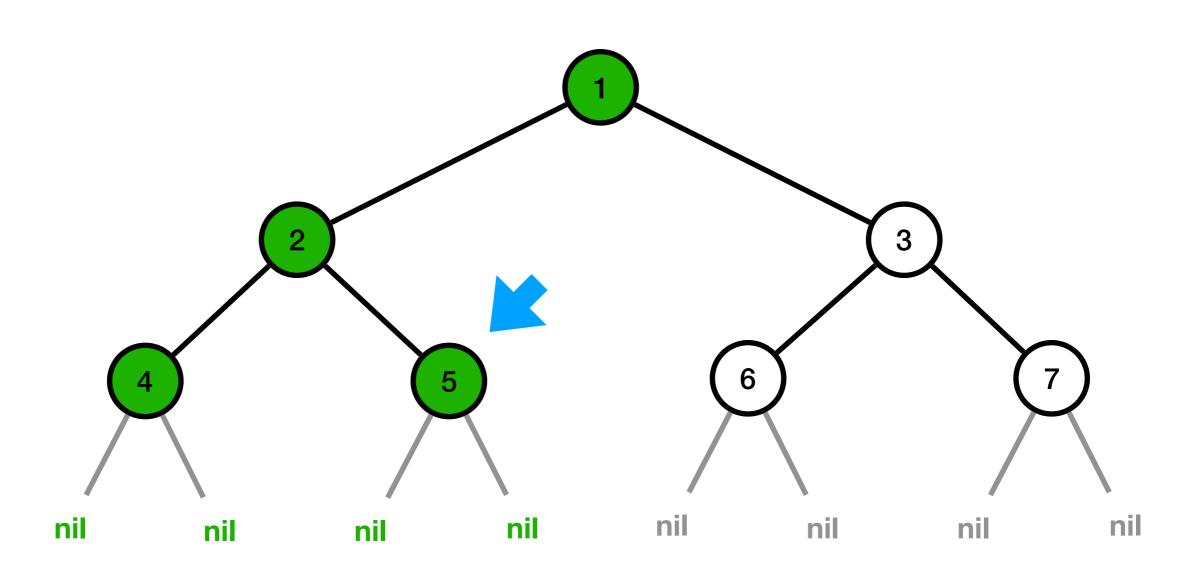


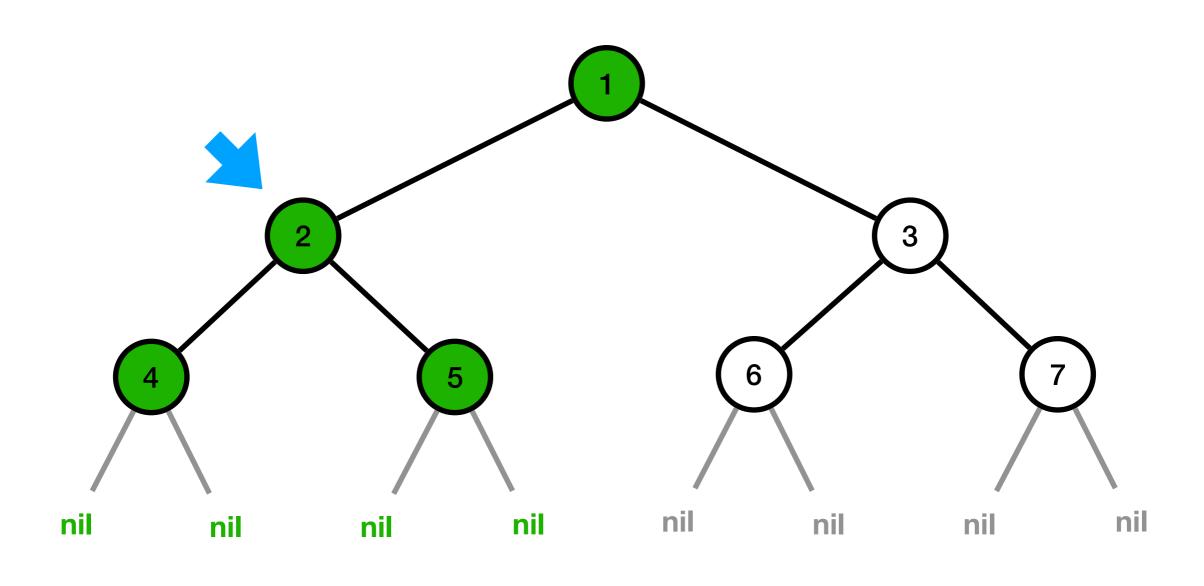


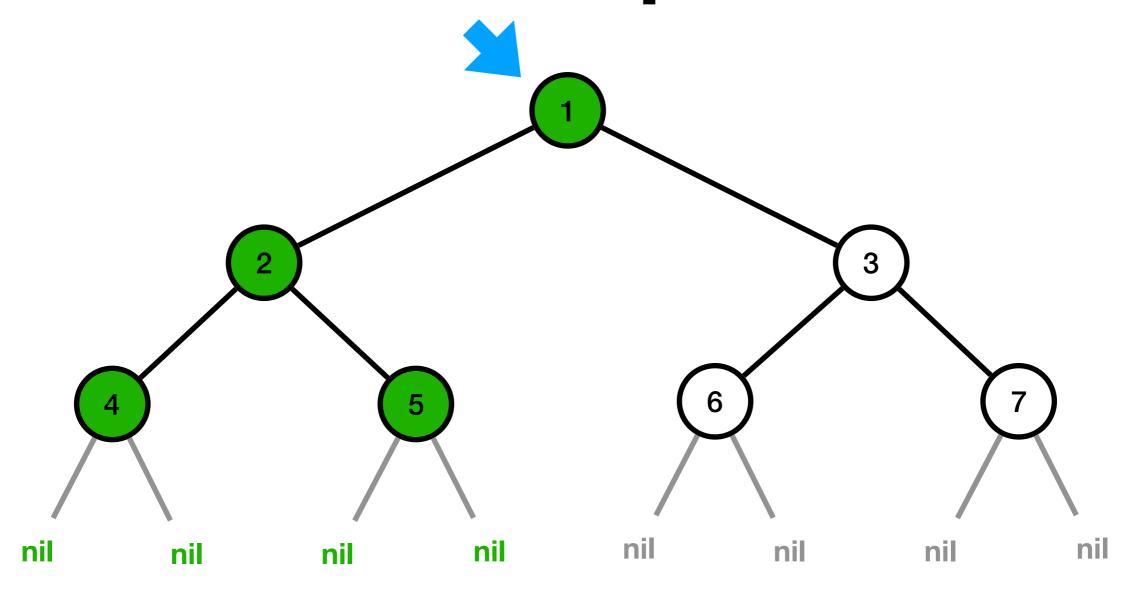


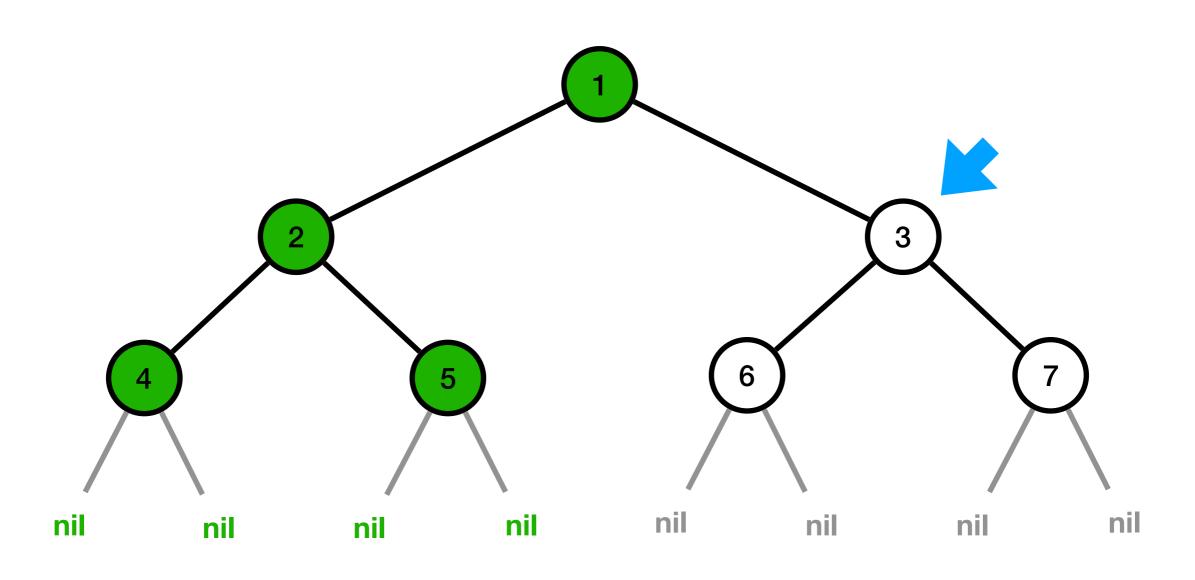


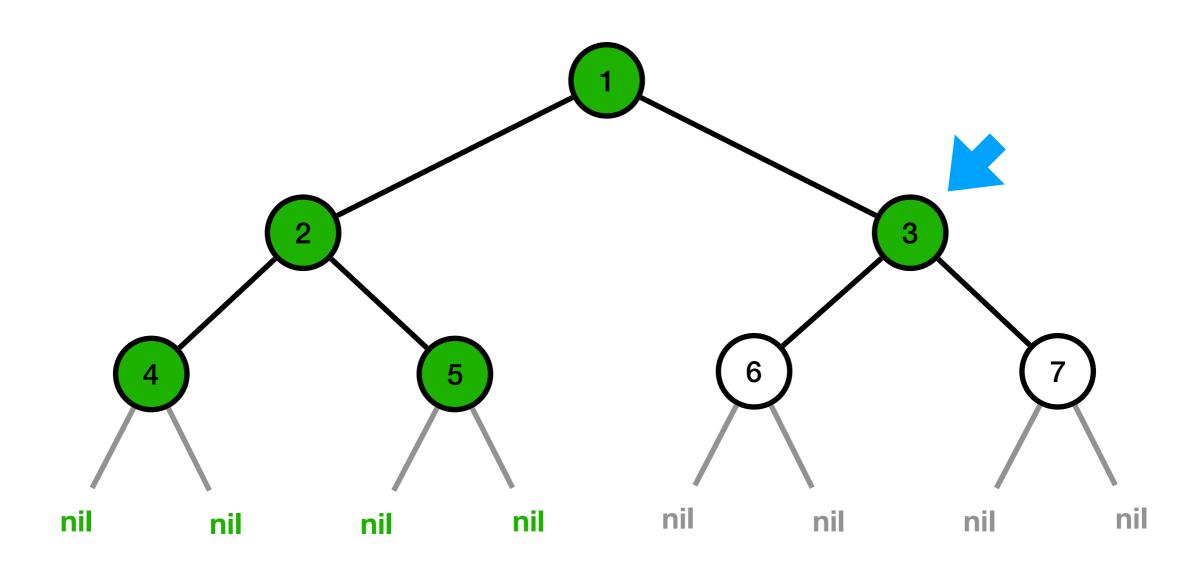


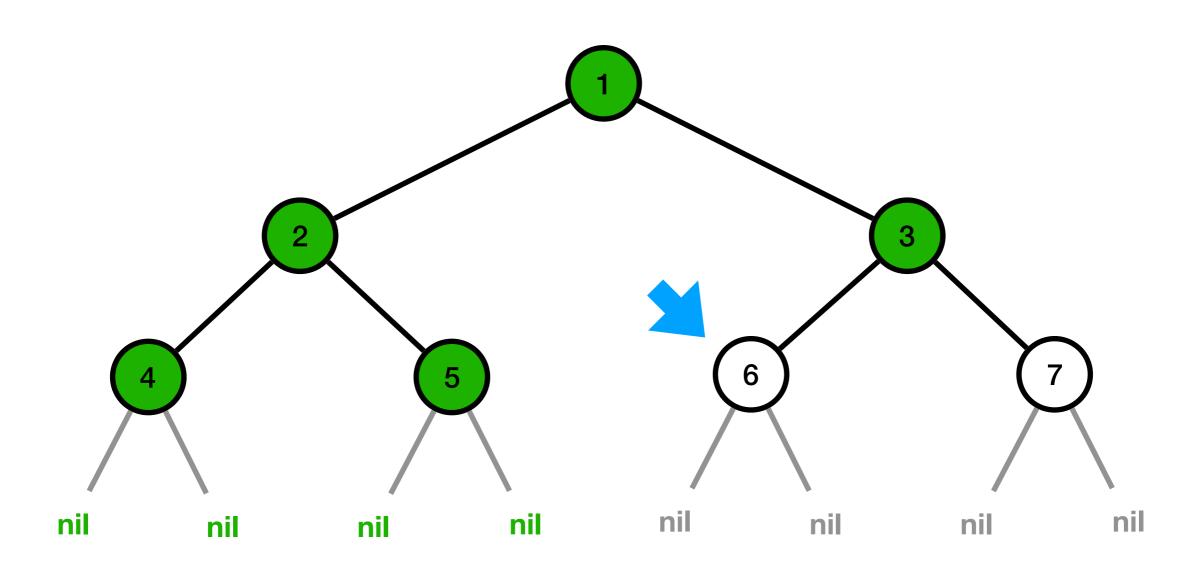


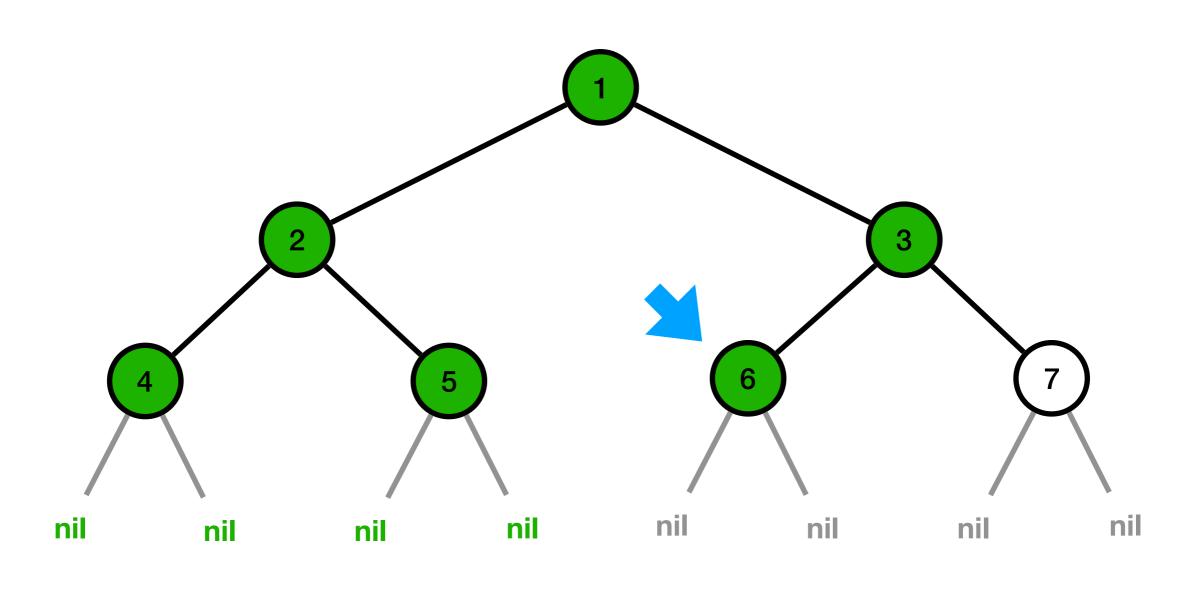


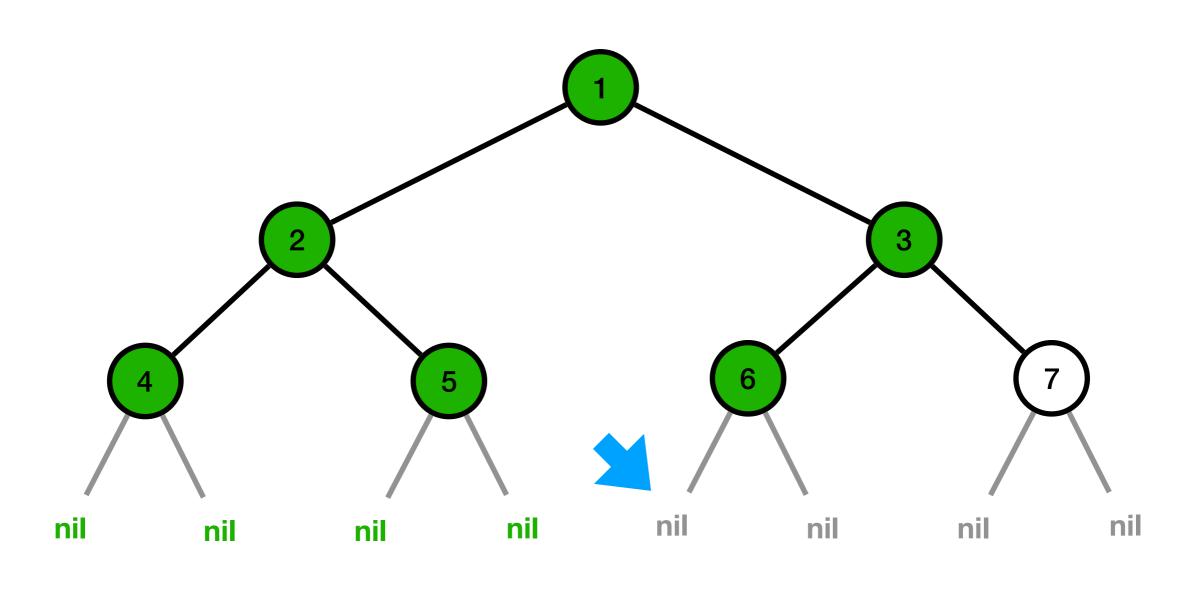


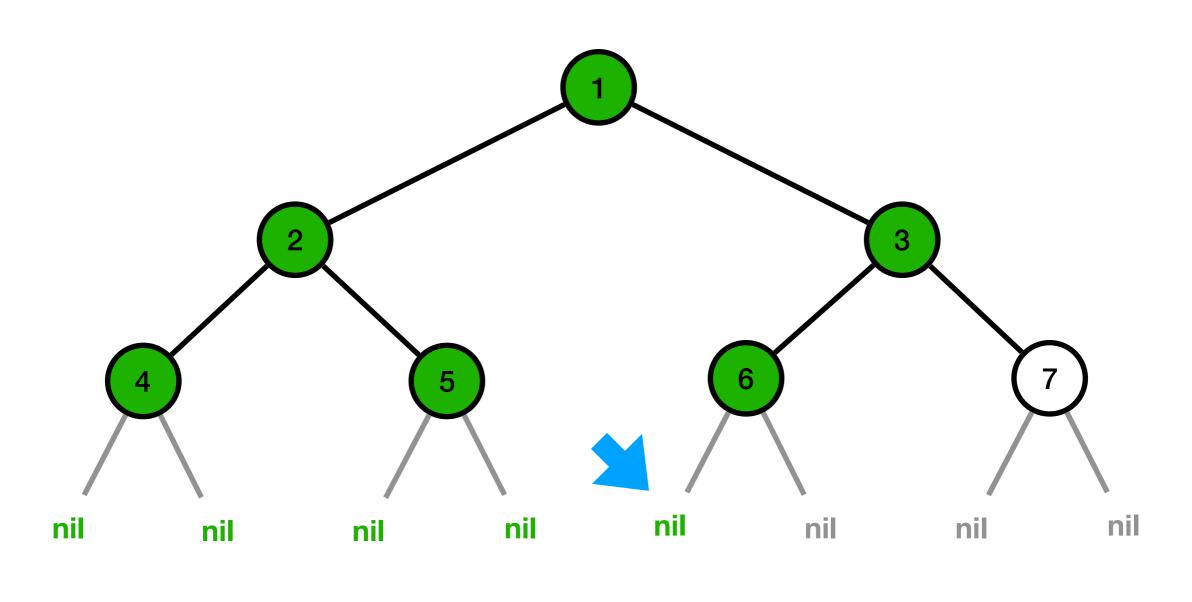


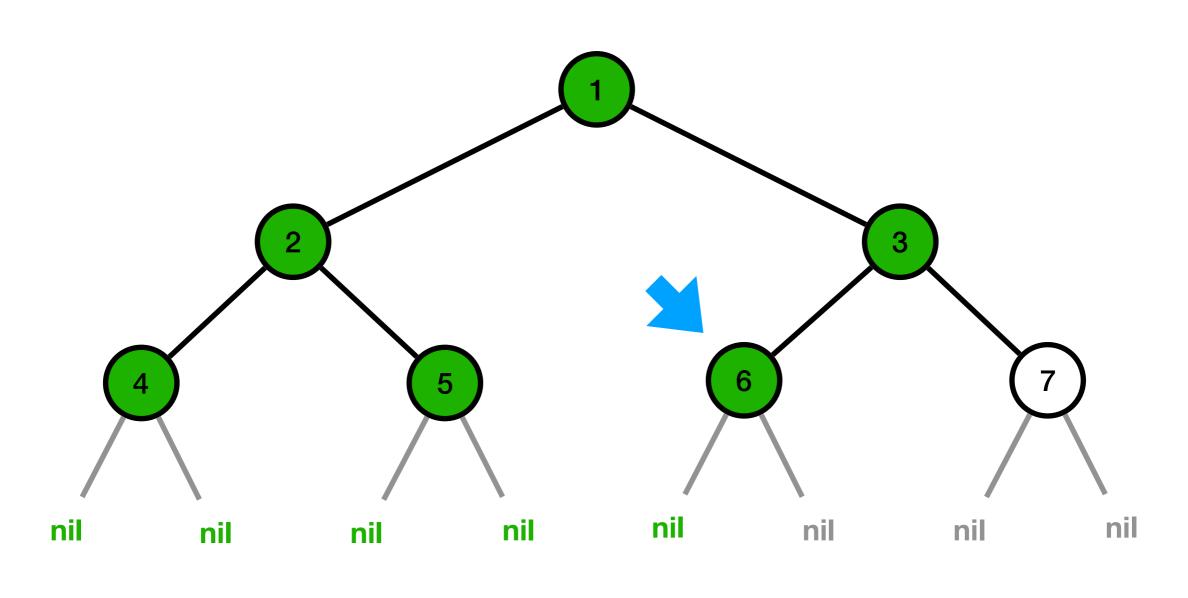


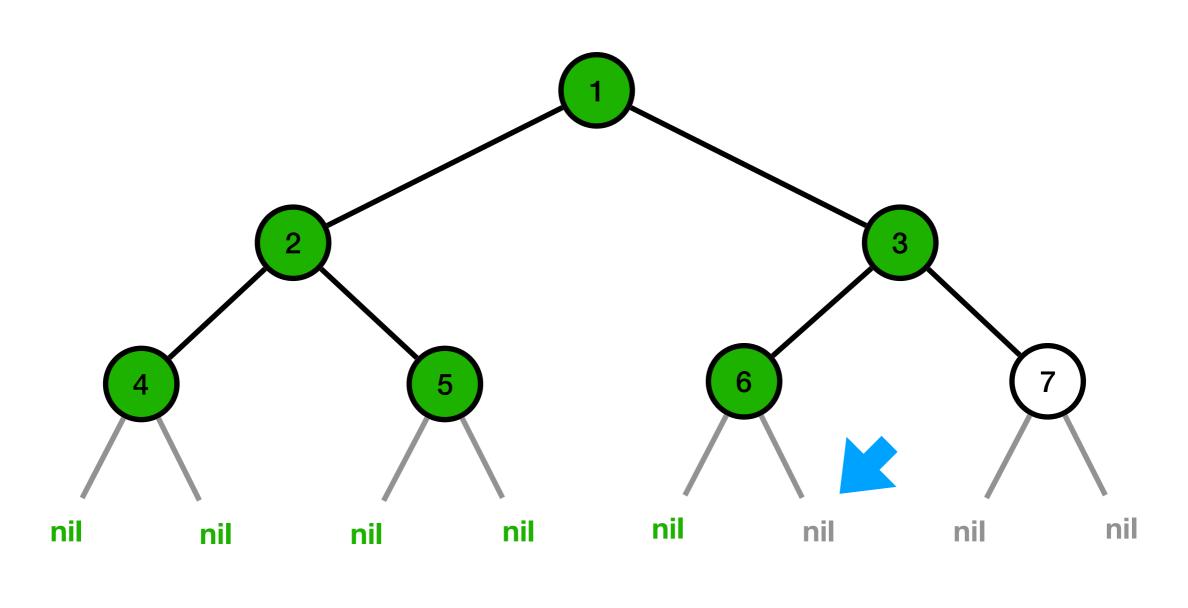


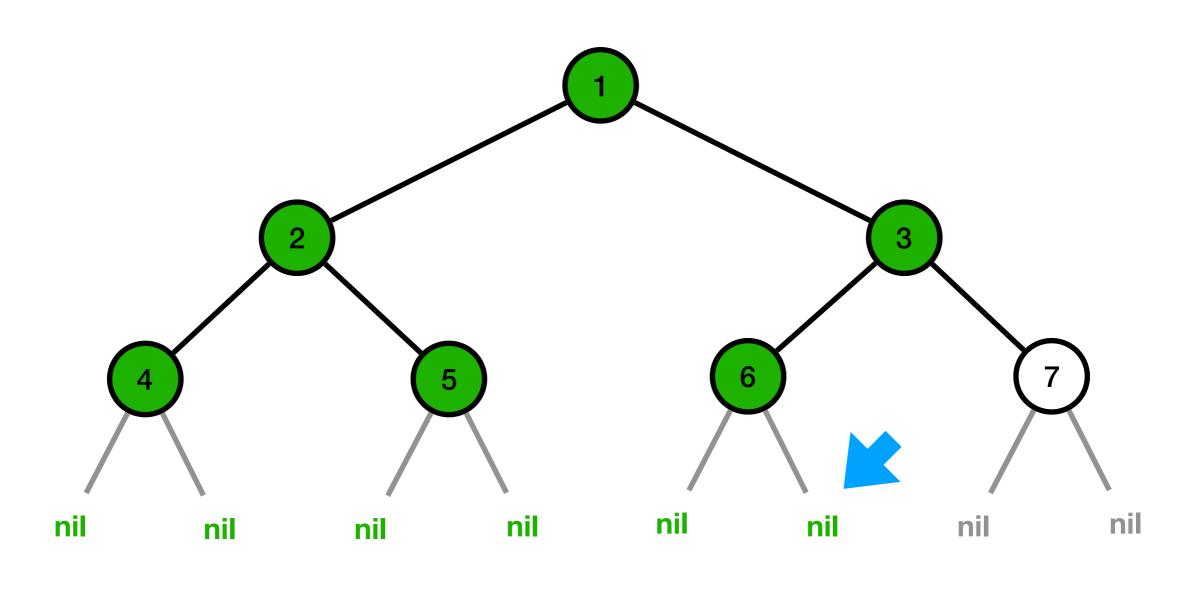


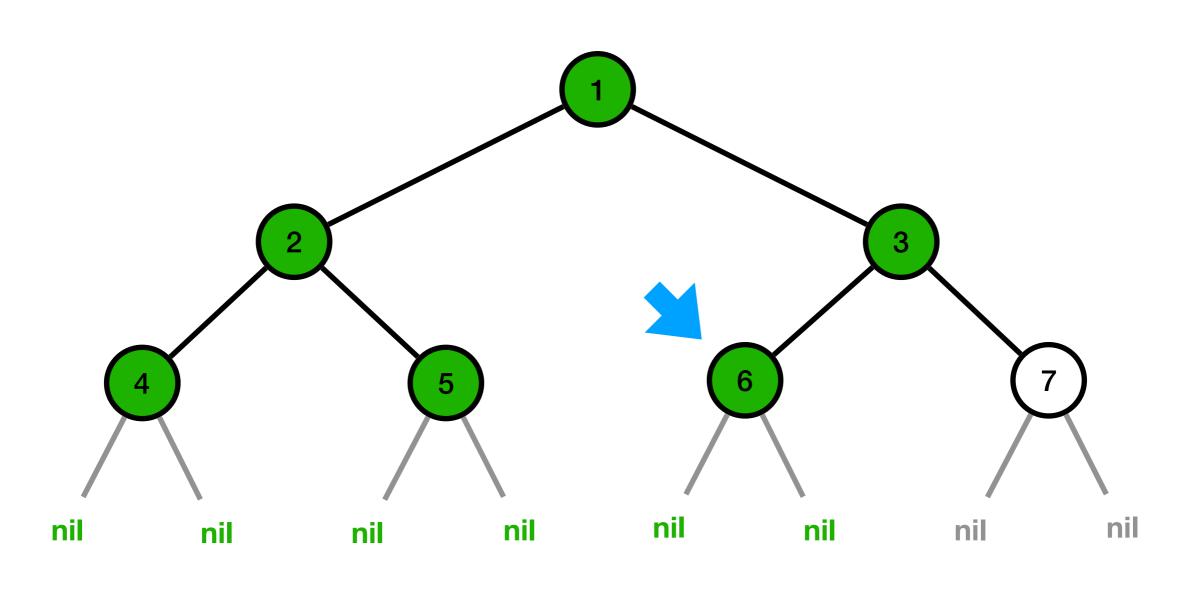


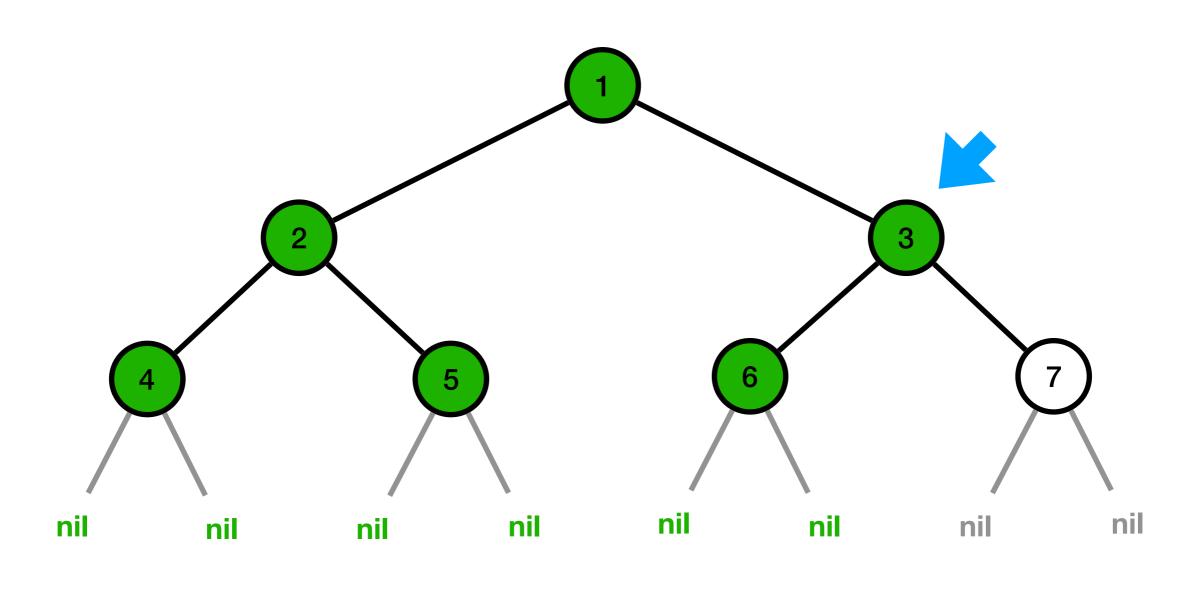


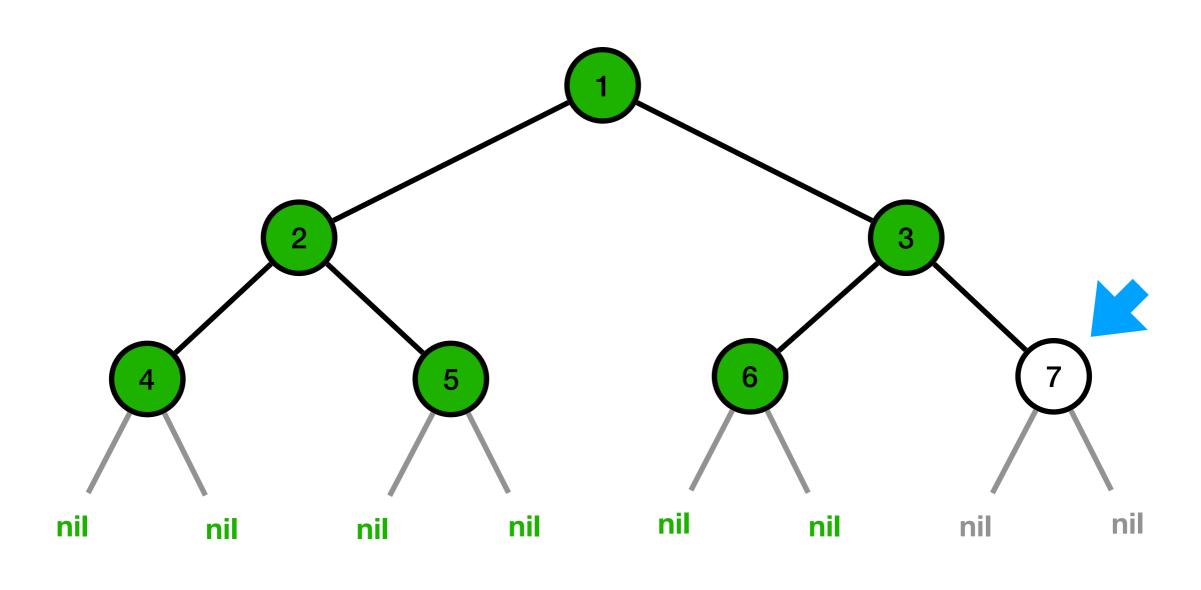


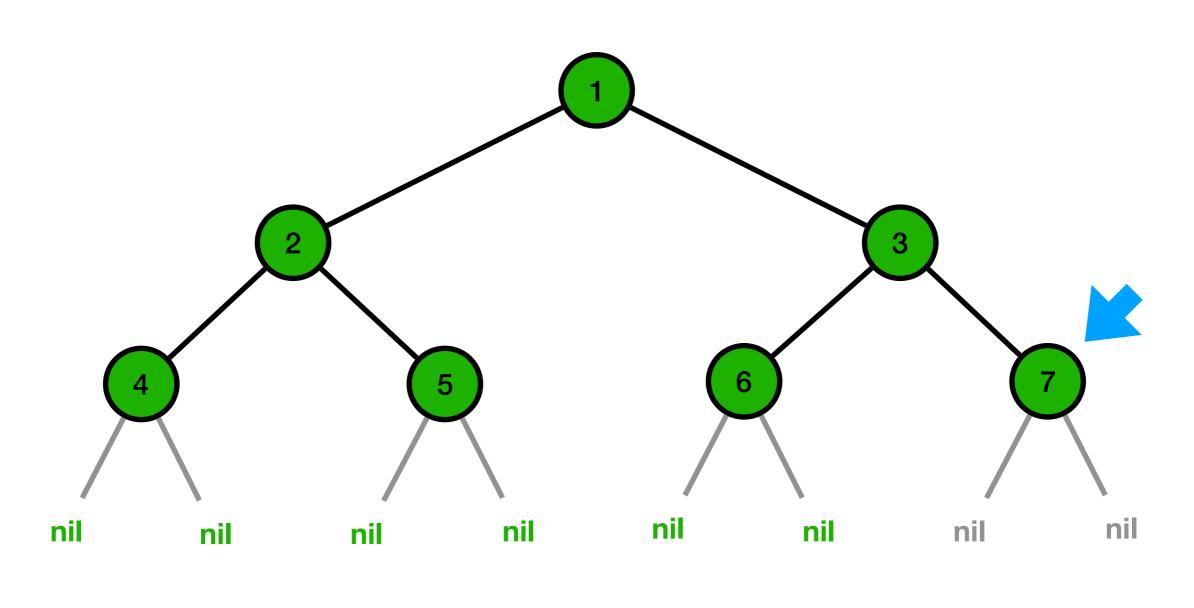


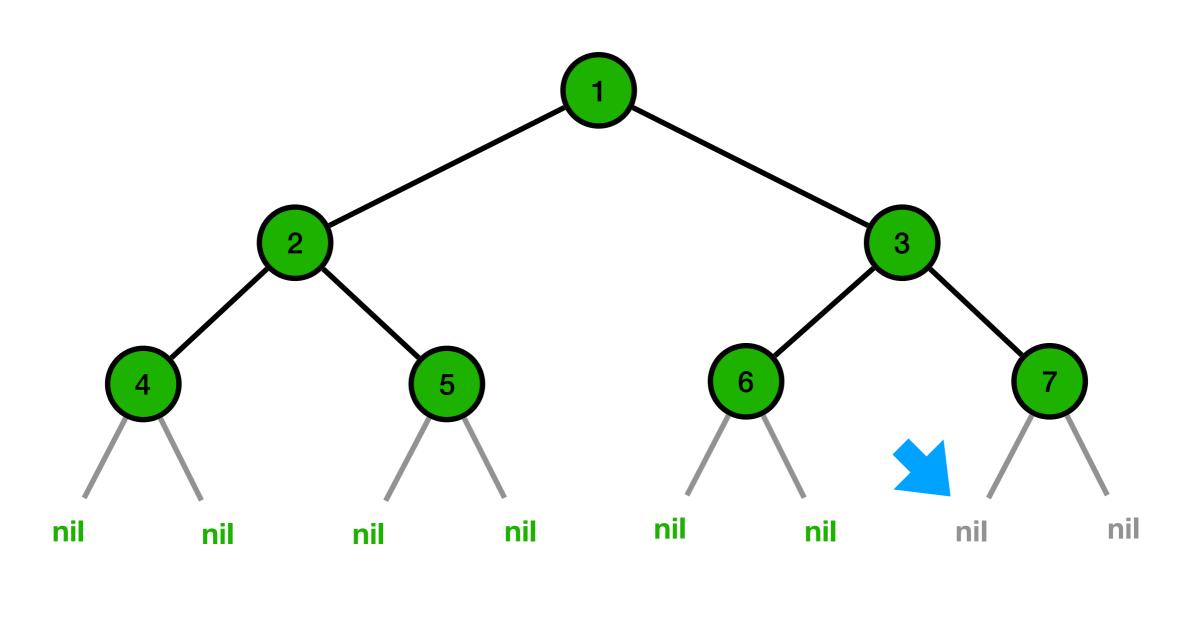


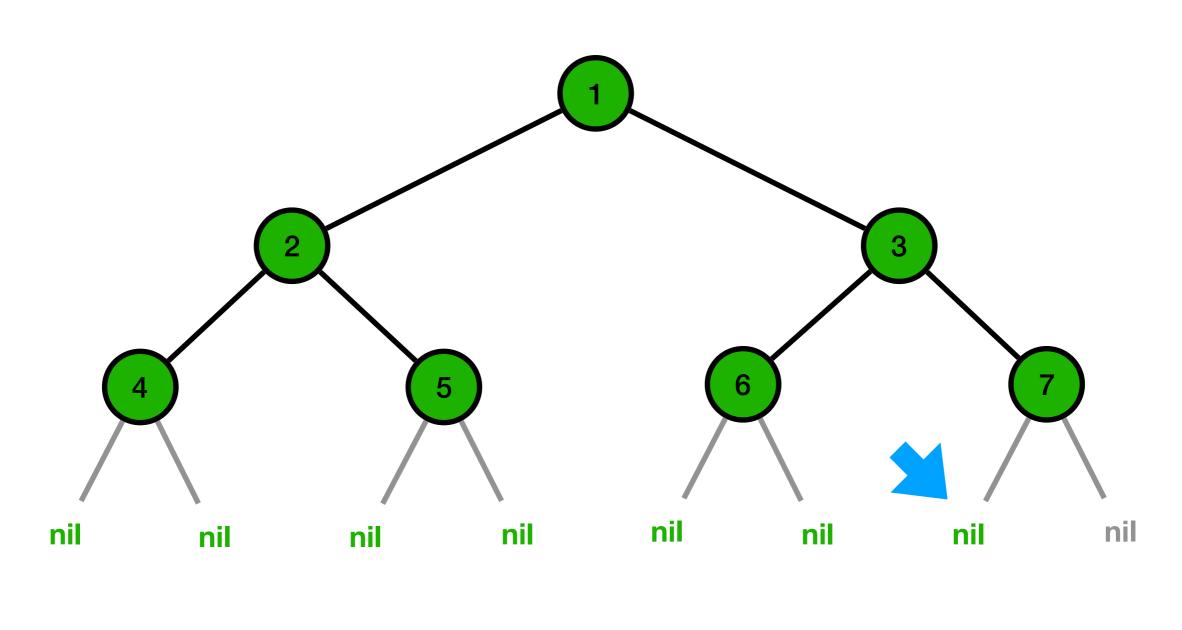


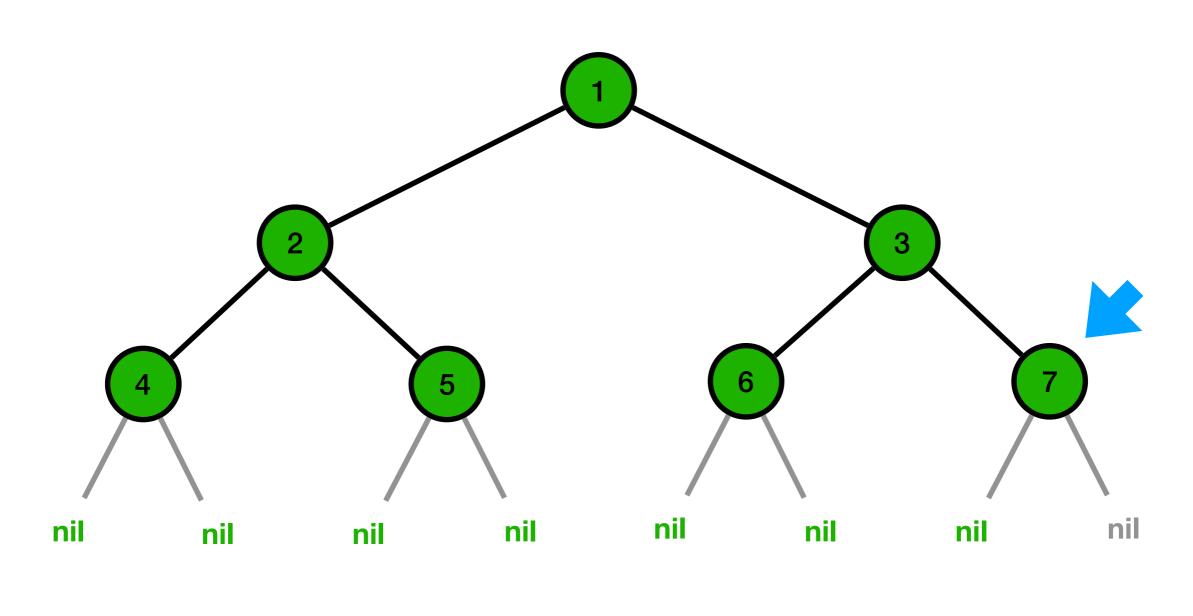


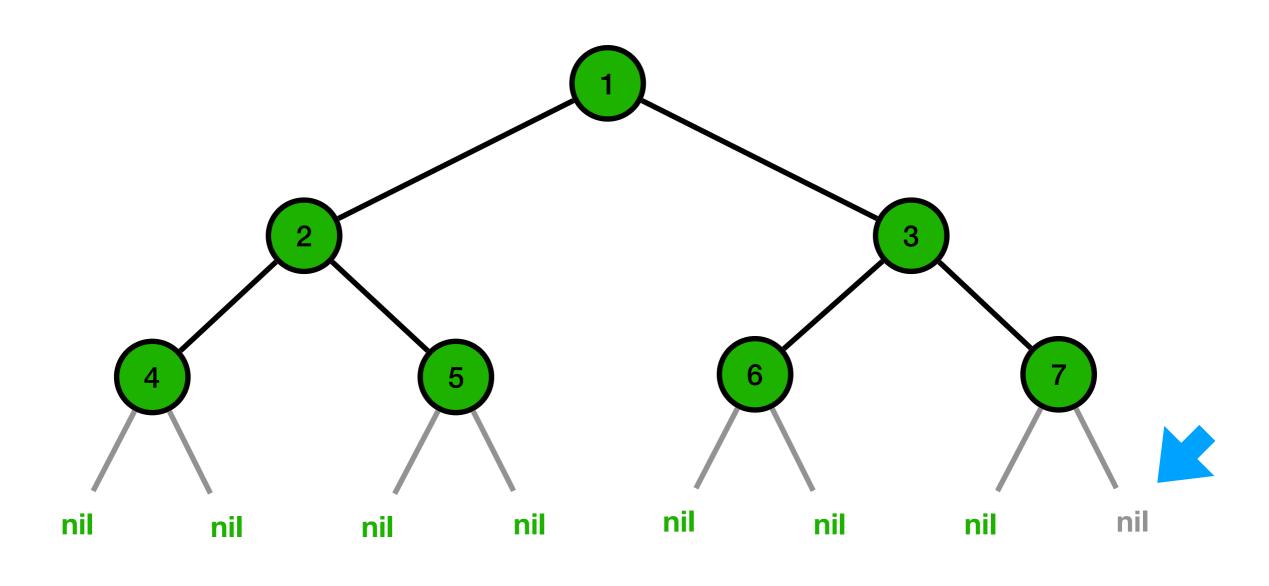


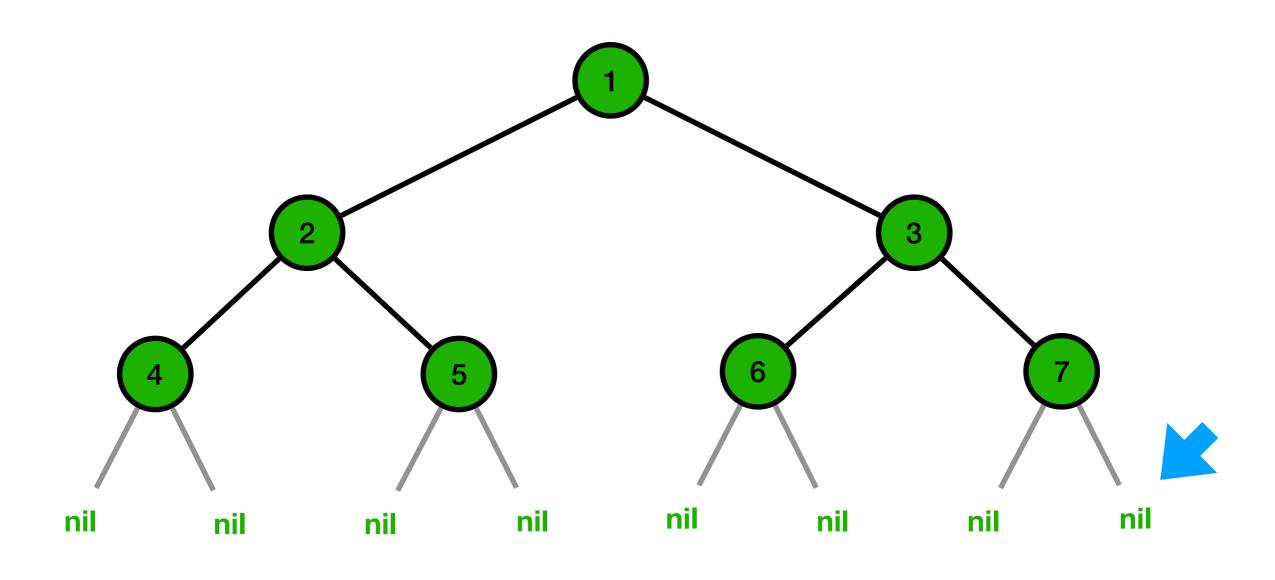


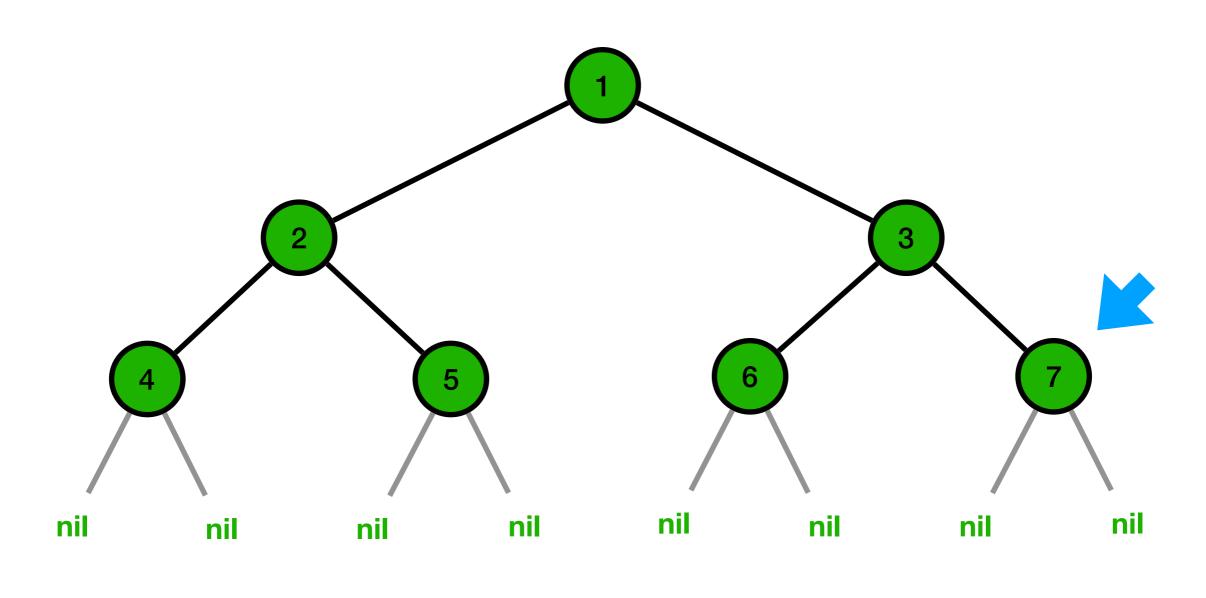


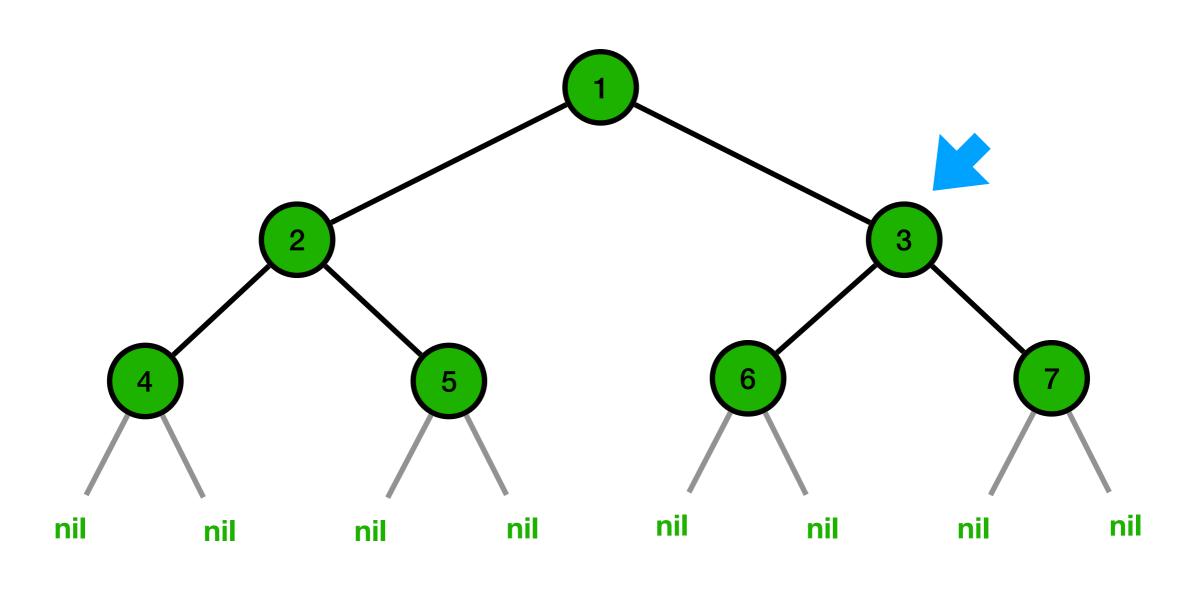


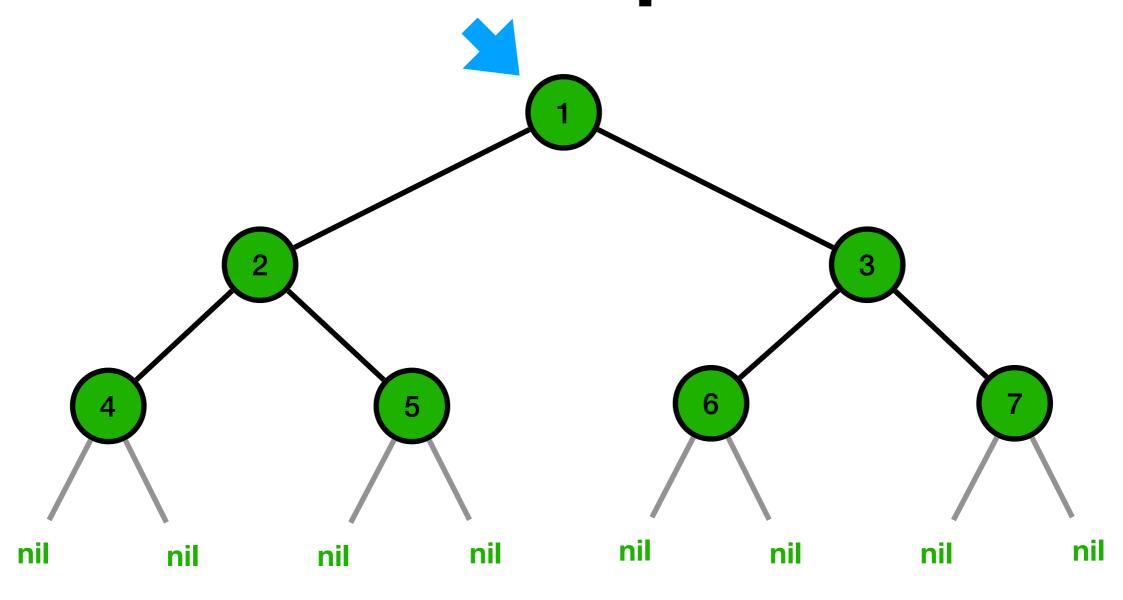


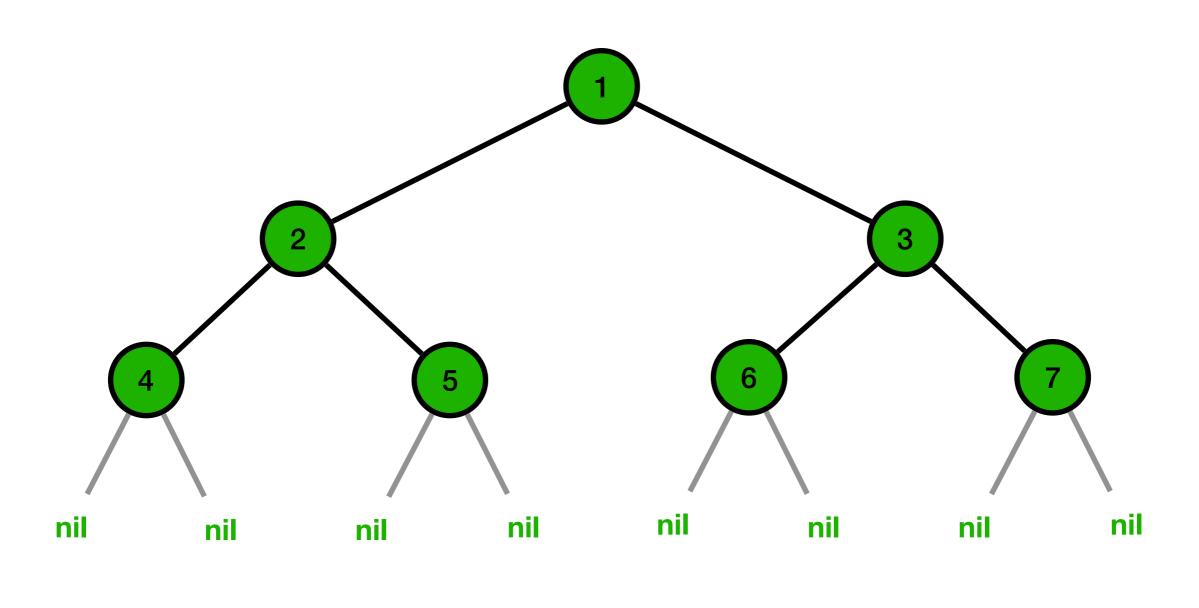






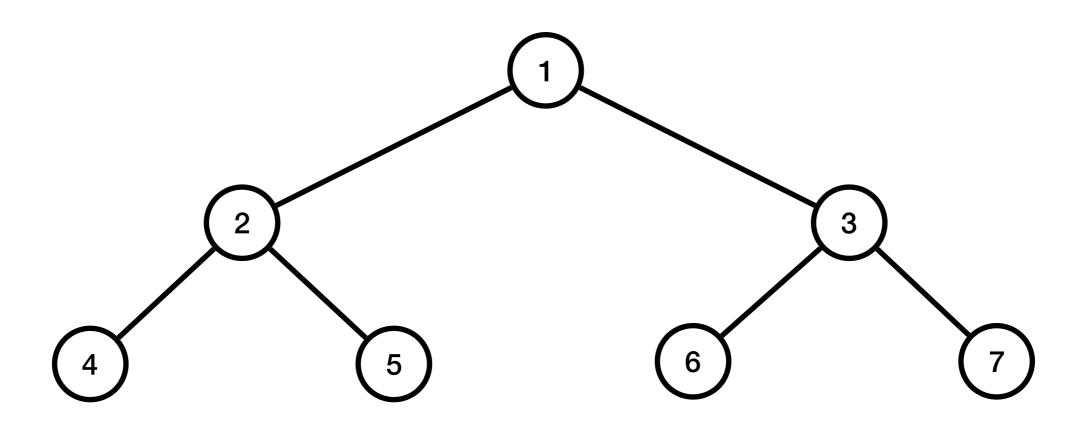


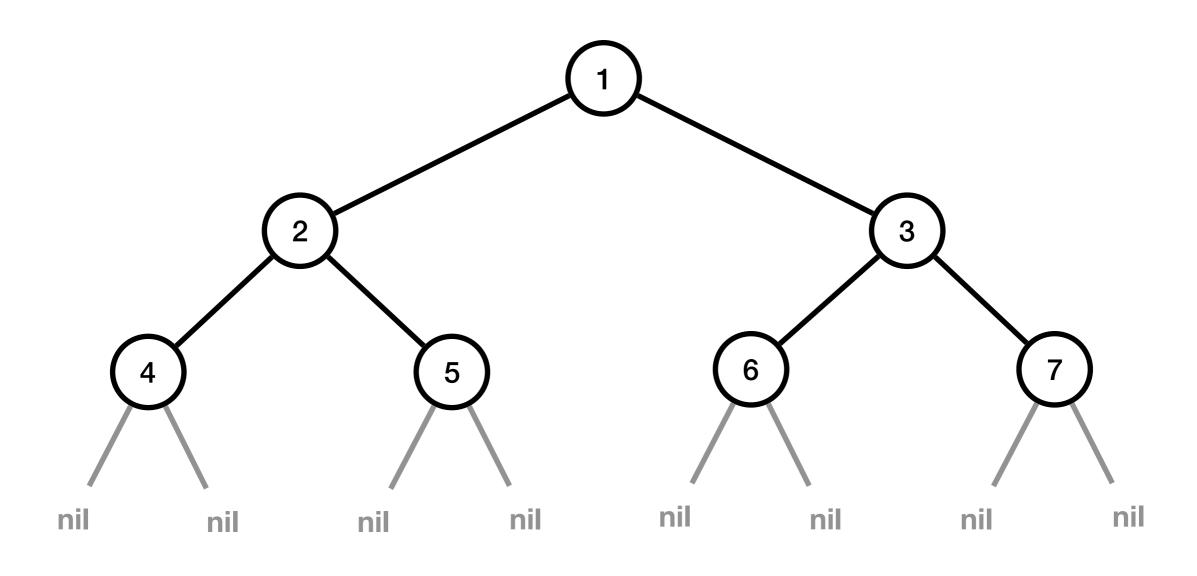


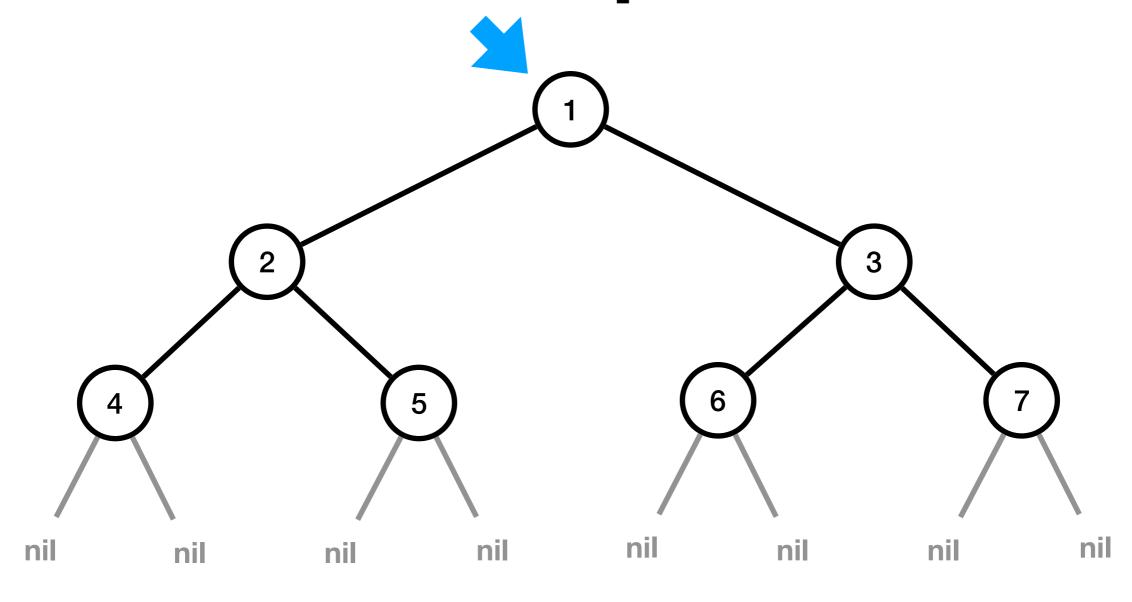


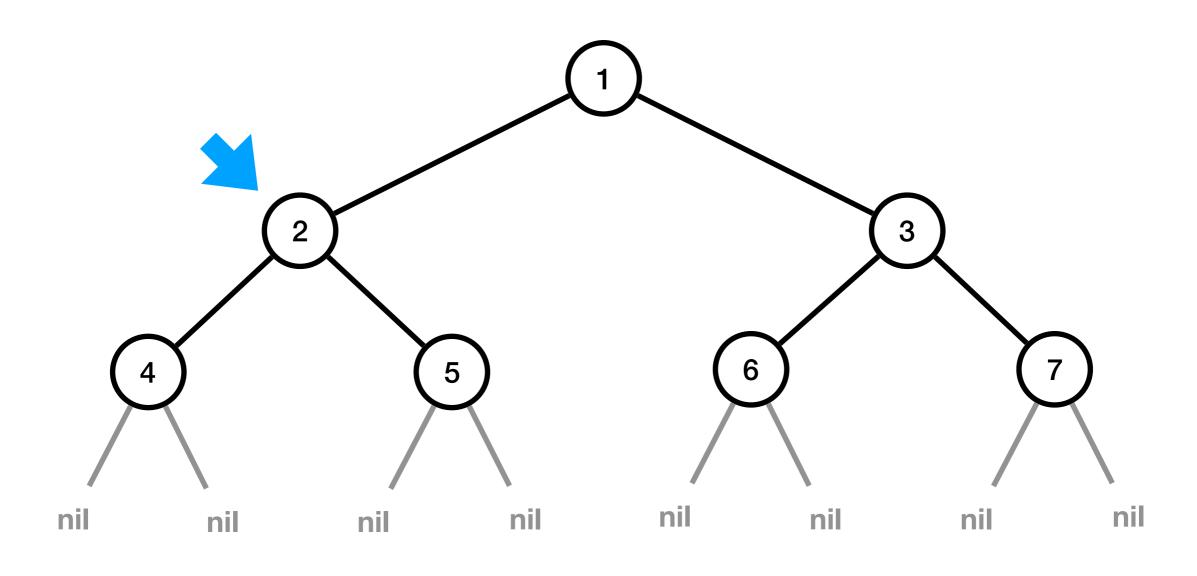
Parcours postfixe (ou suffixe)

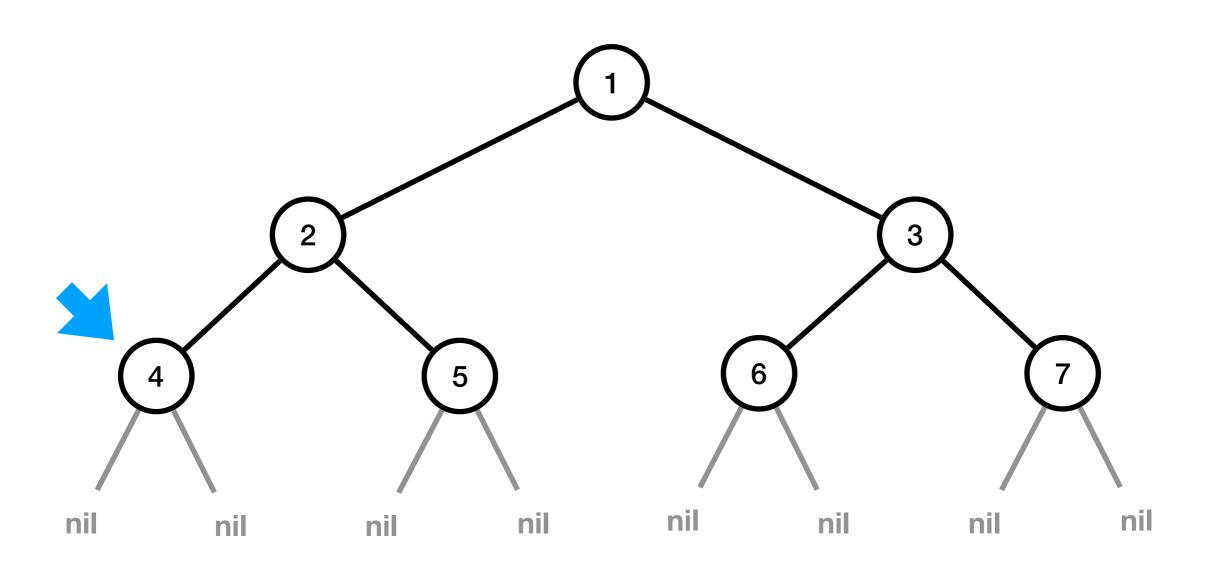
```
procédure visiter-postfixe(nœud)
    si nœud ≠ nil alors
    visiter-postfixe(gauche[nœud])
    visiter-postfixe(droite[nœud])
    afficher(valeur[nœud])
```

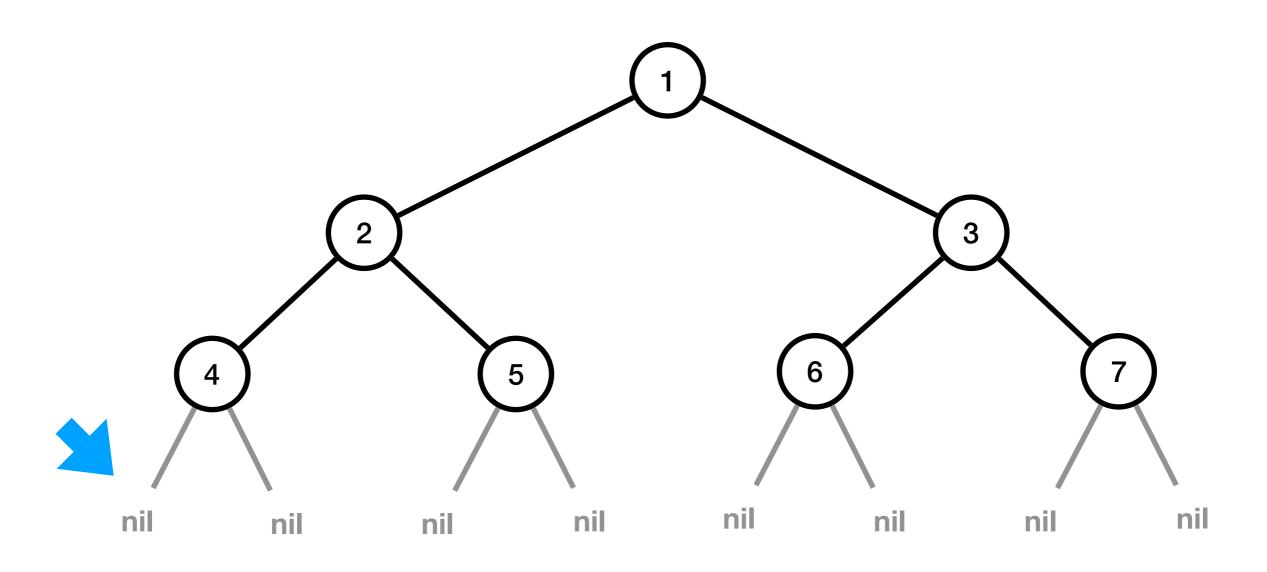


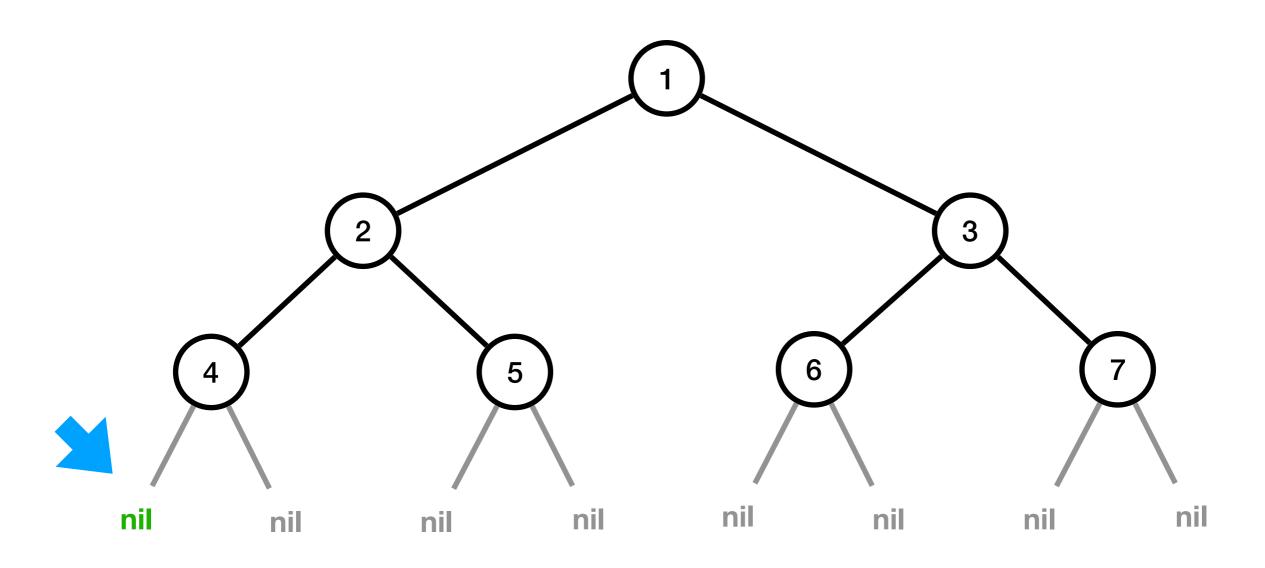


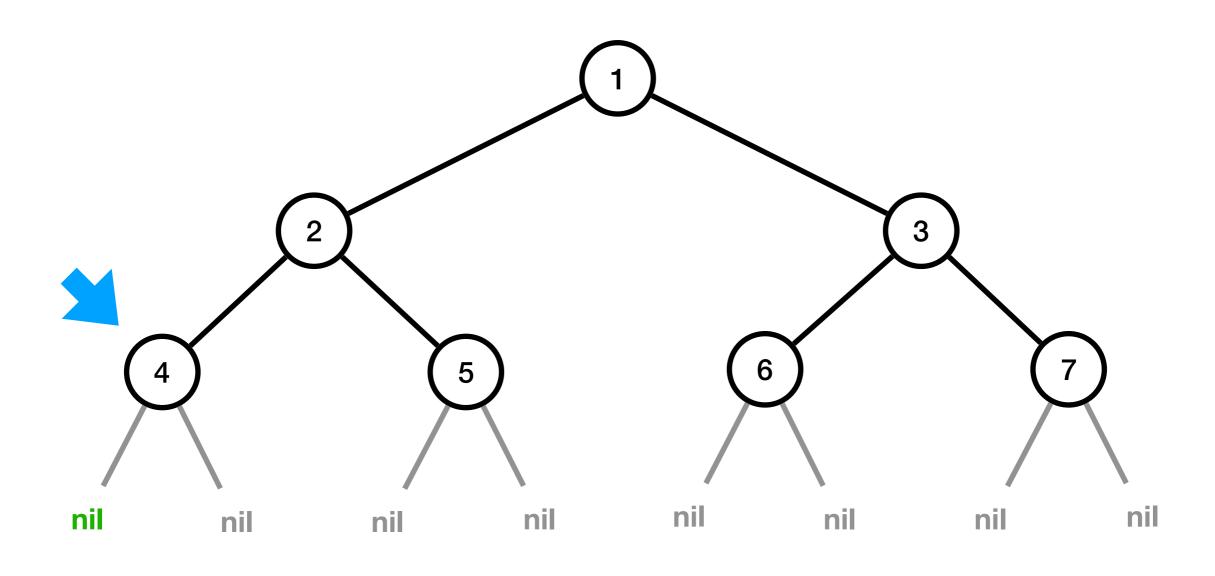


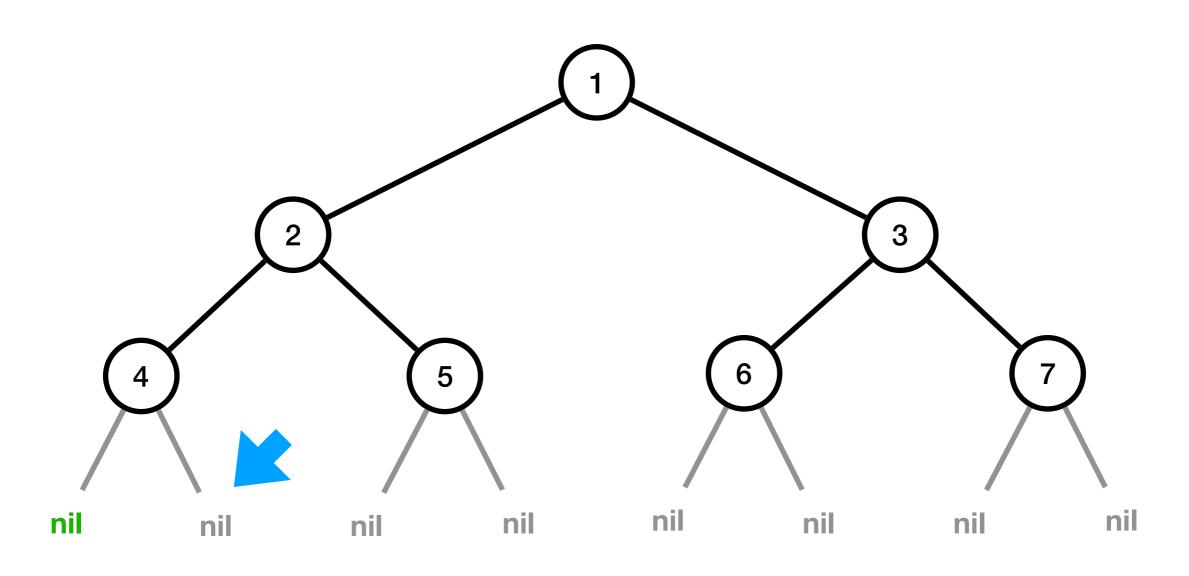


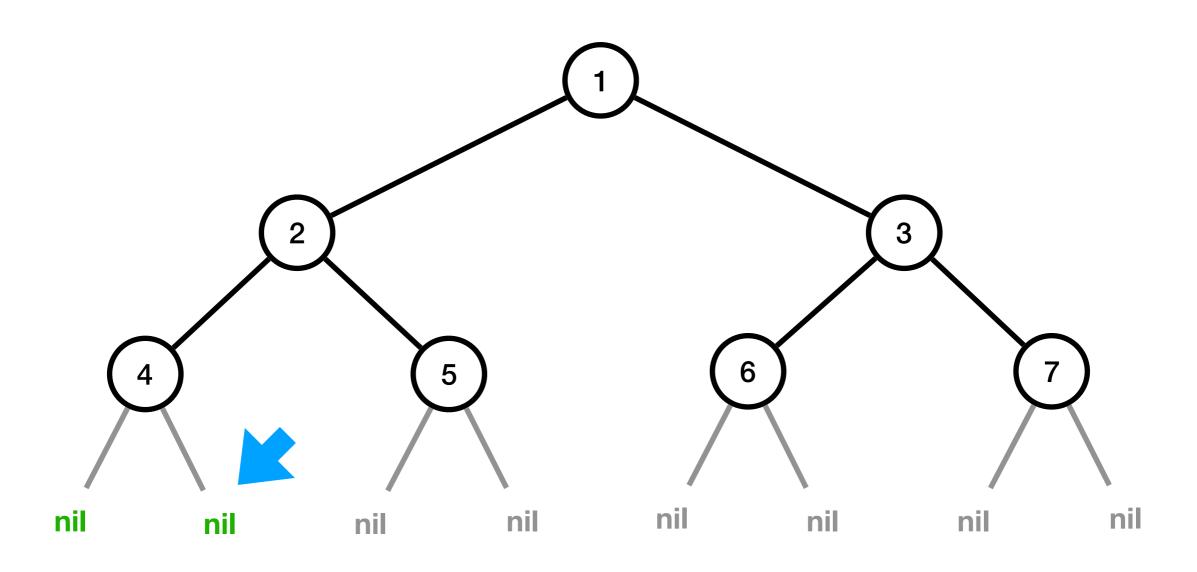


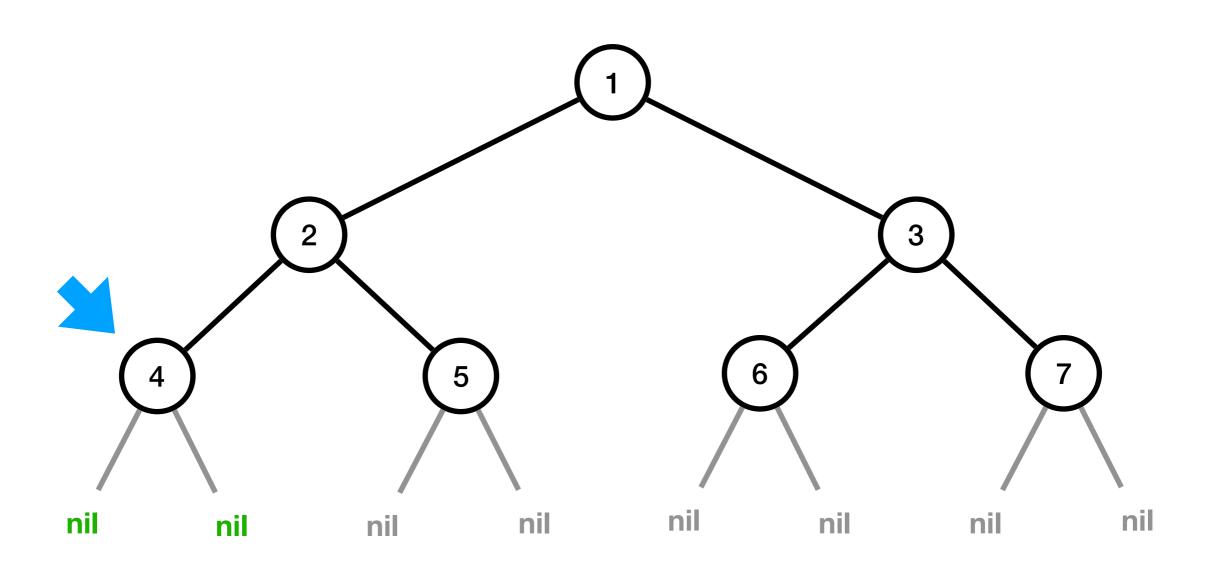


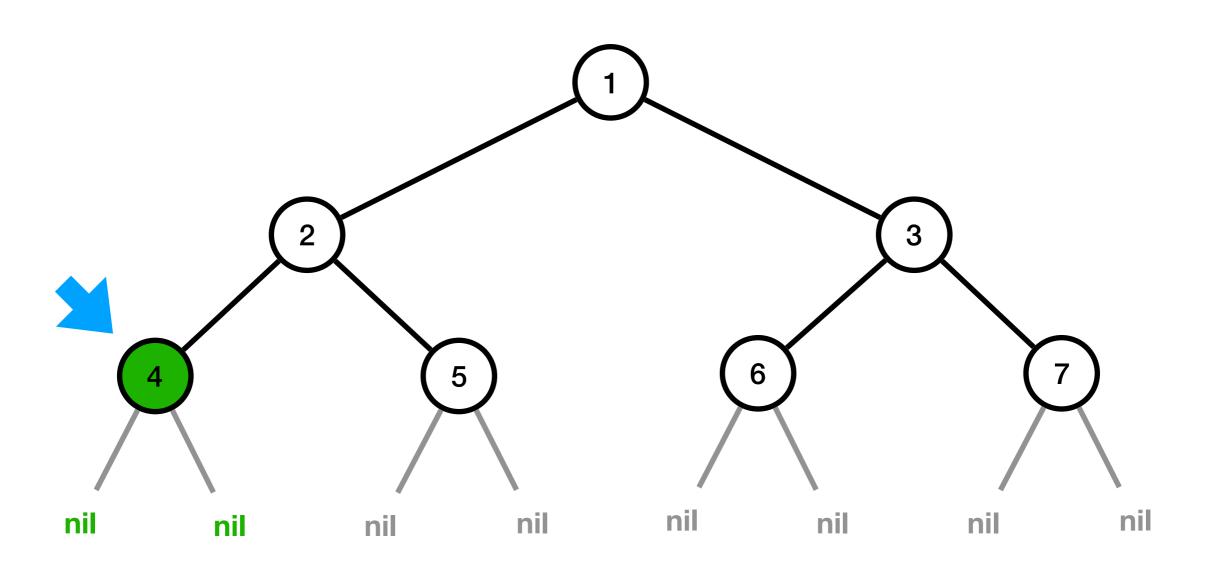




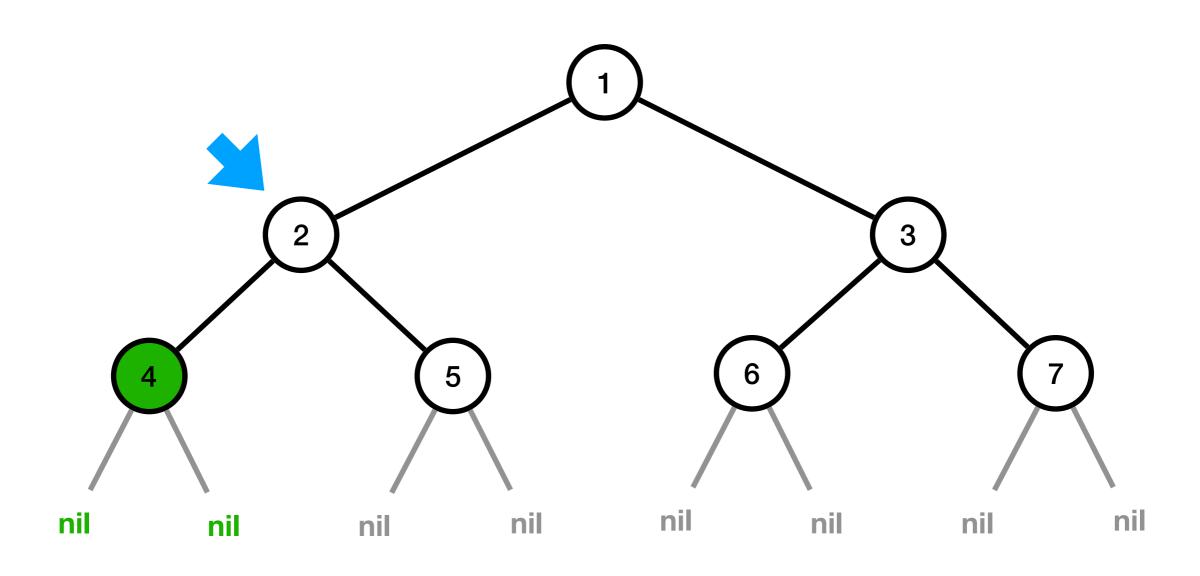




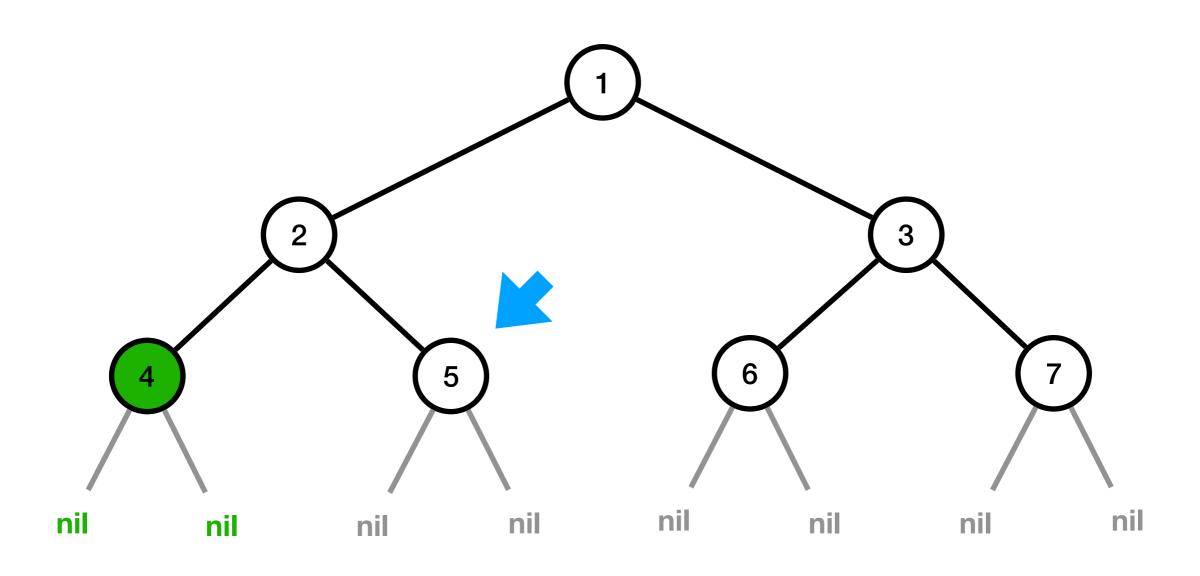




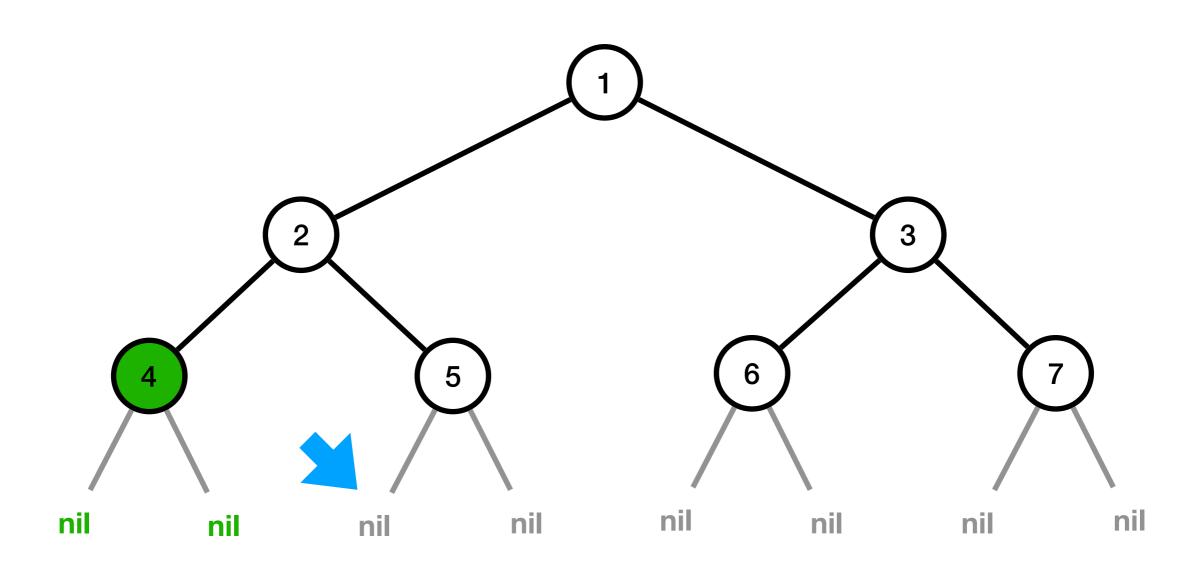




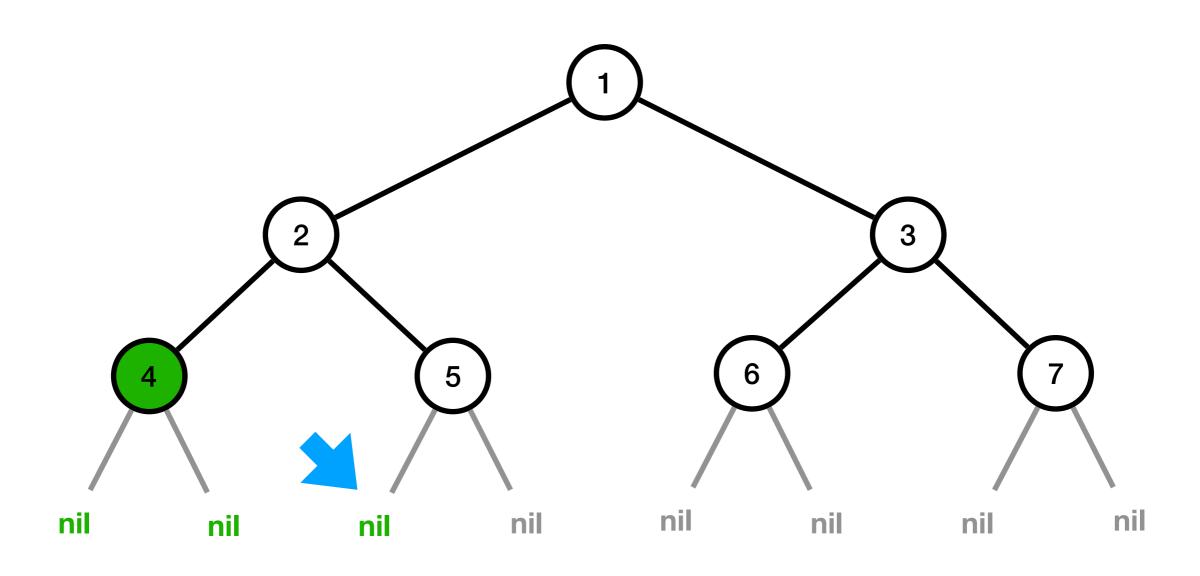




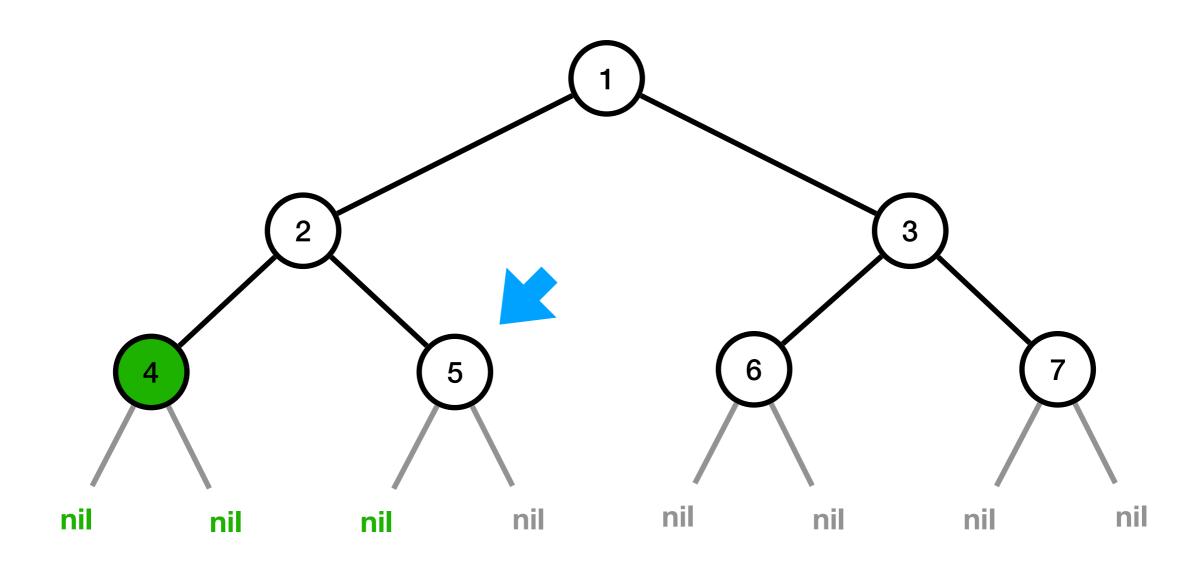




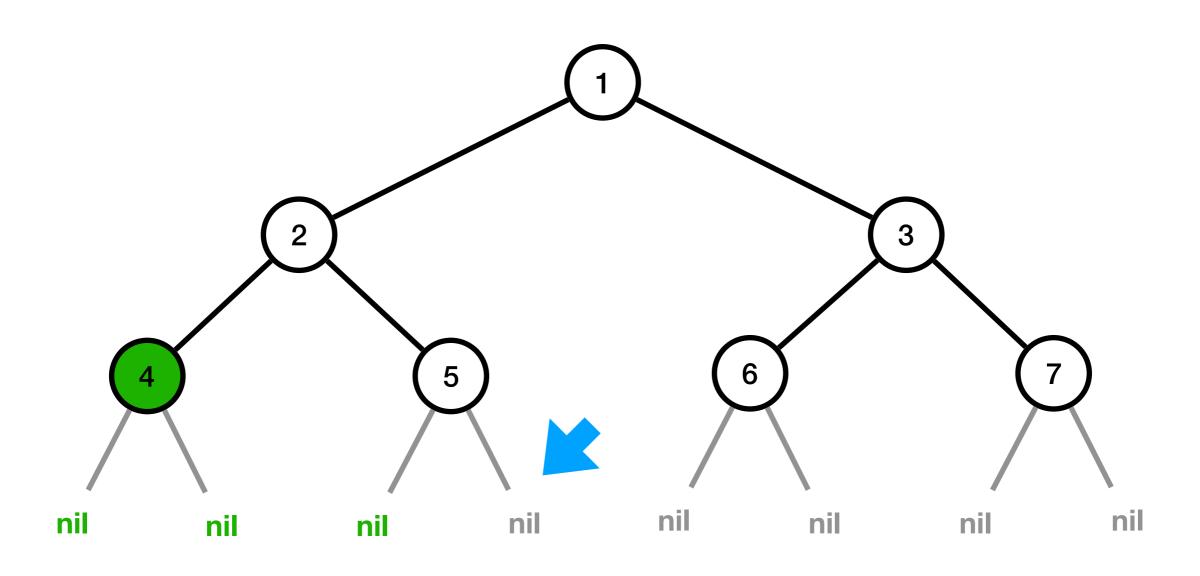




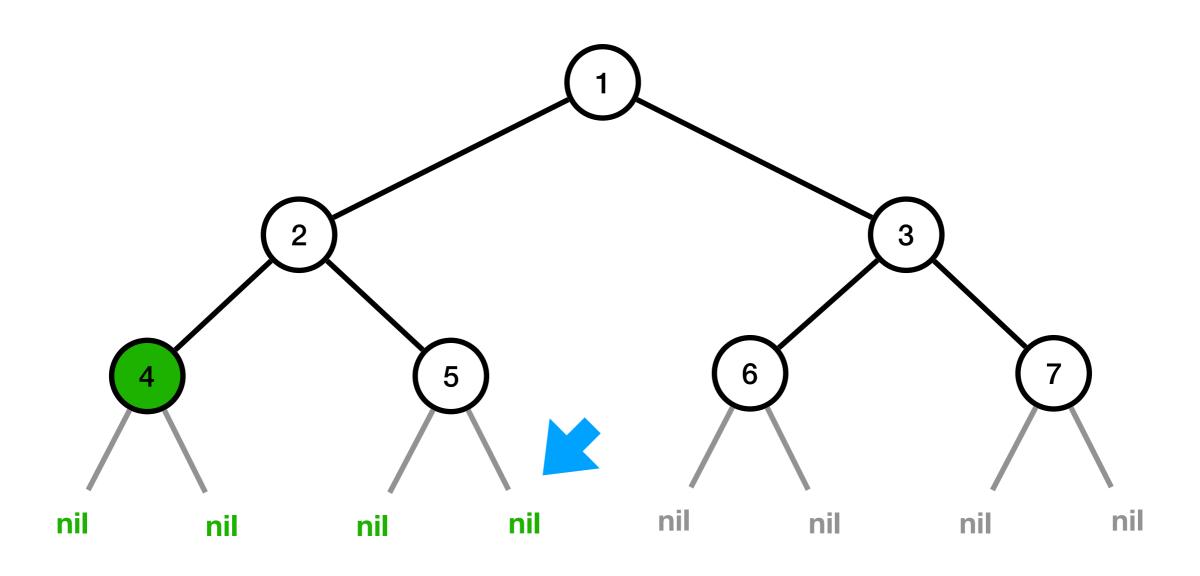




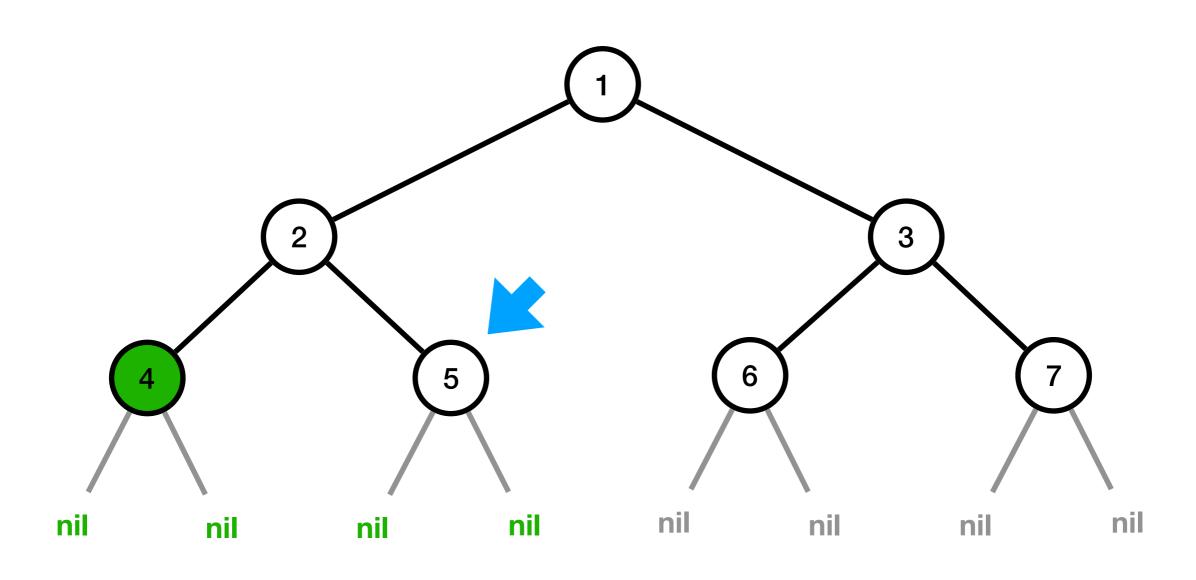




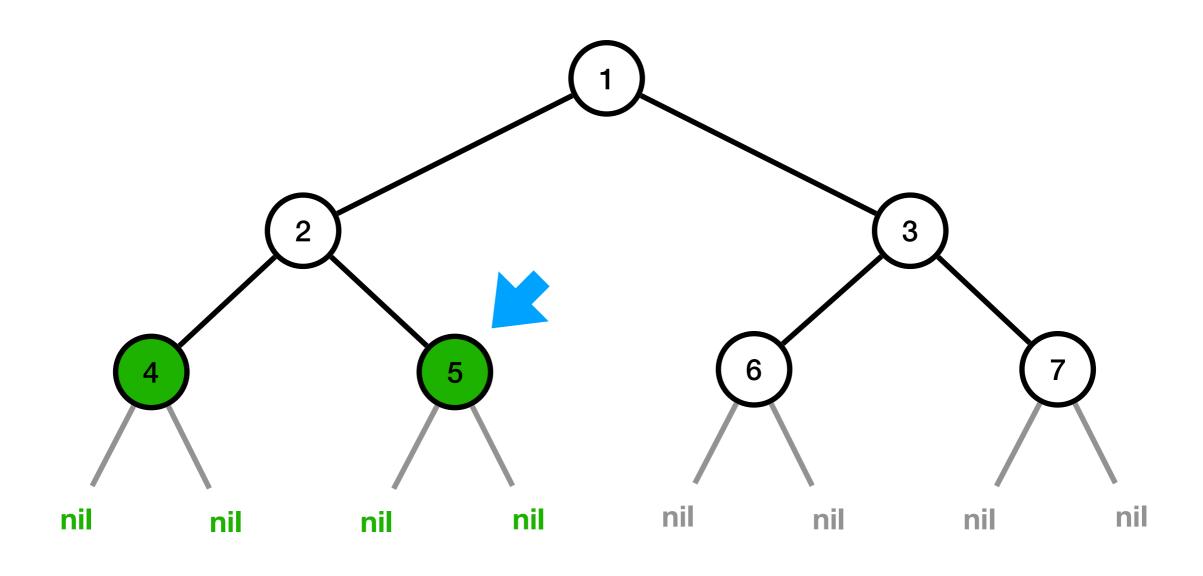




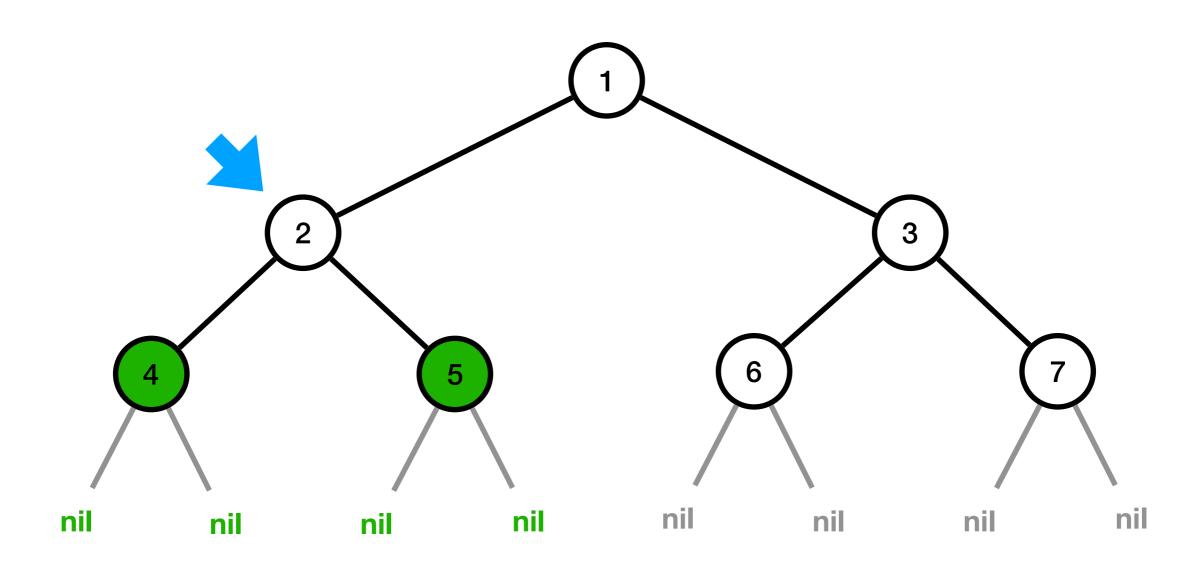






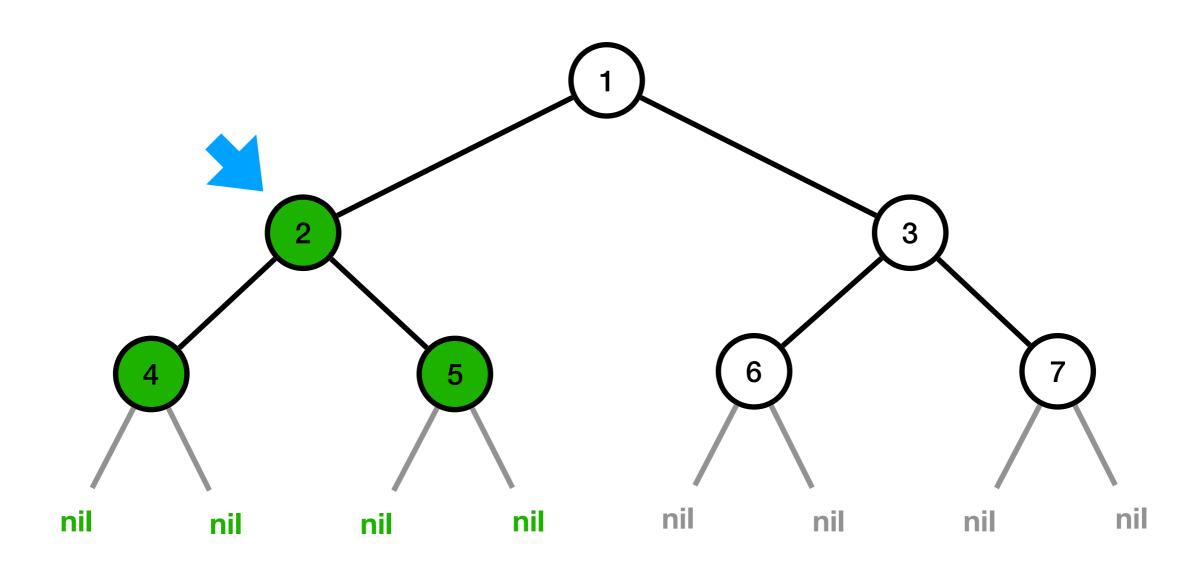


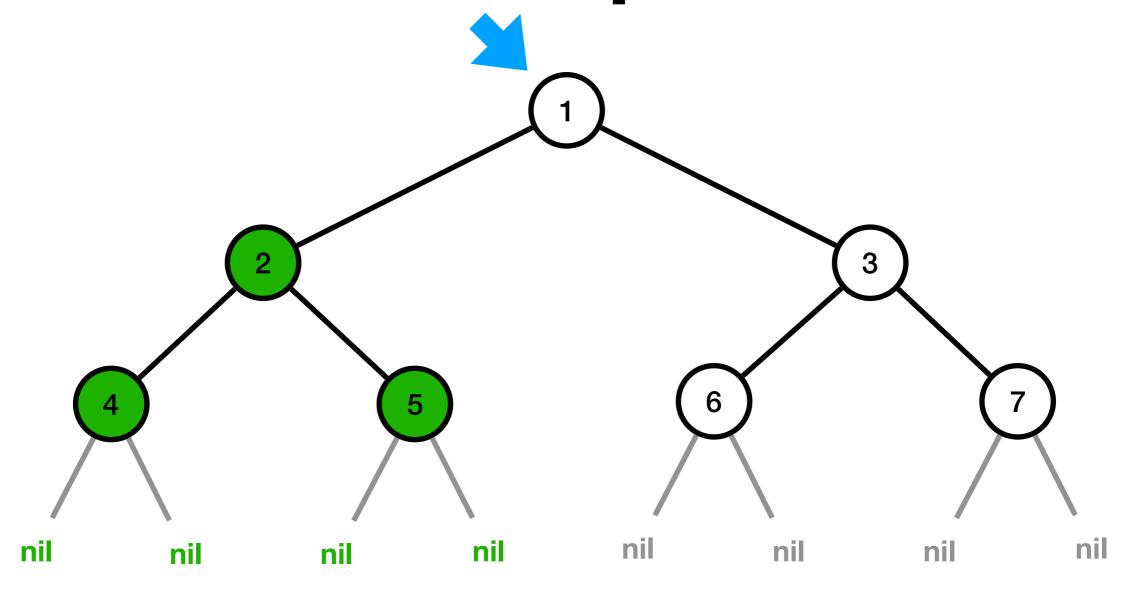








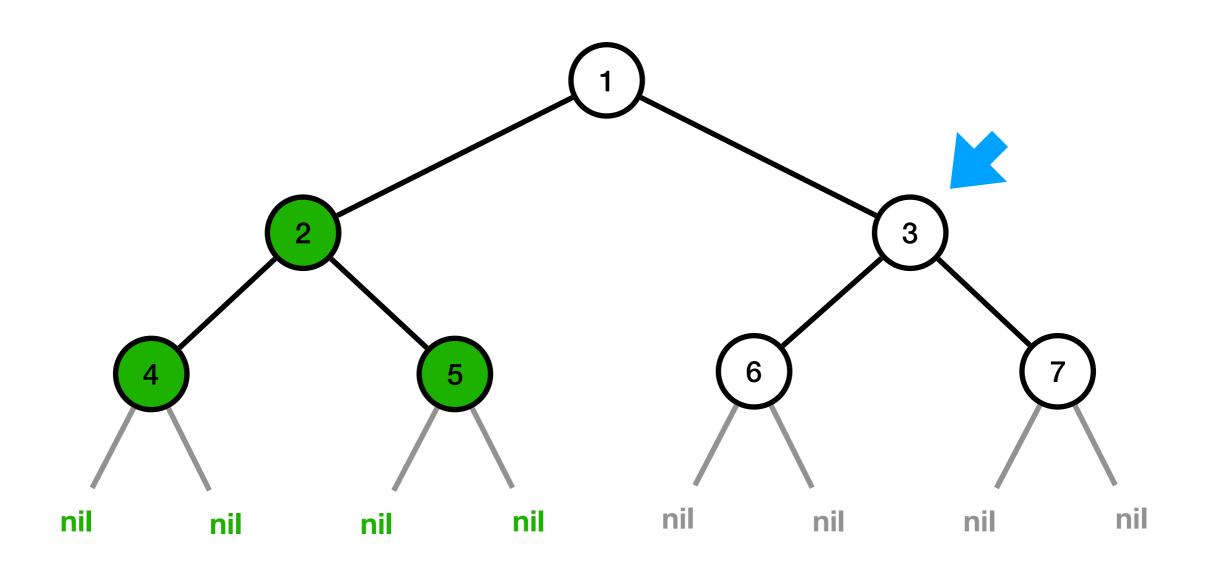


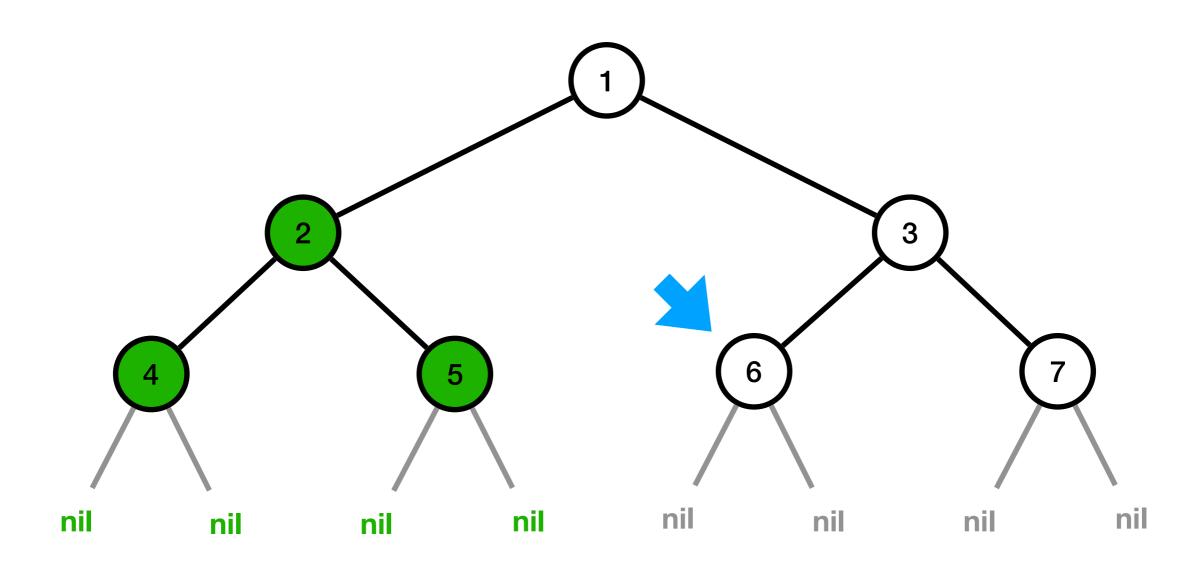


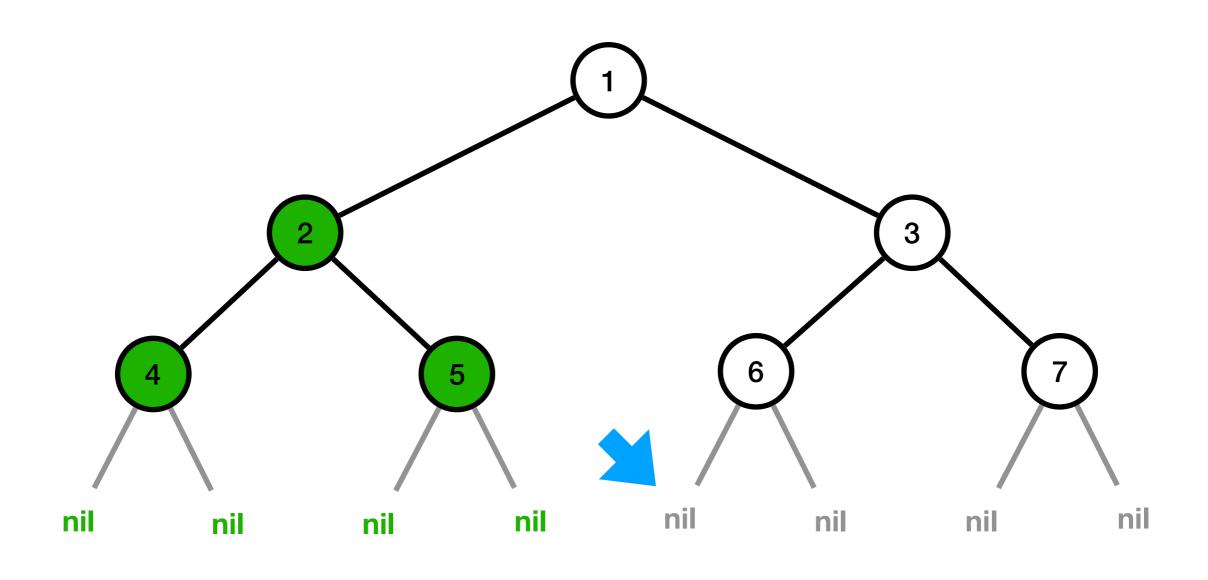


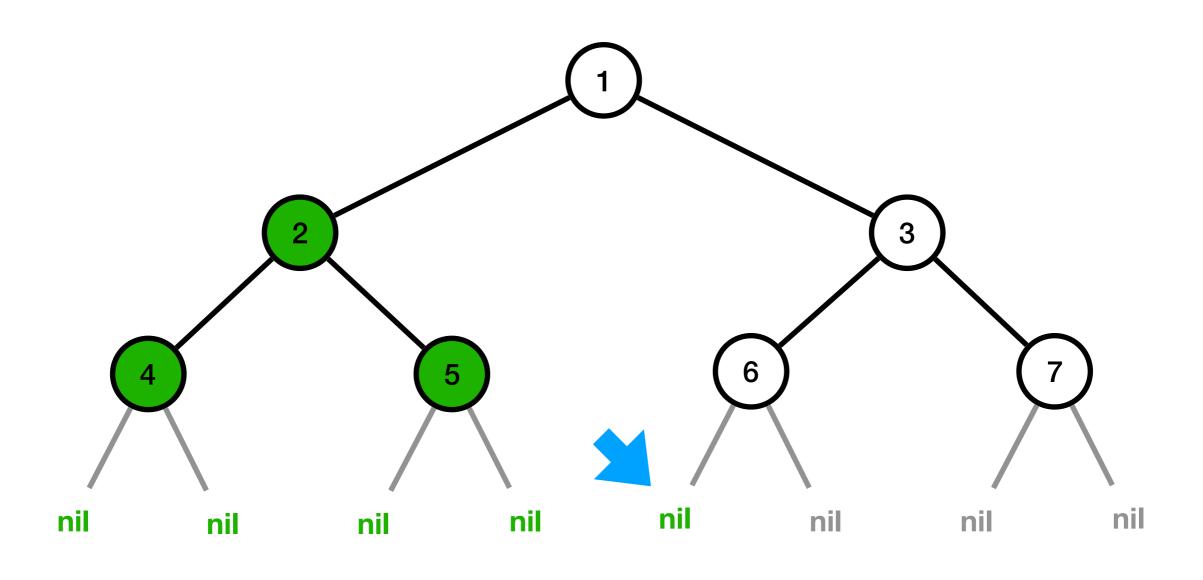


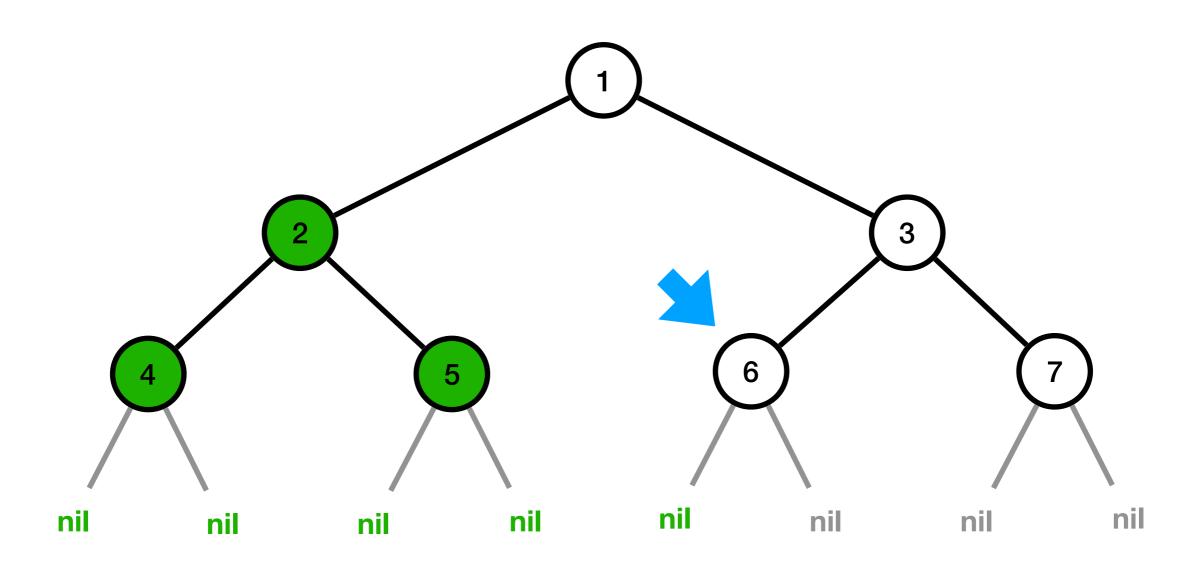


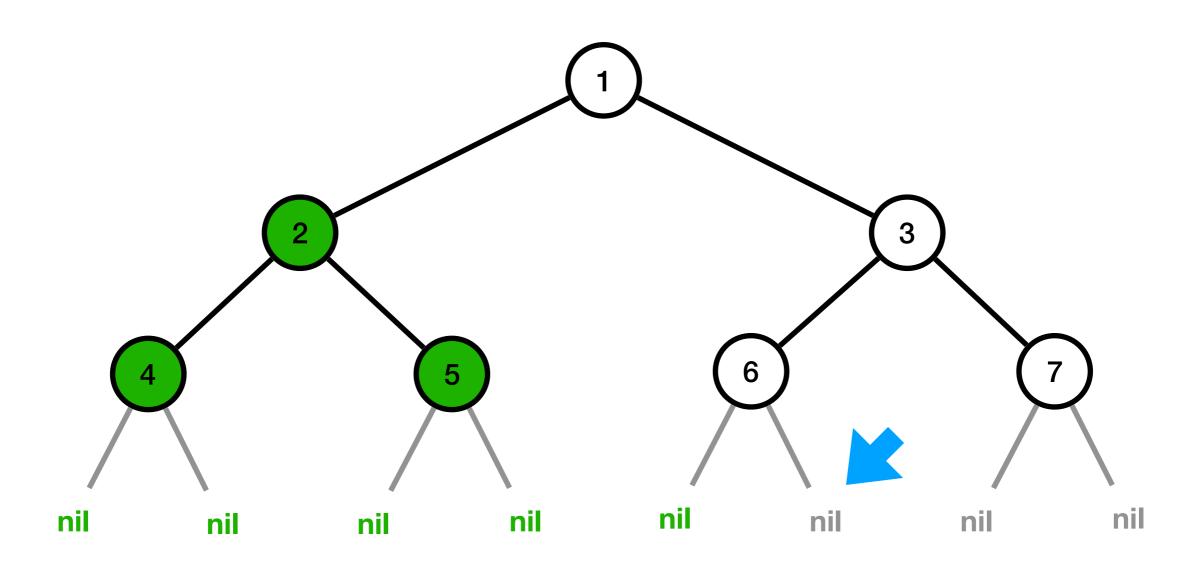


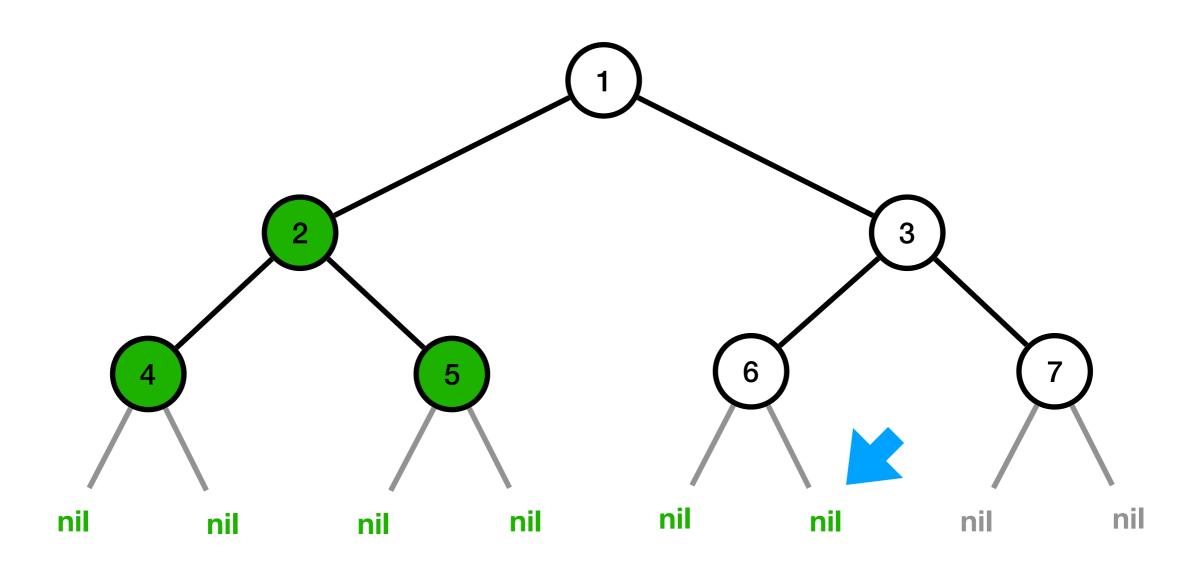


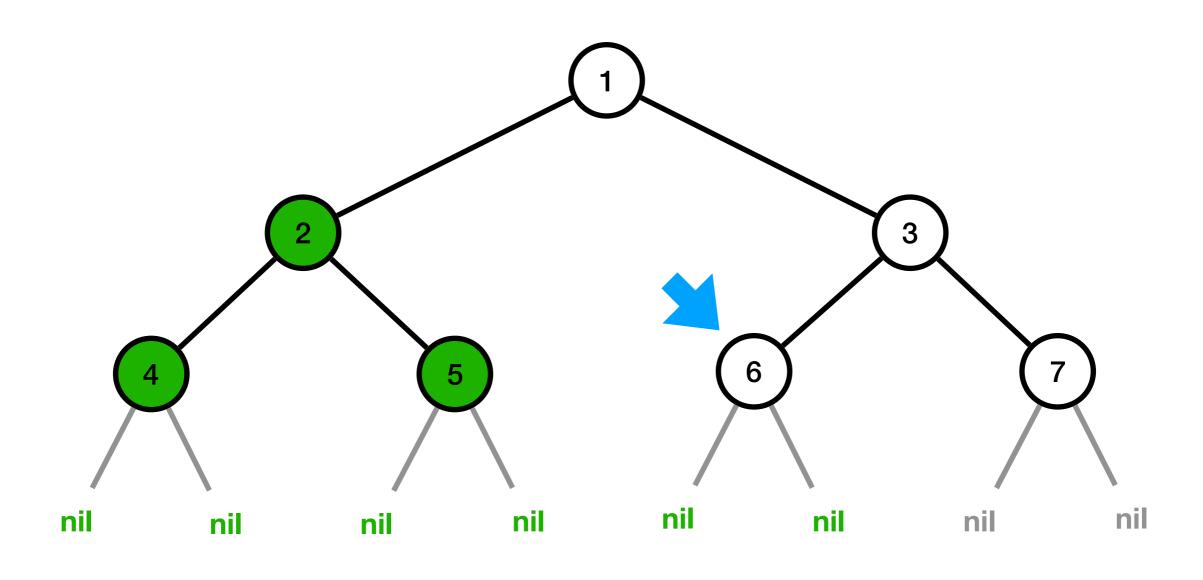


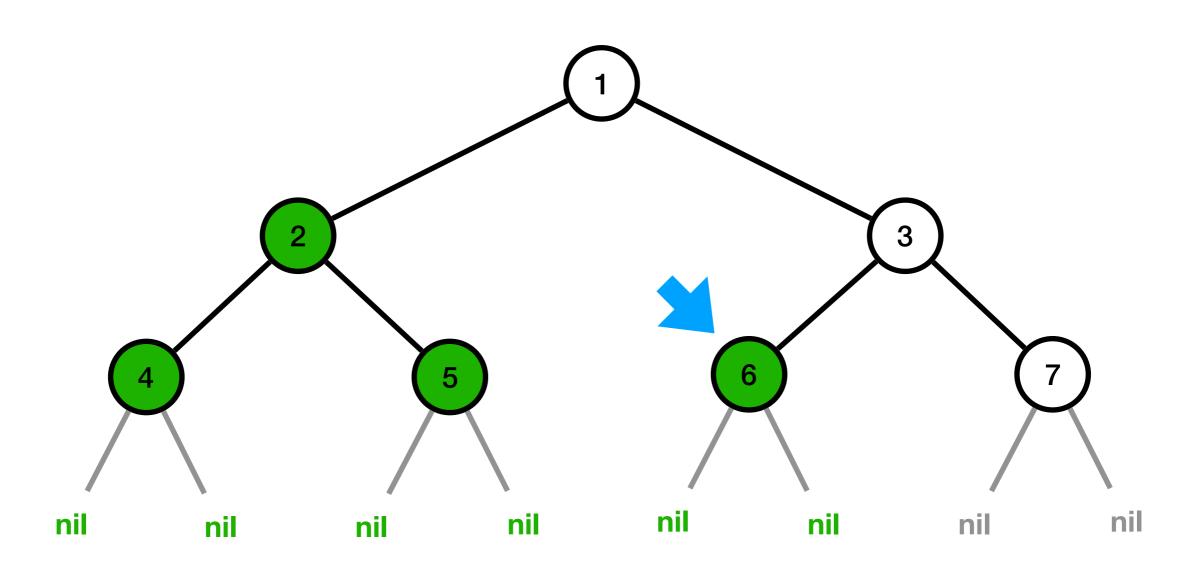


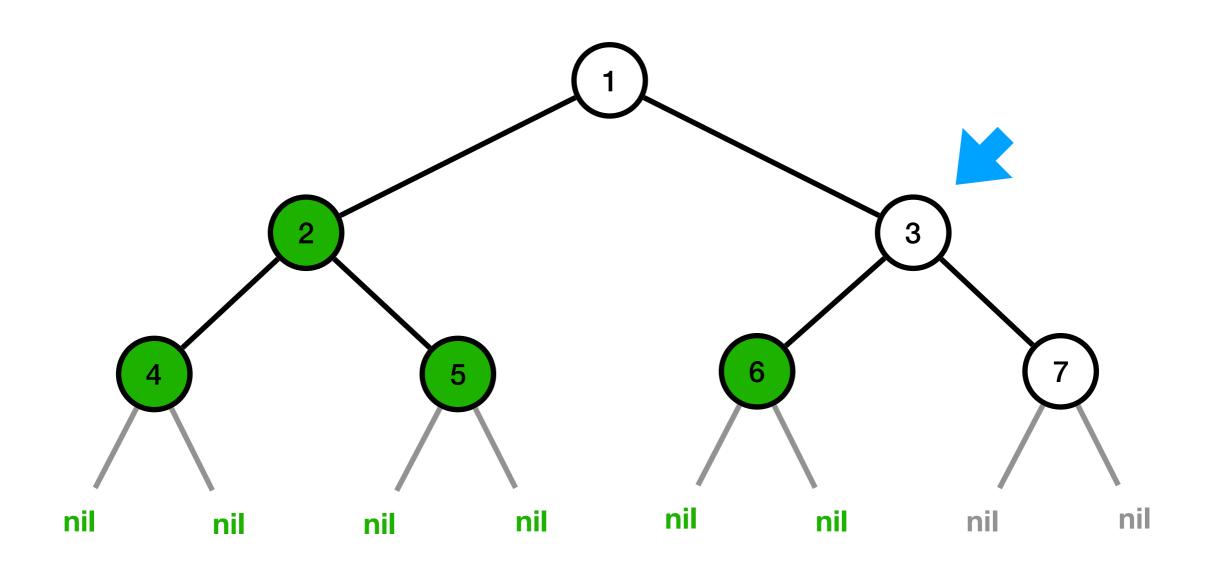


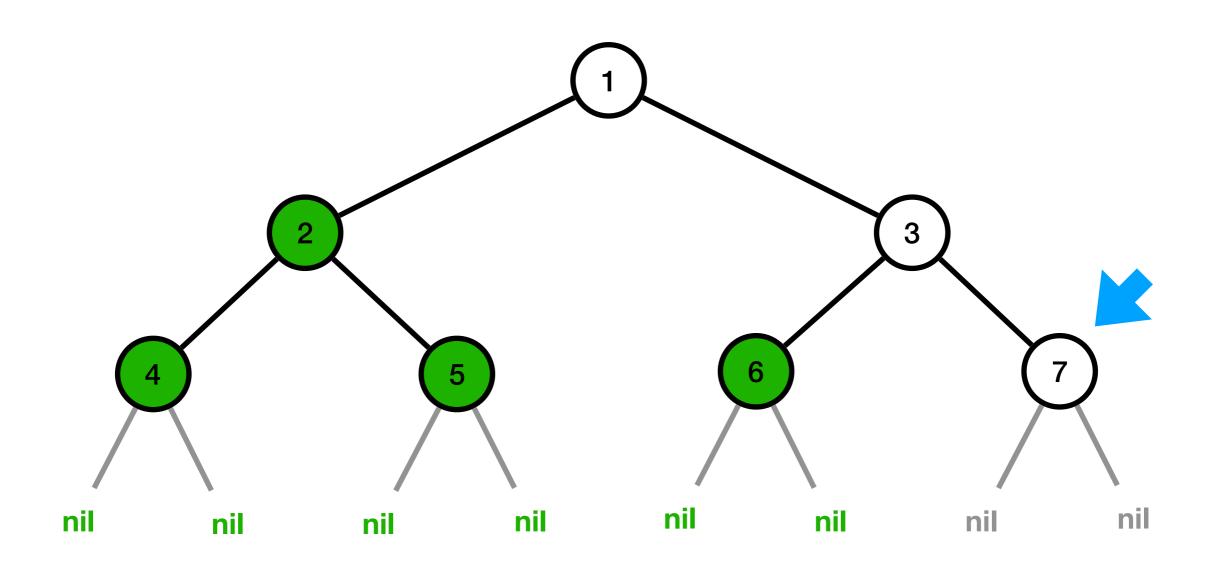






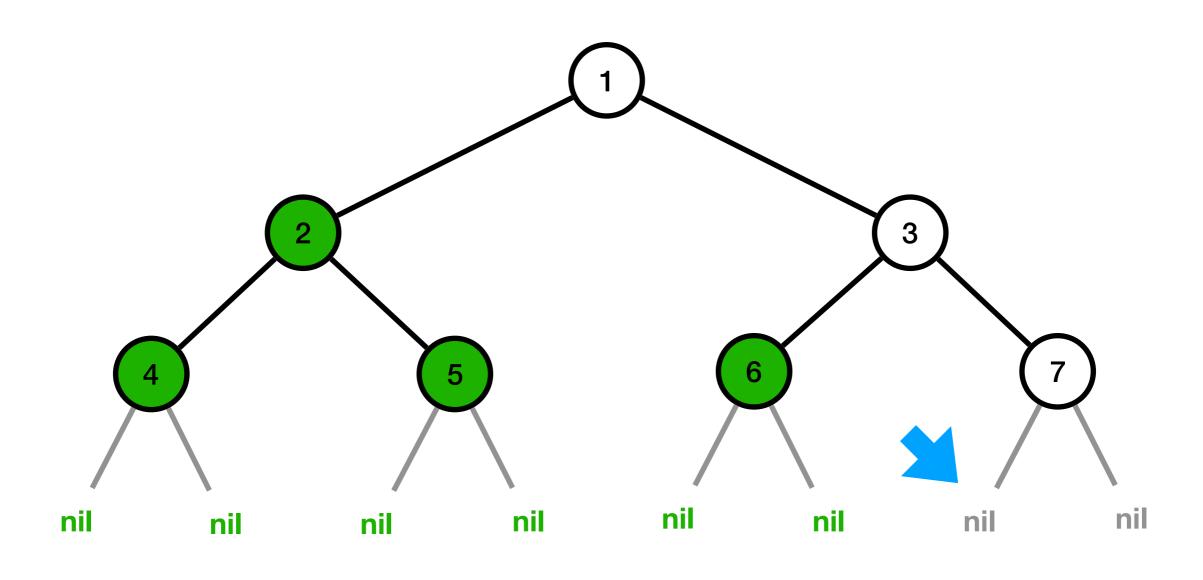


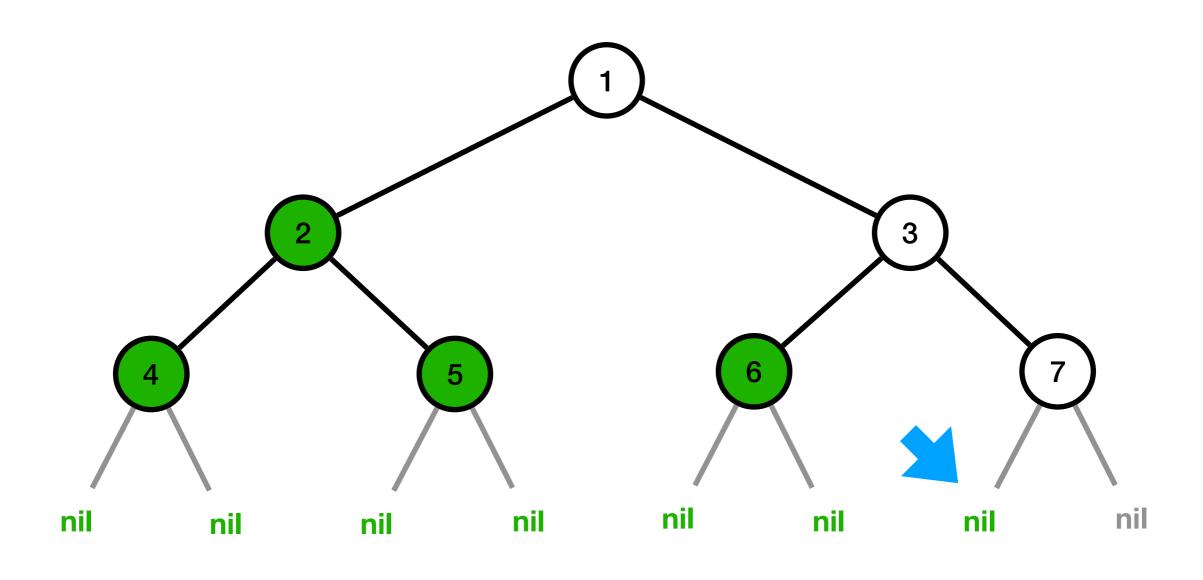


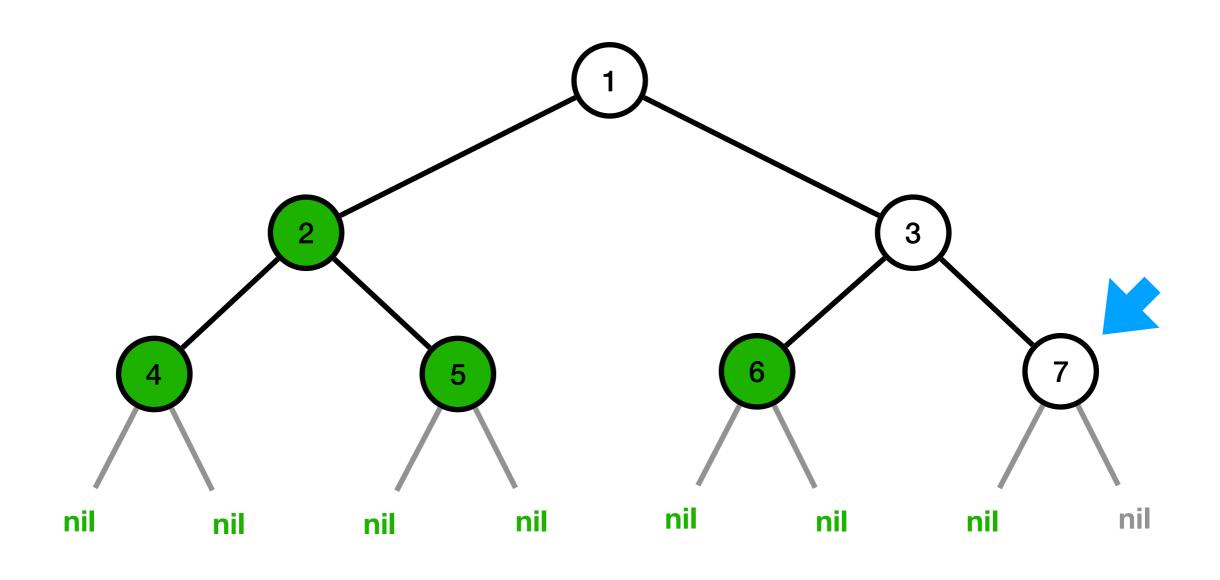


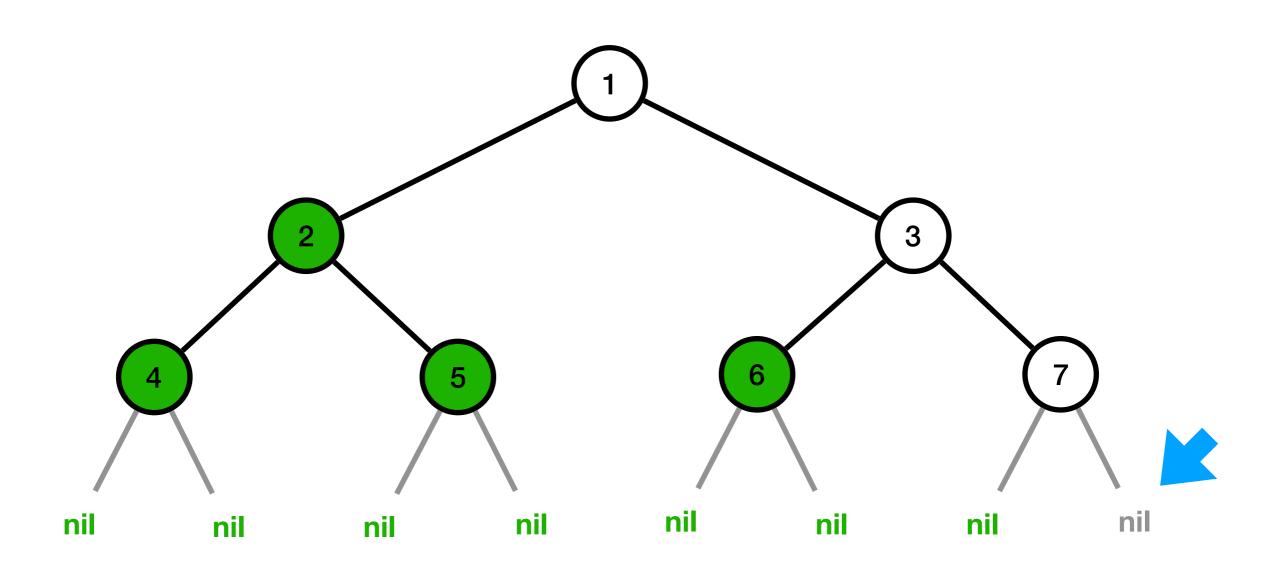


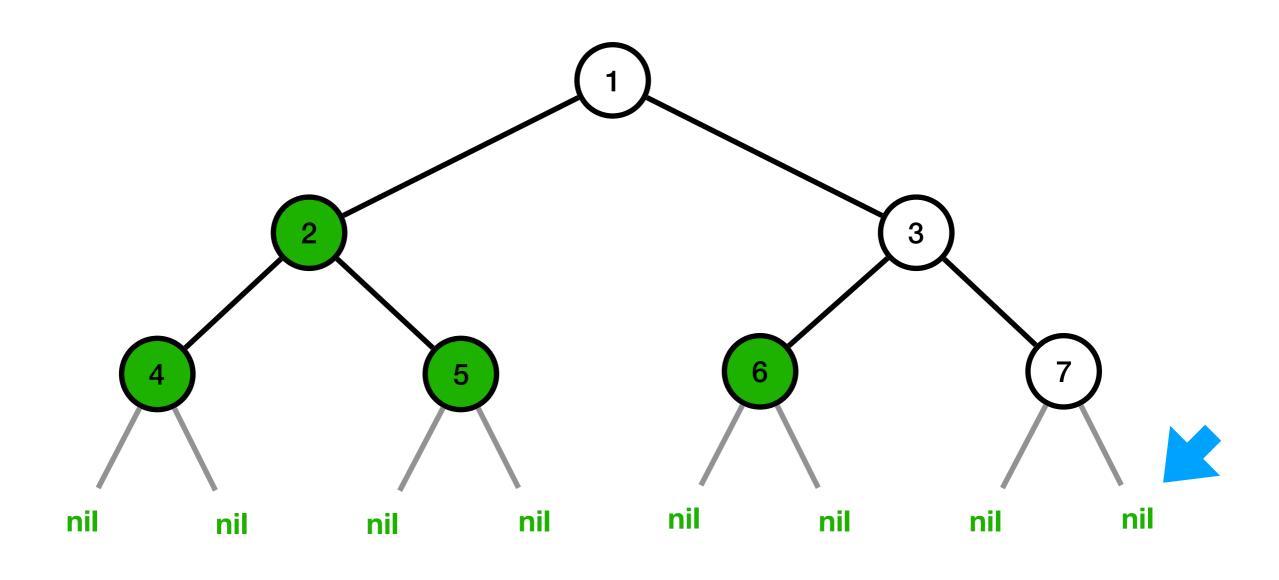


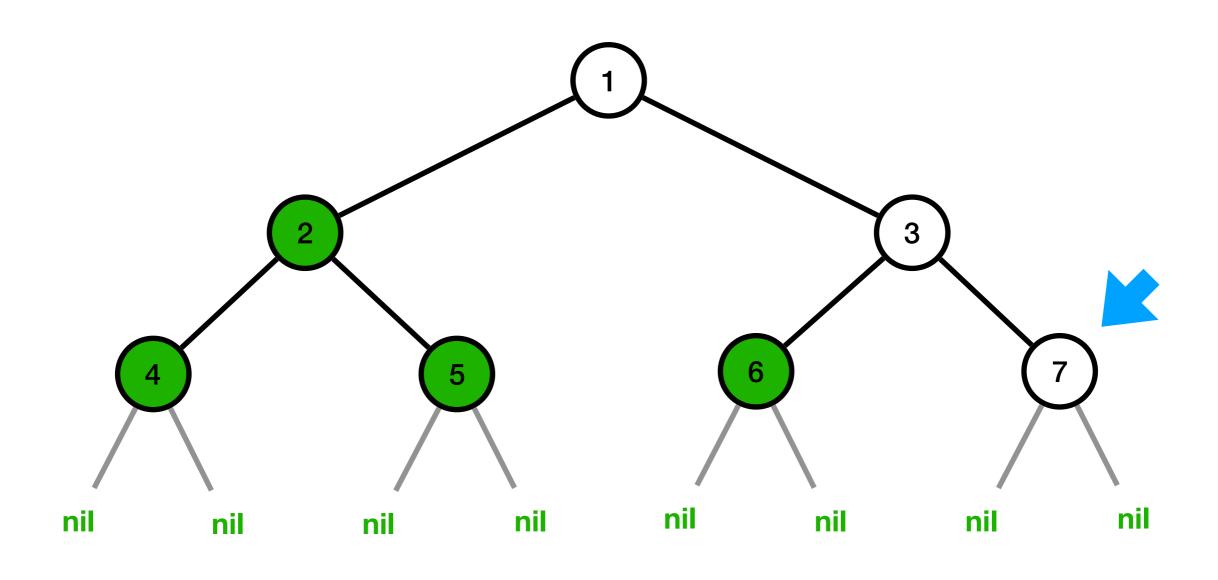


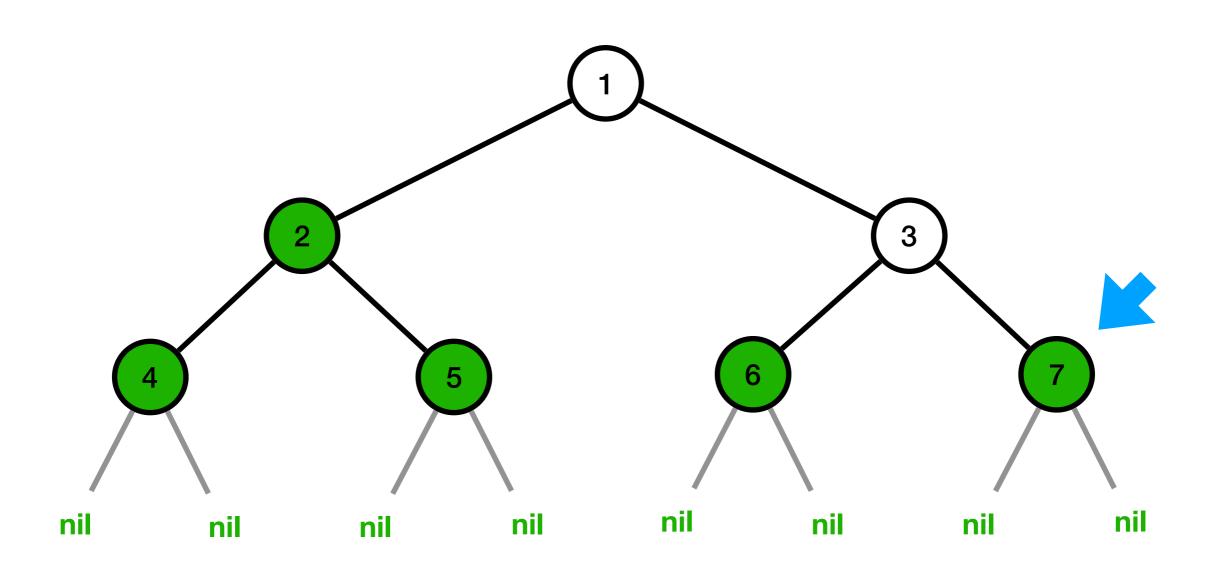


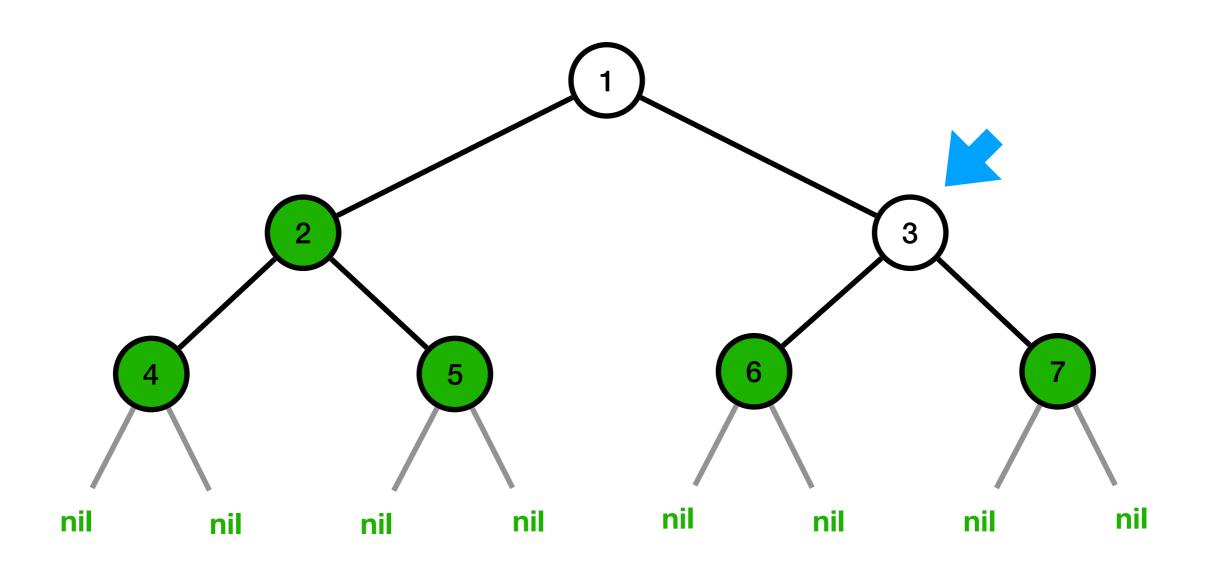


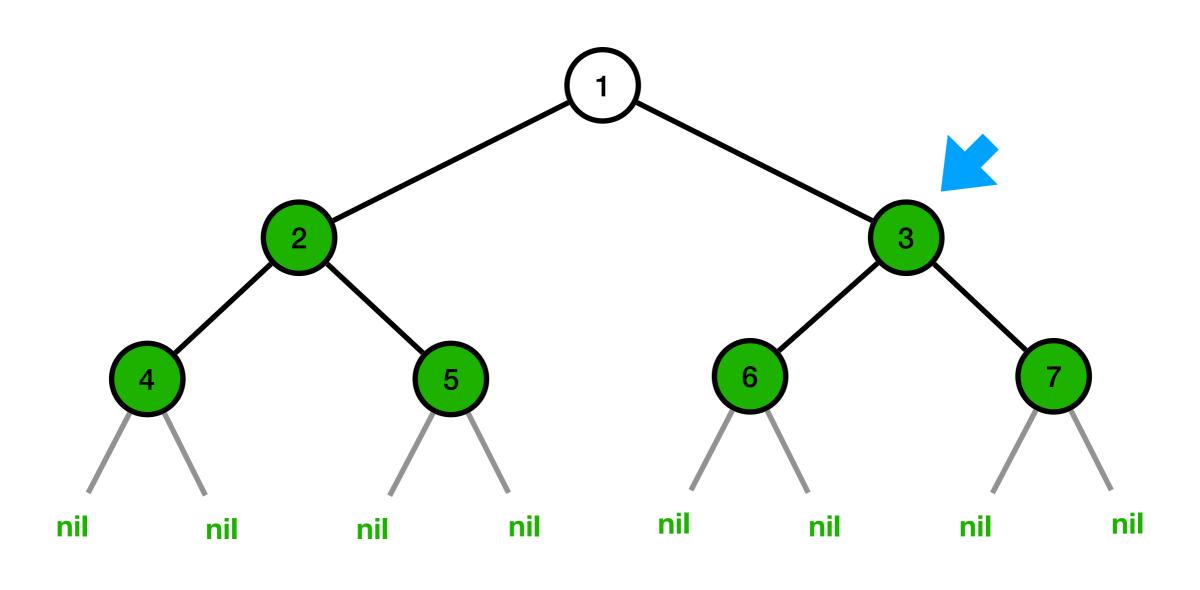


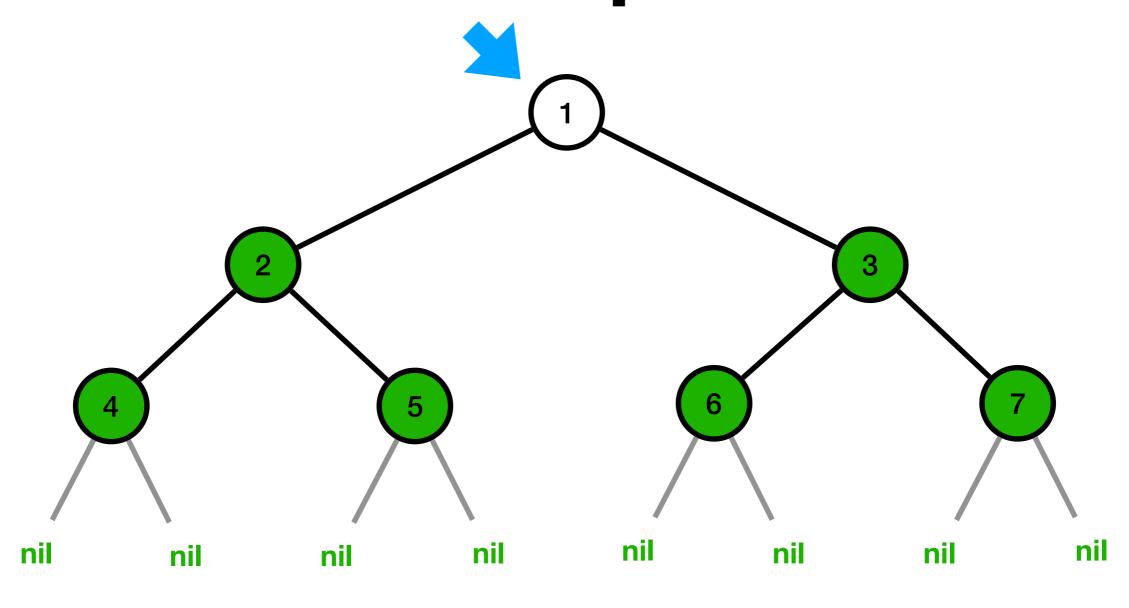


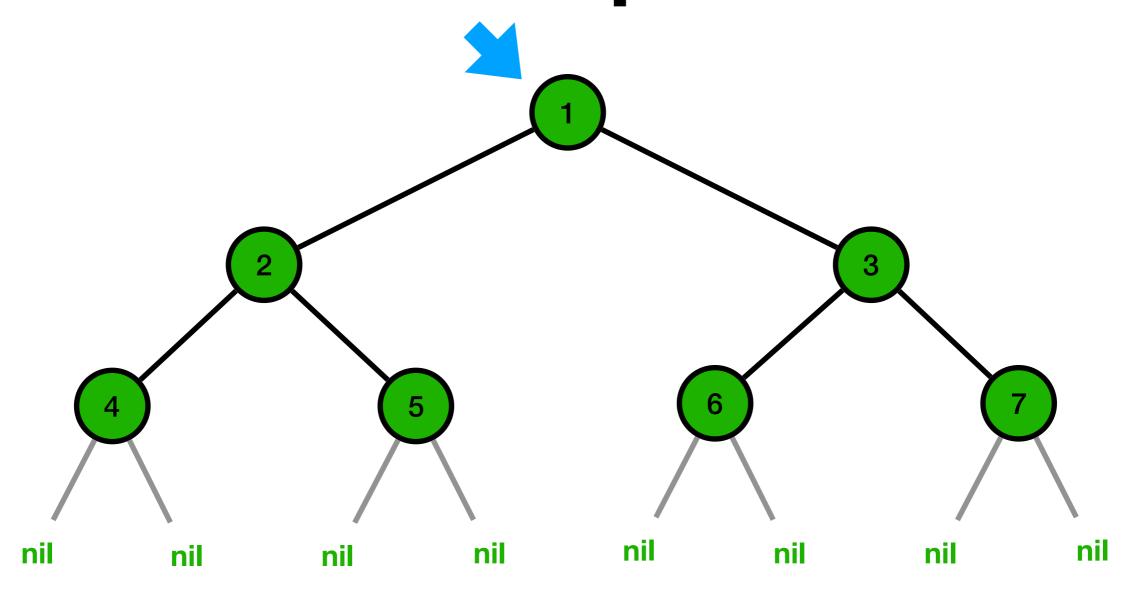


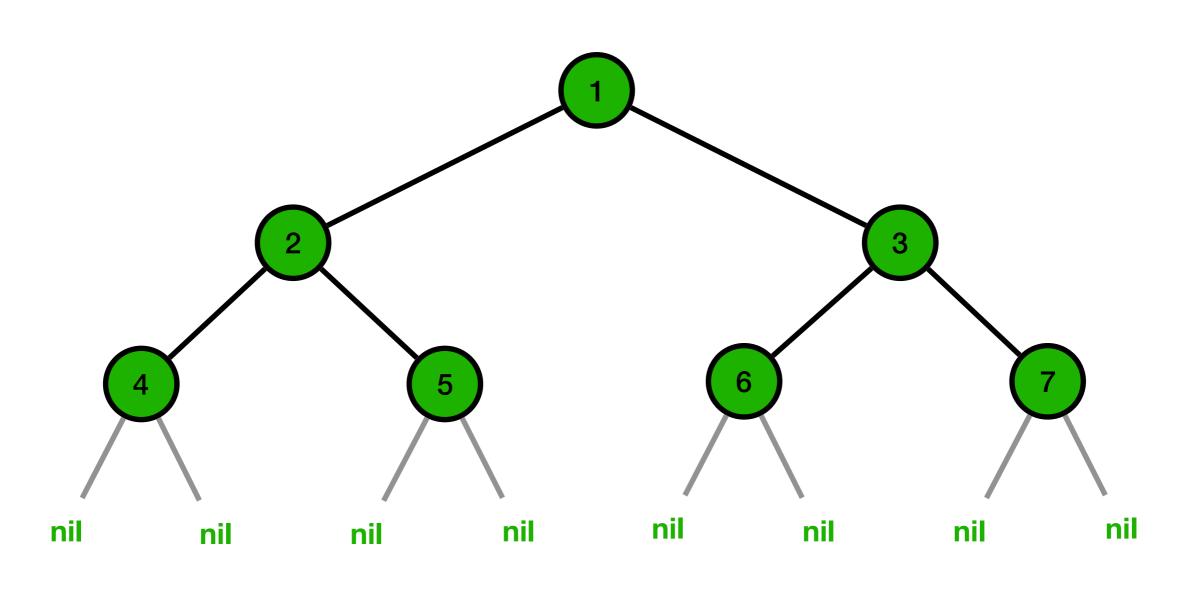








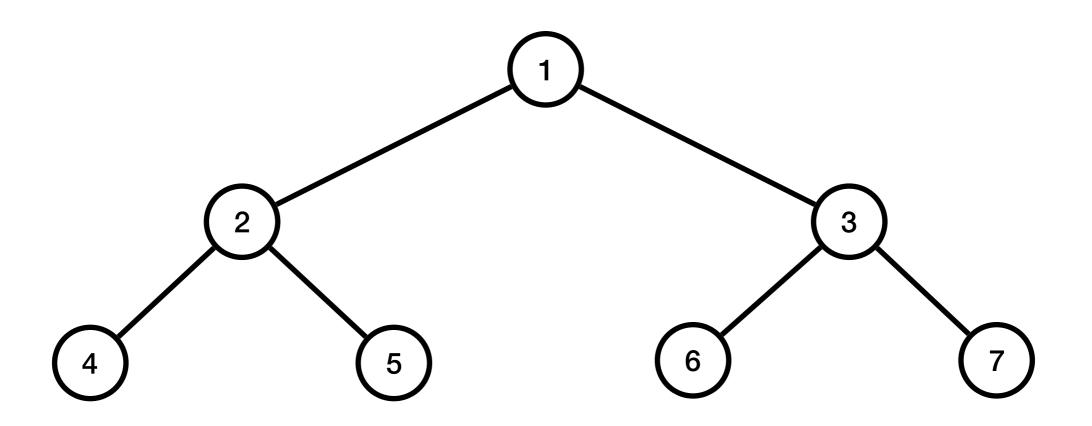


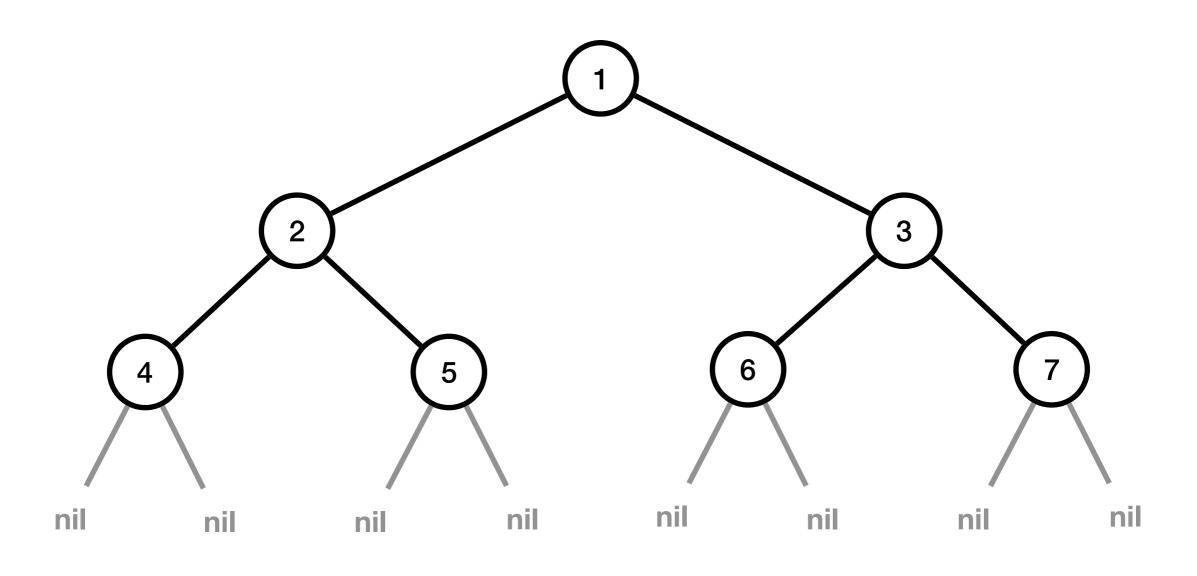


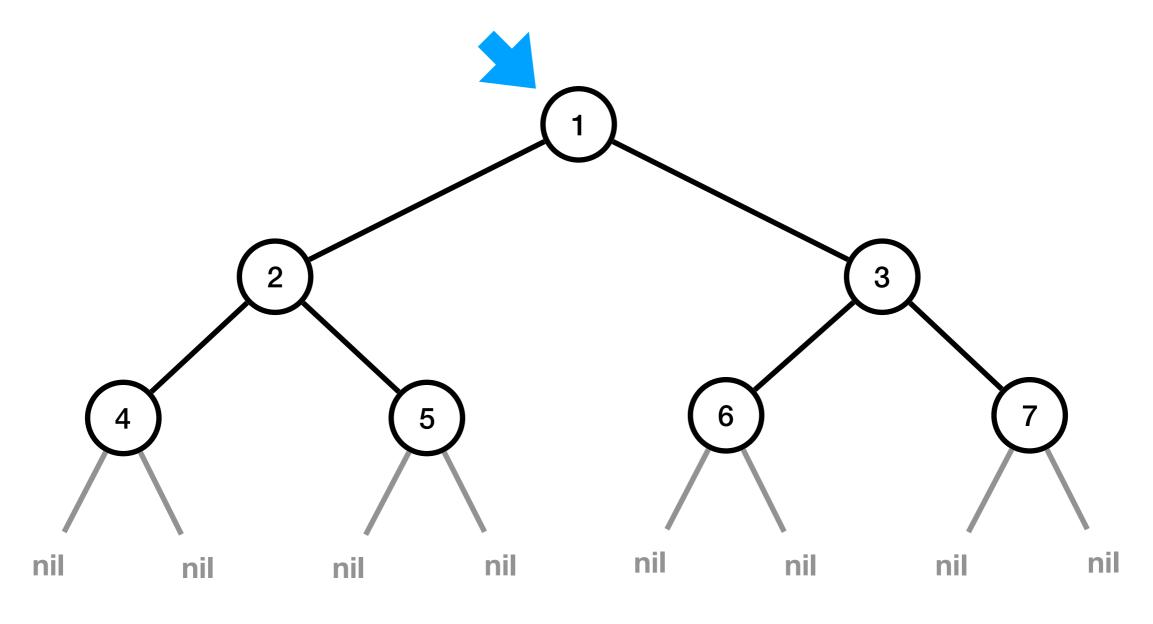
Parcours infixe

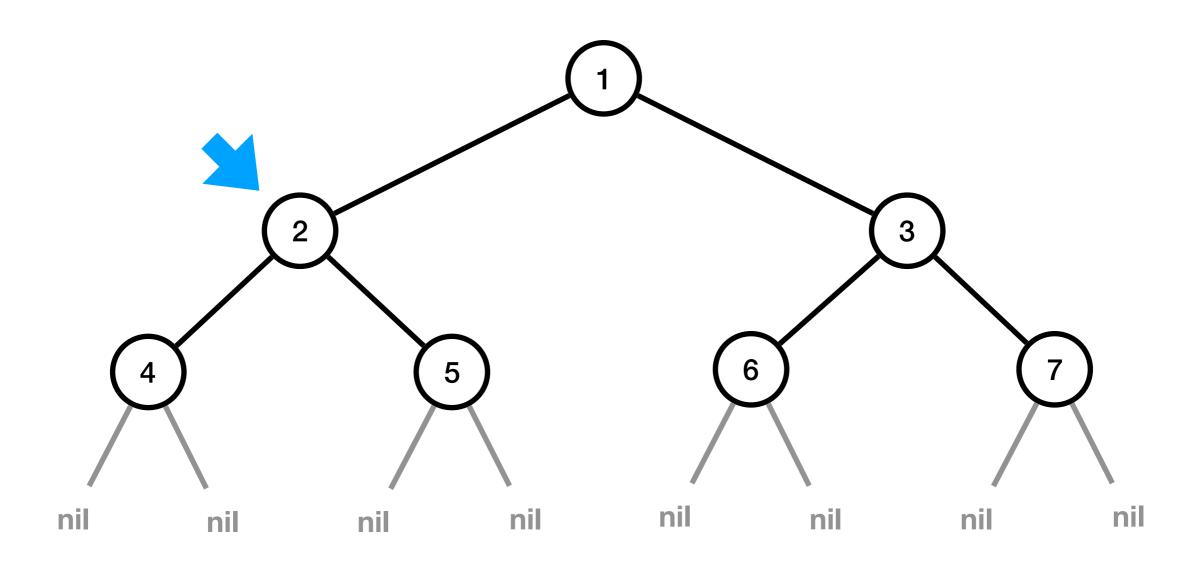
```
procédure visiter-infixe(nœud)
    si nœud ≠ nil alors
    visiter-infixe(gauche[nœud])
    afficher(valeur[nœud])
    visiter-infixe(droite[nœud])
```

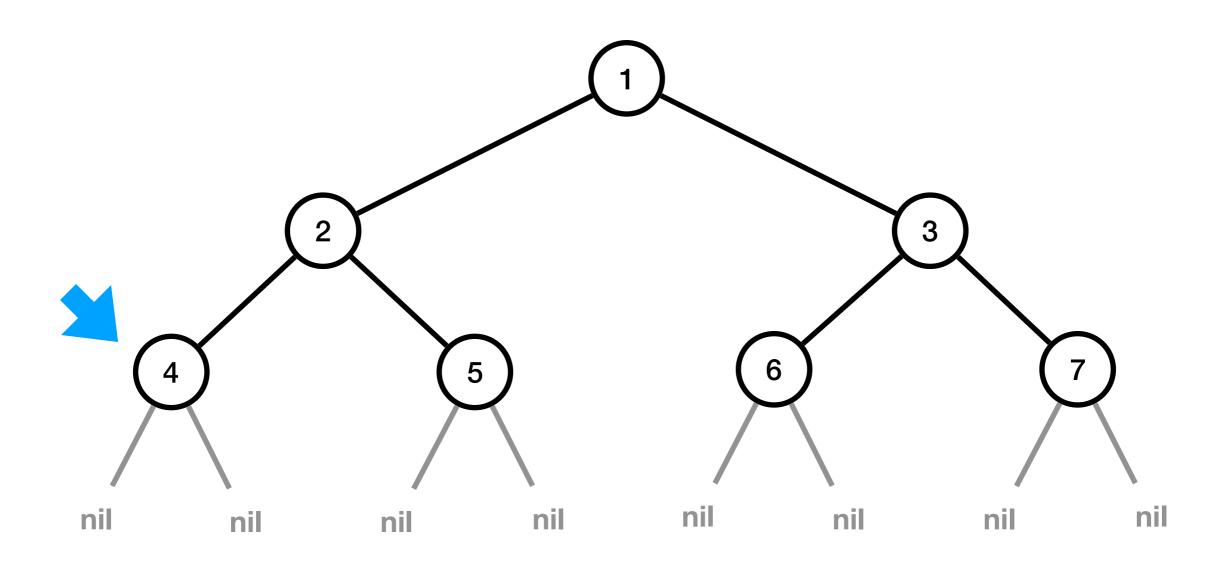
Parcours infixe

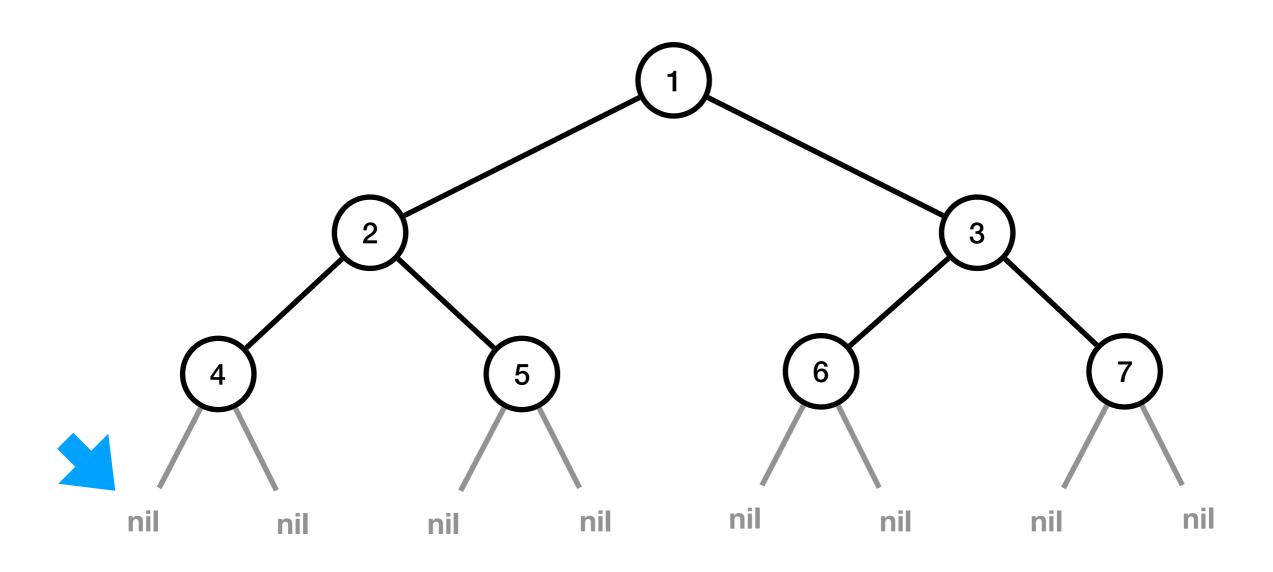


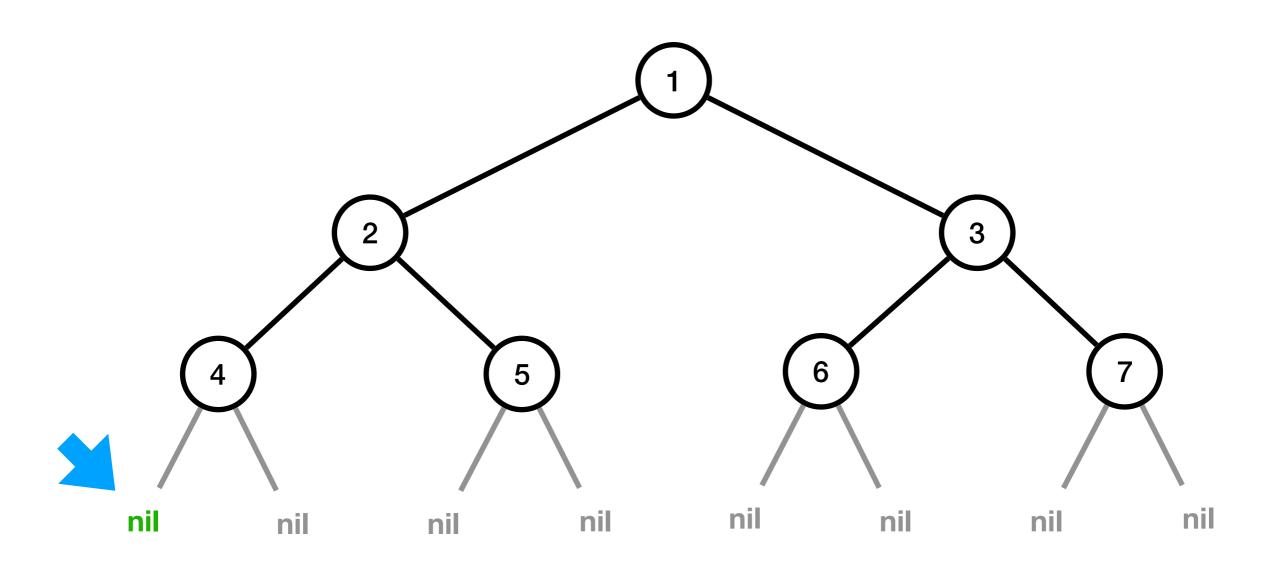


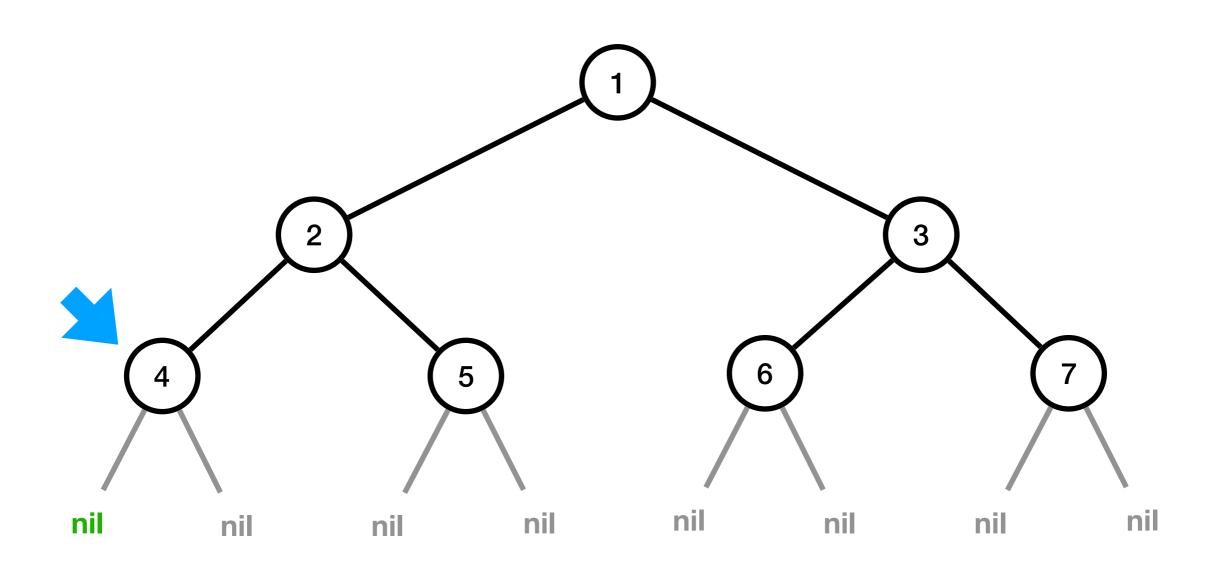


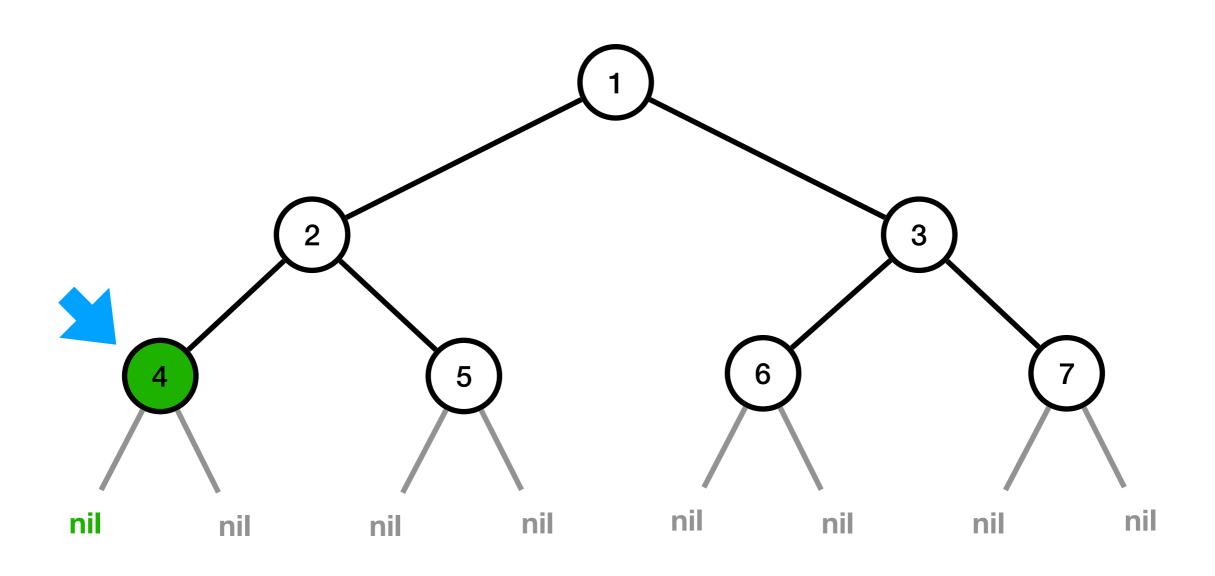




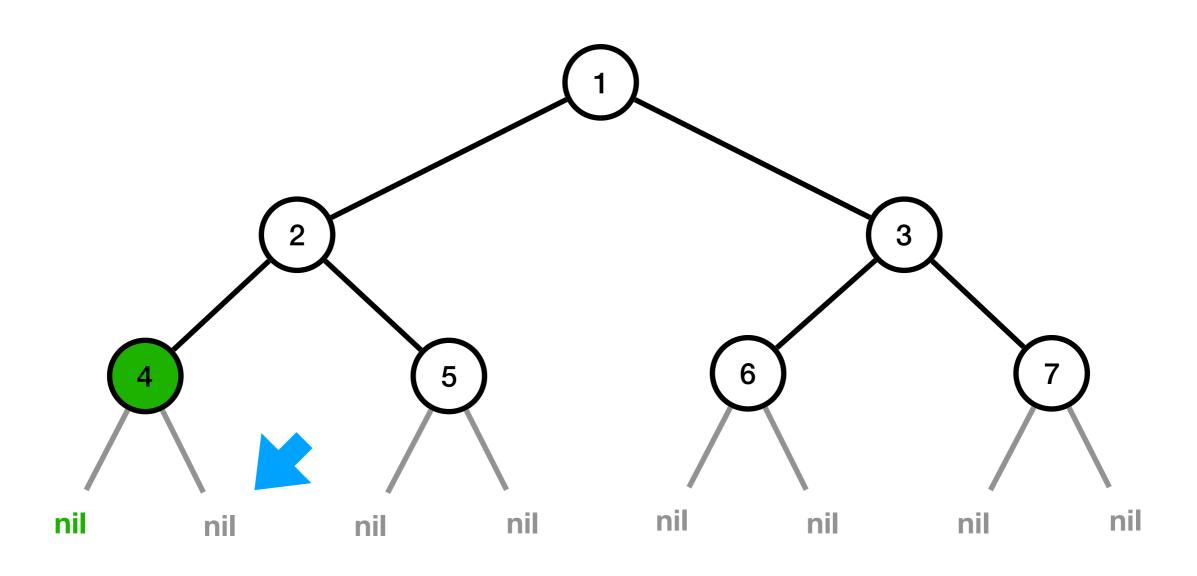




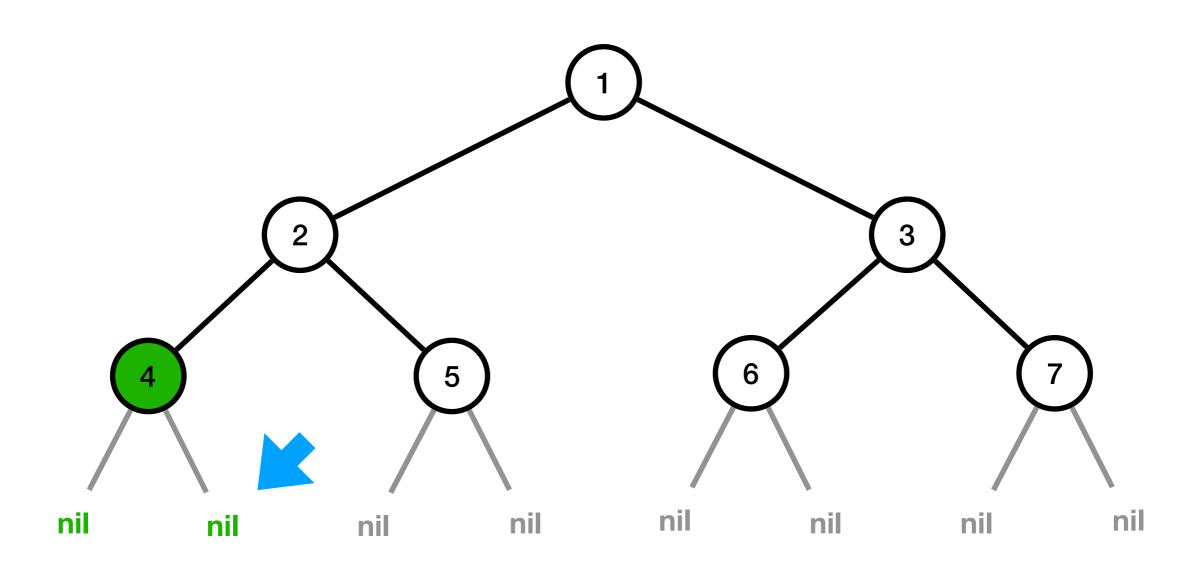




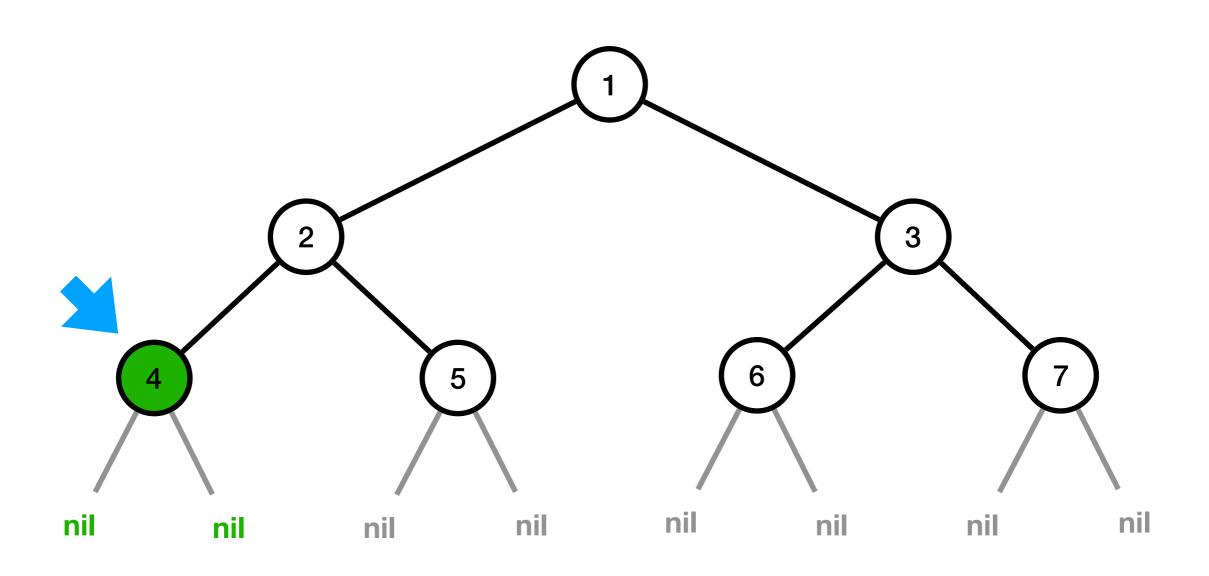




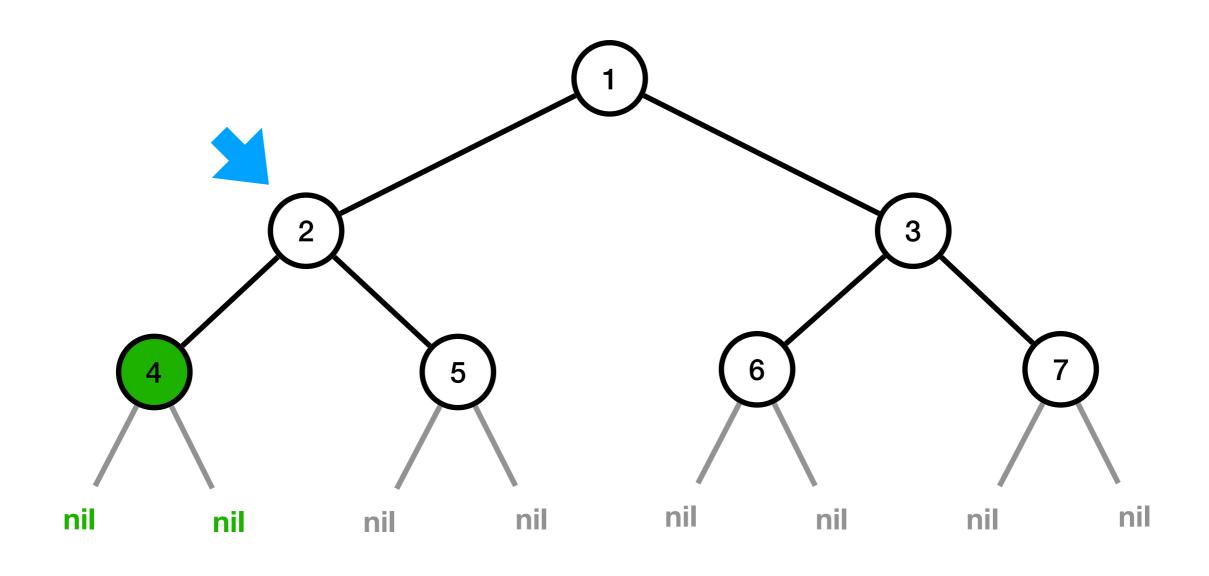




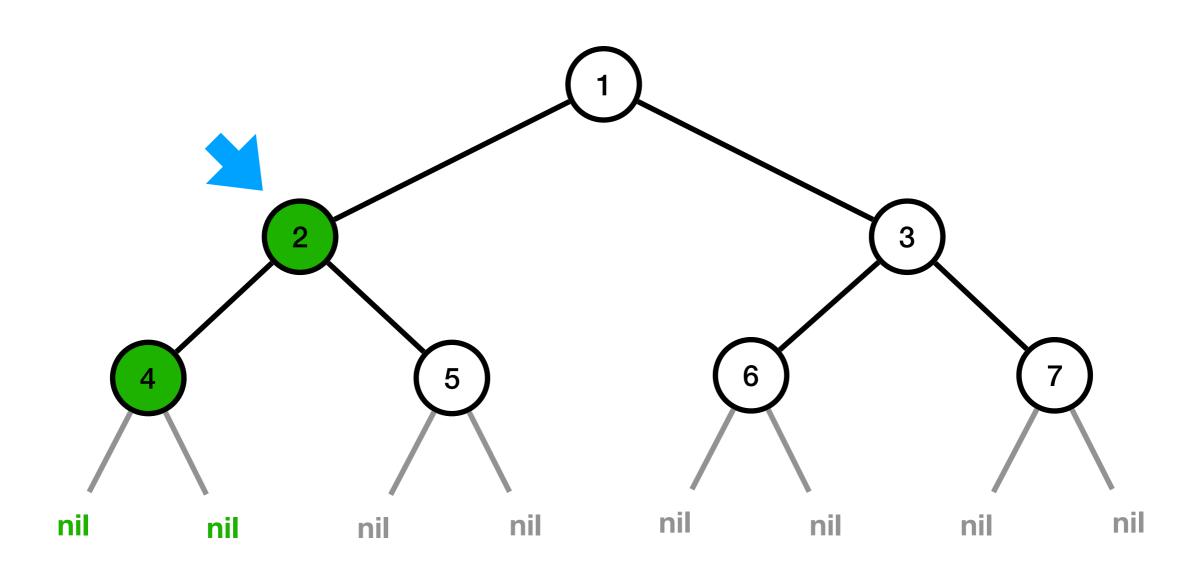




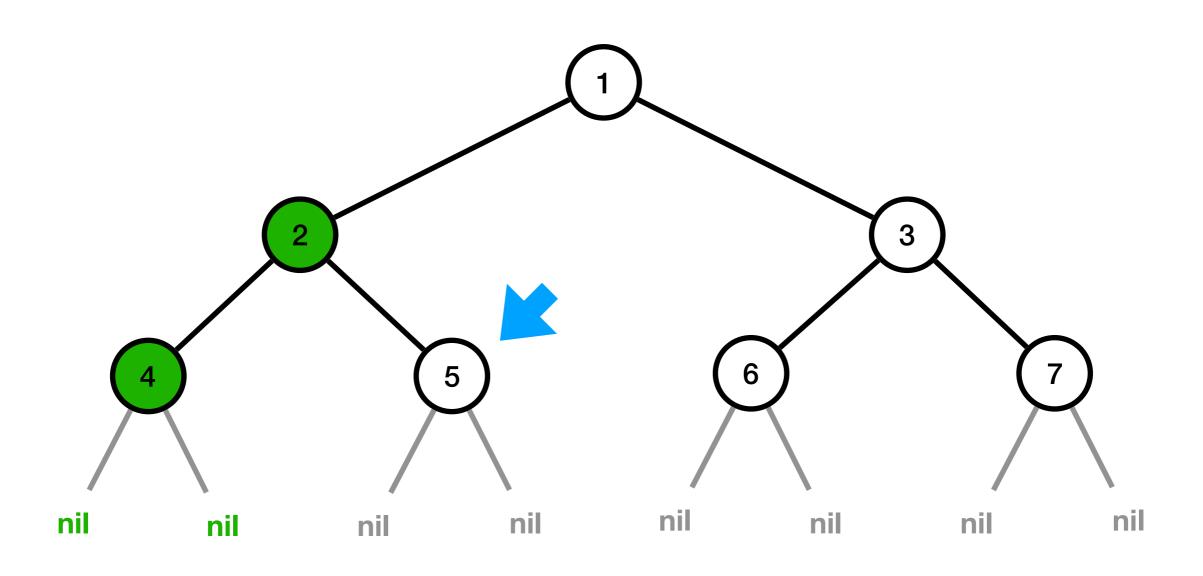




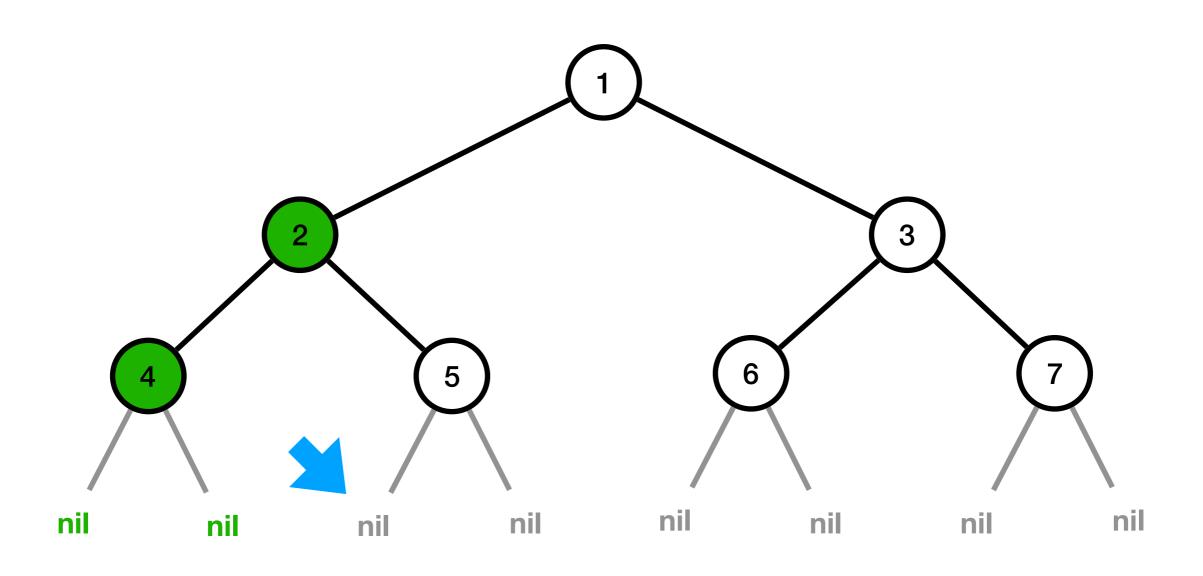




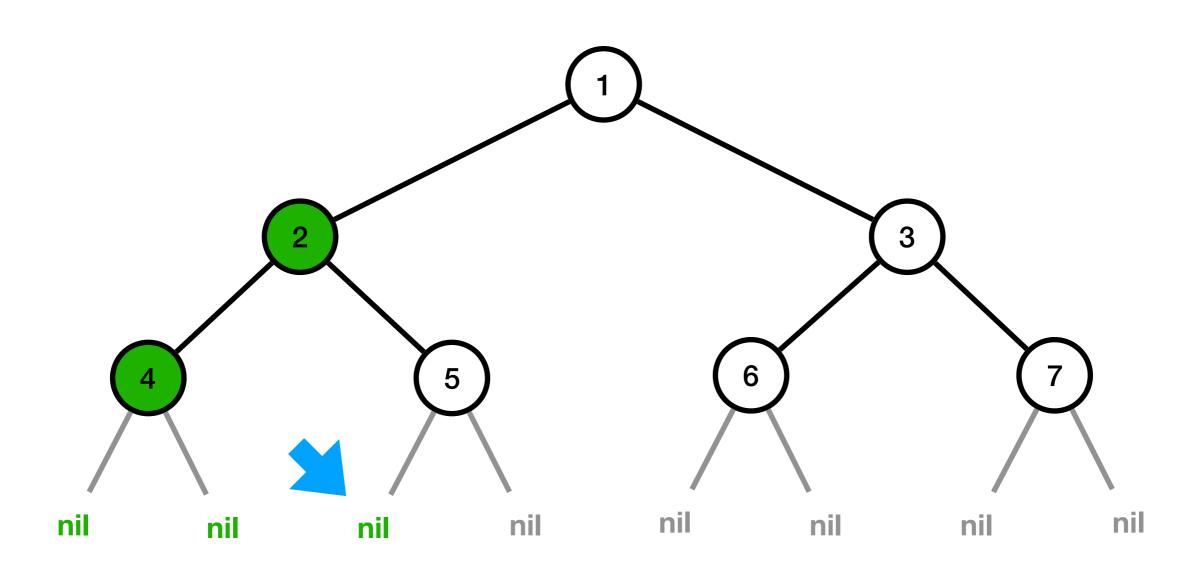




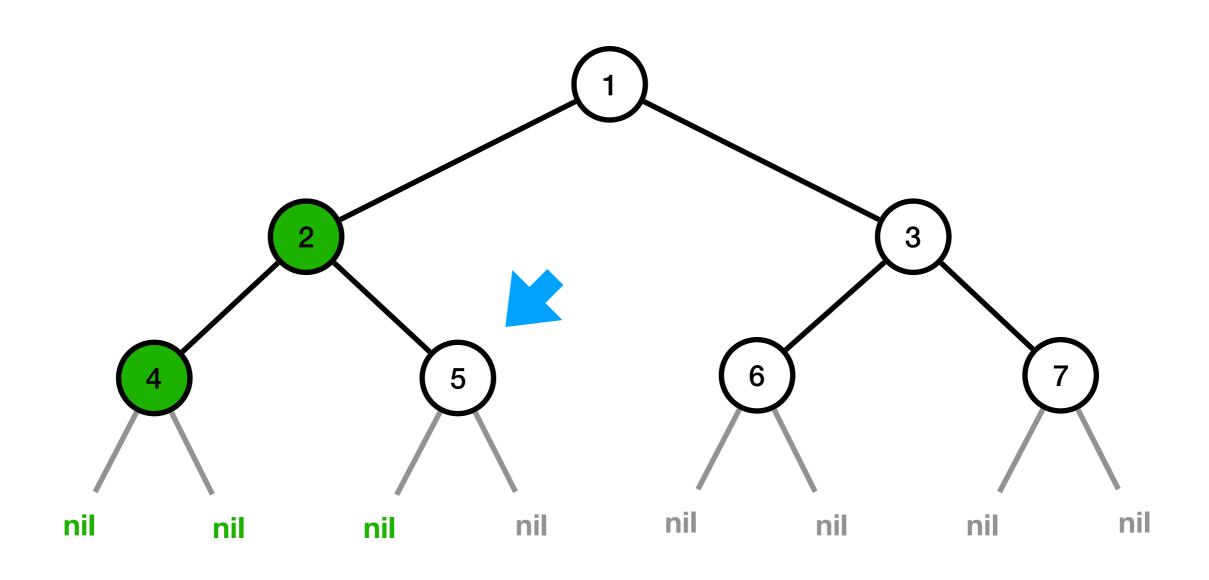






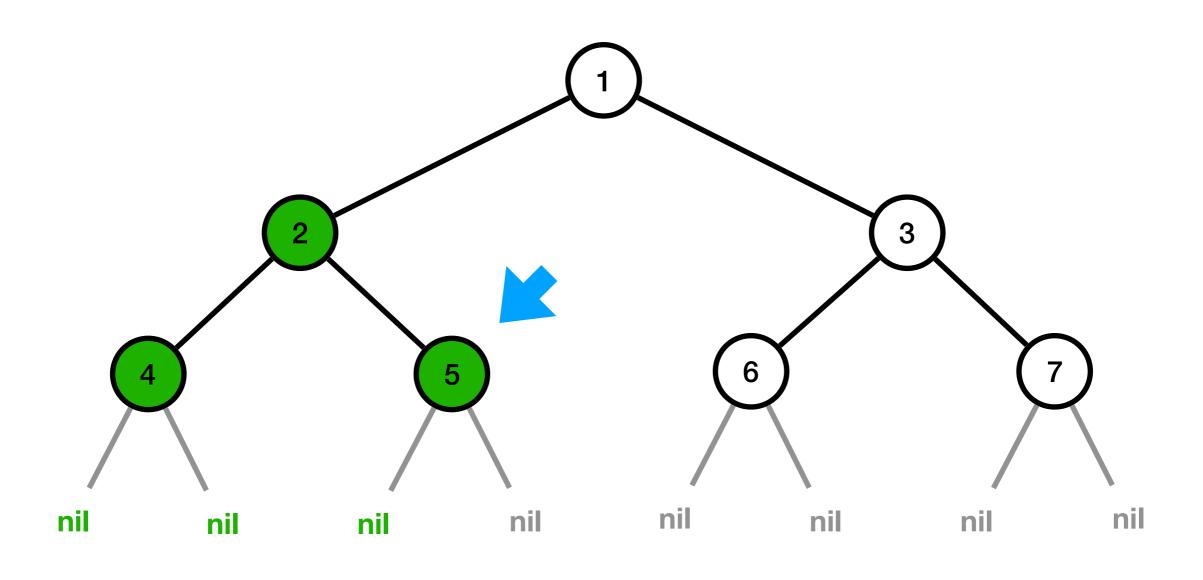


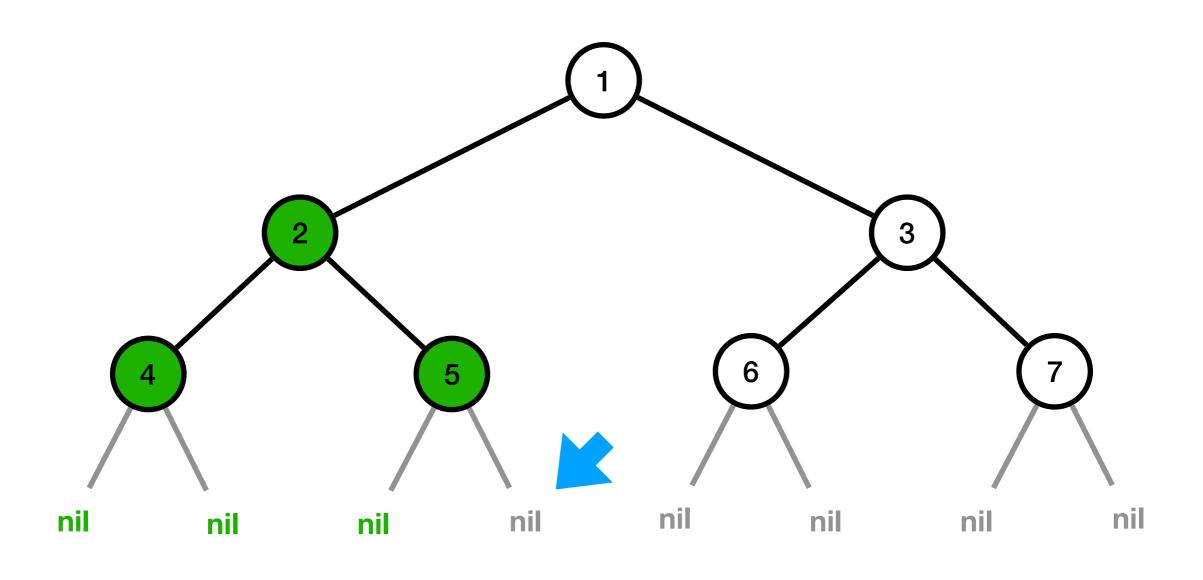


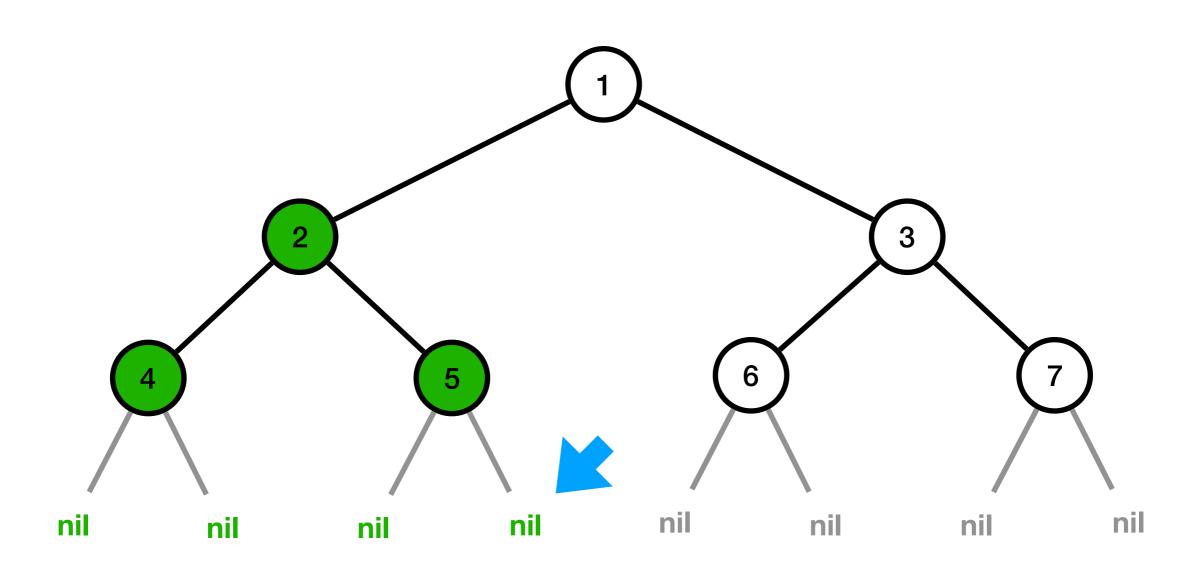


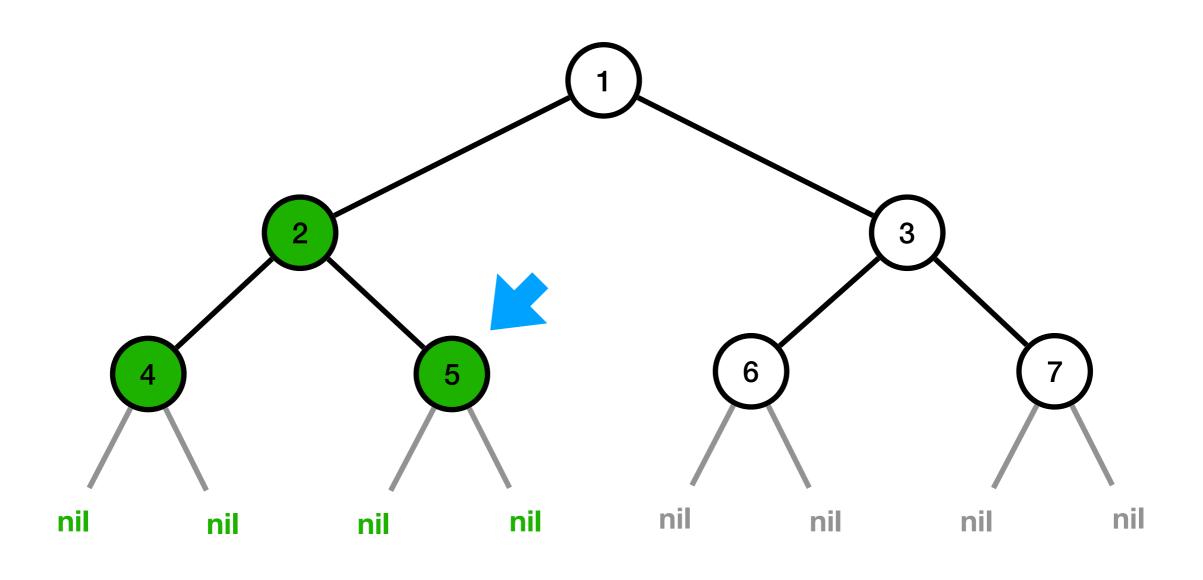


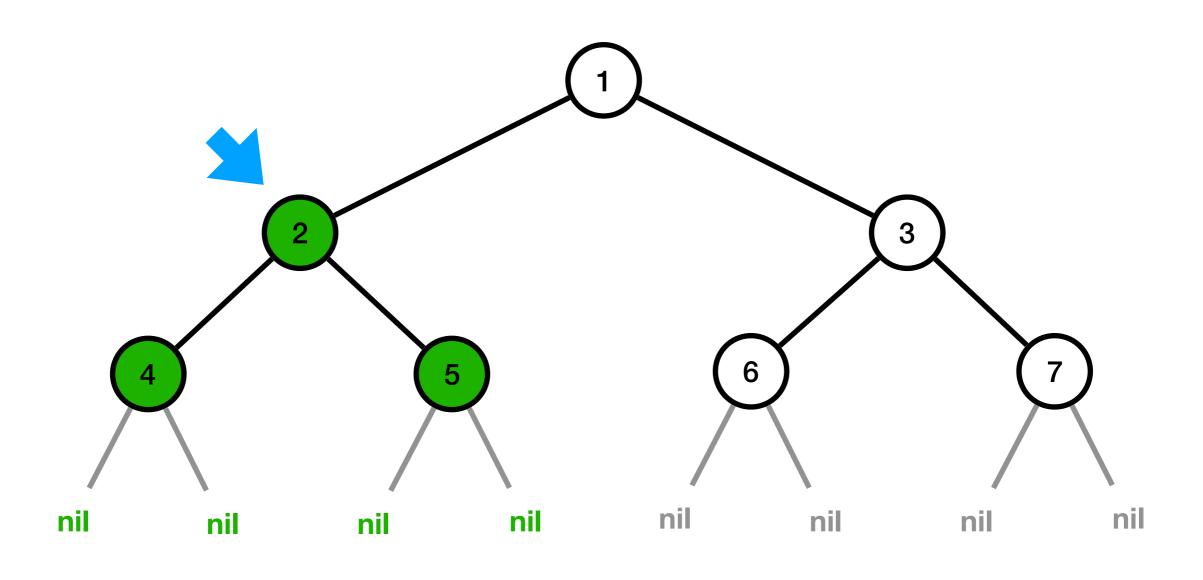


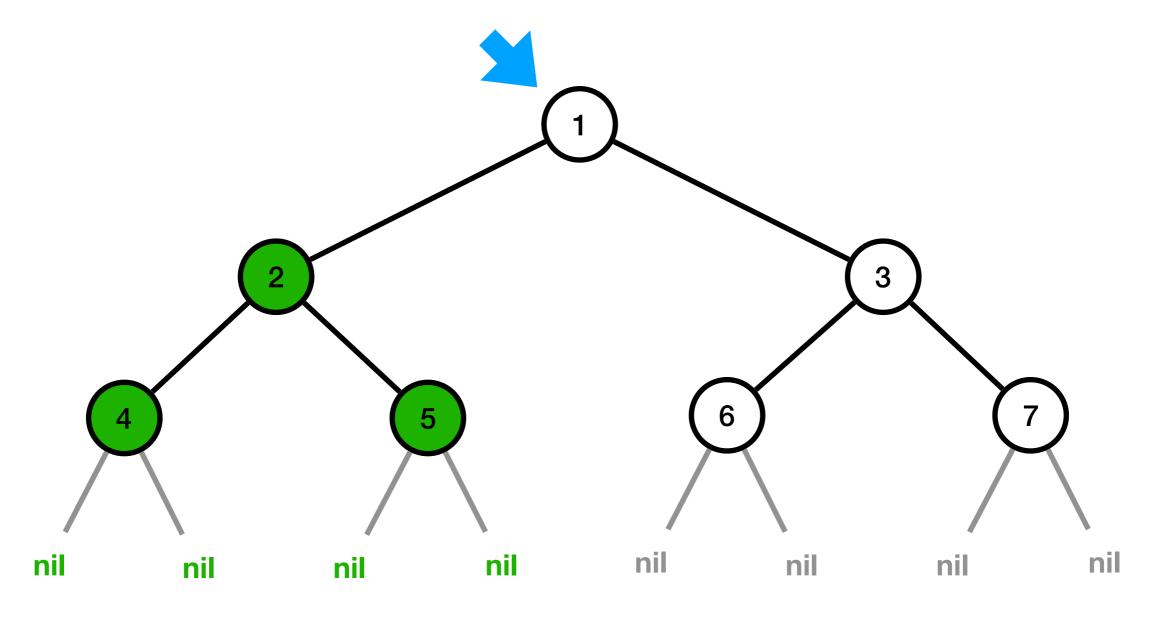








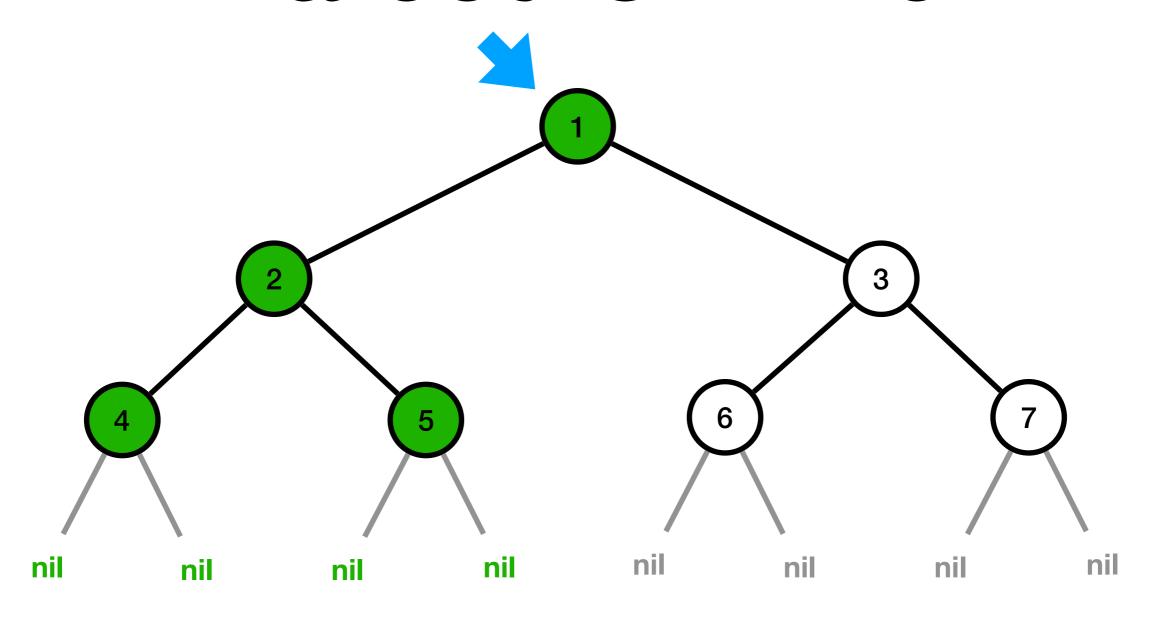


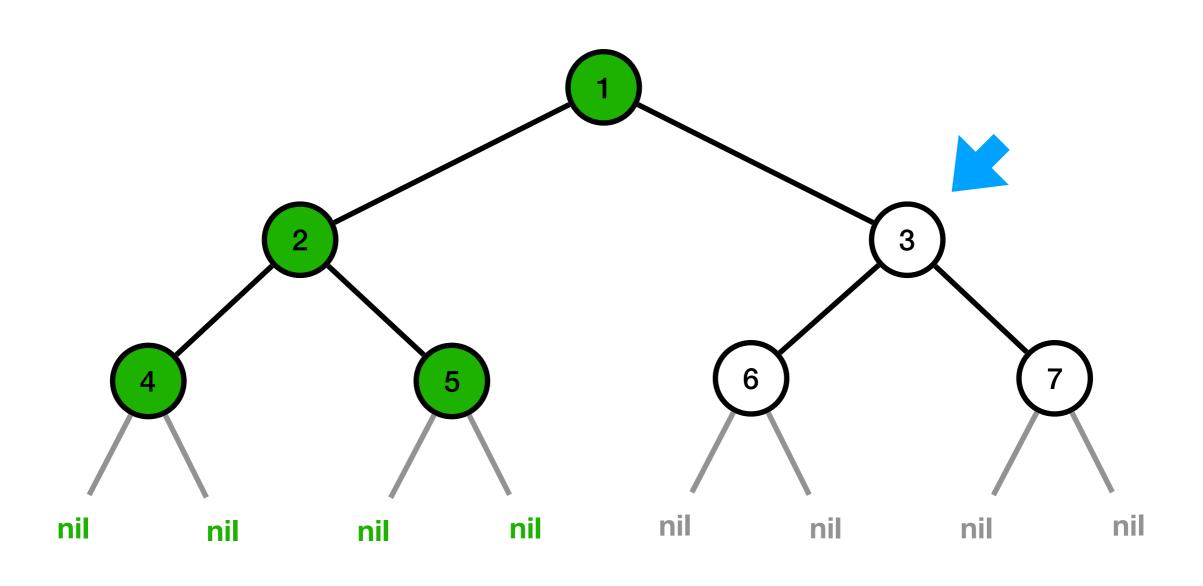


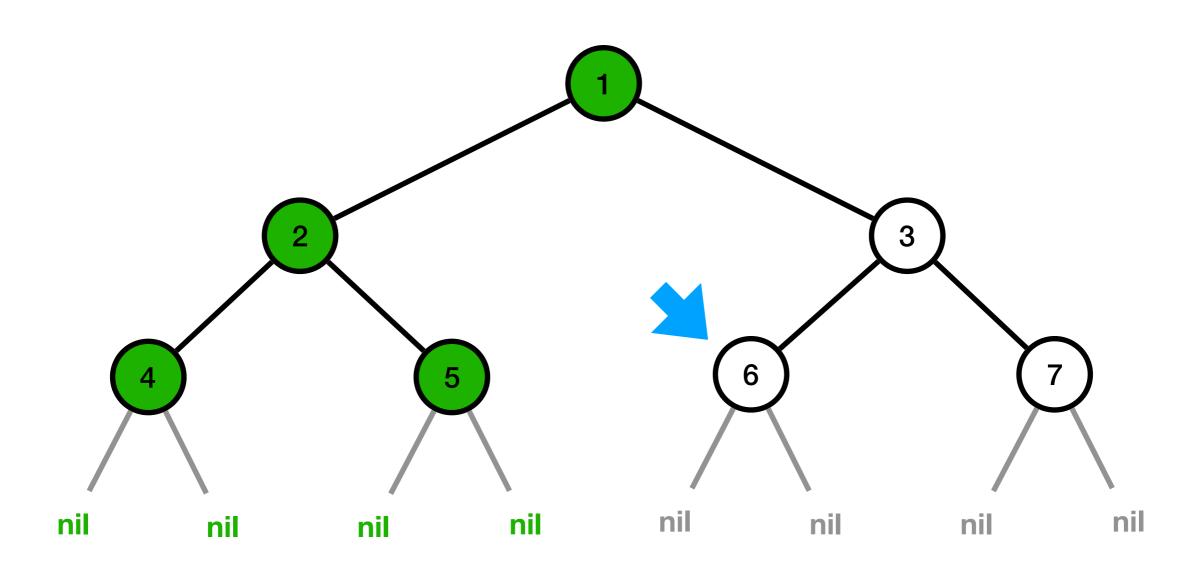


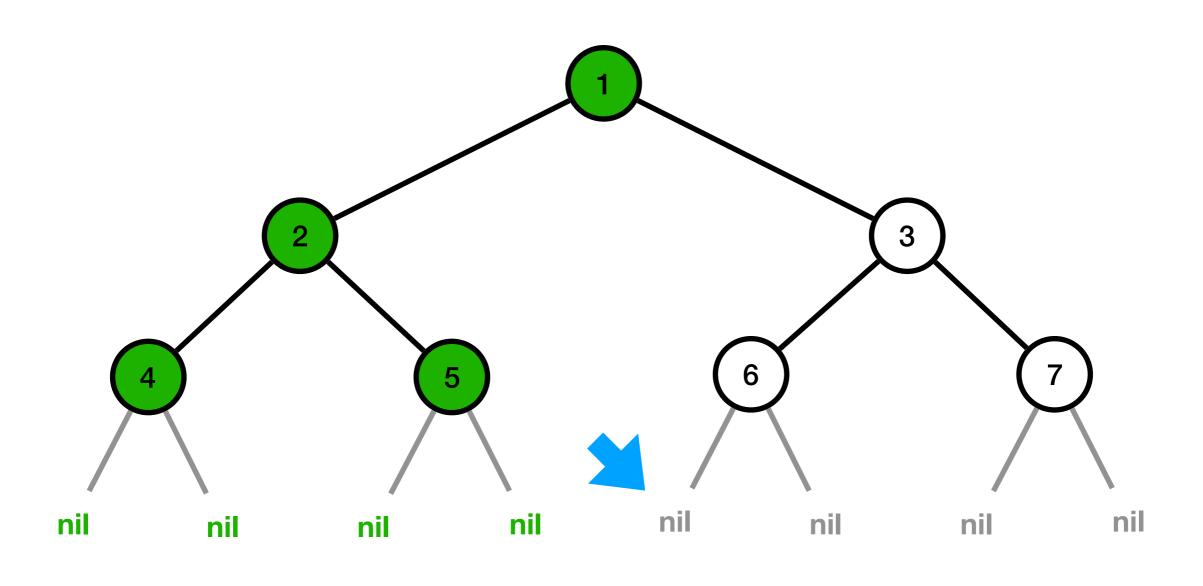


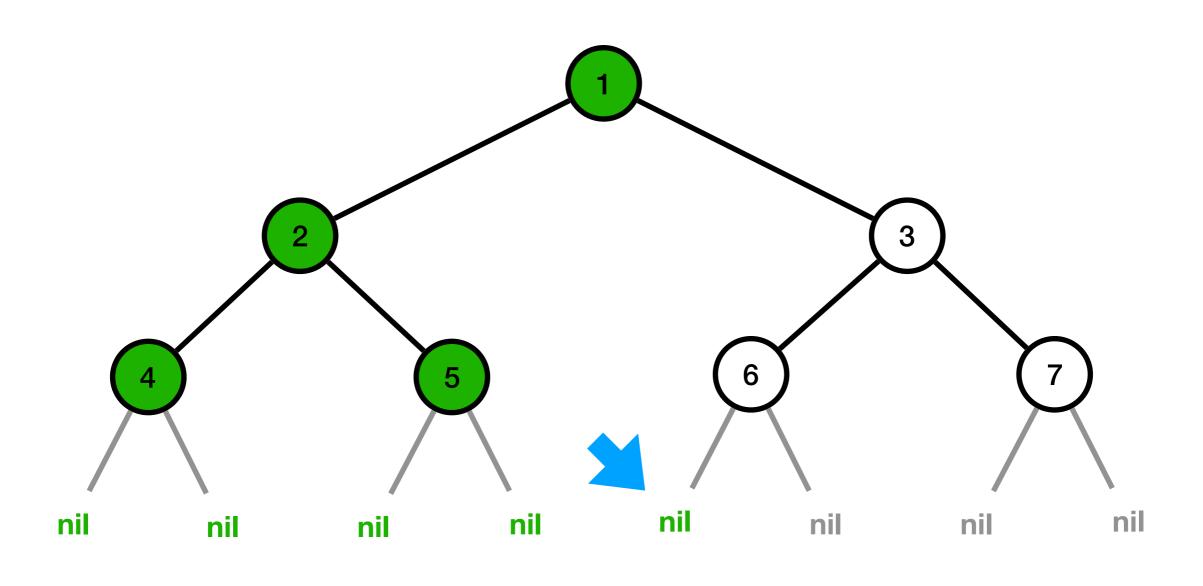


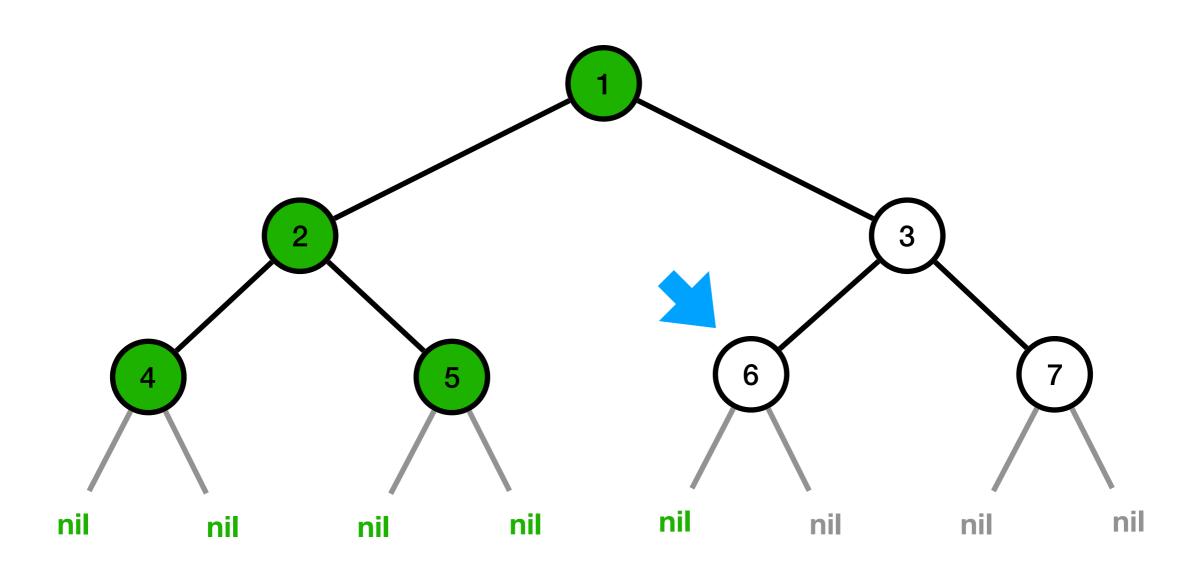


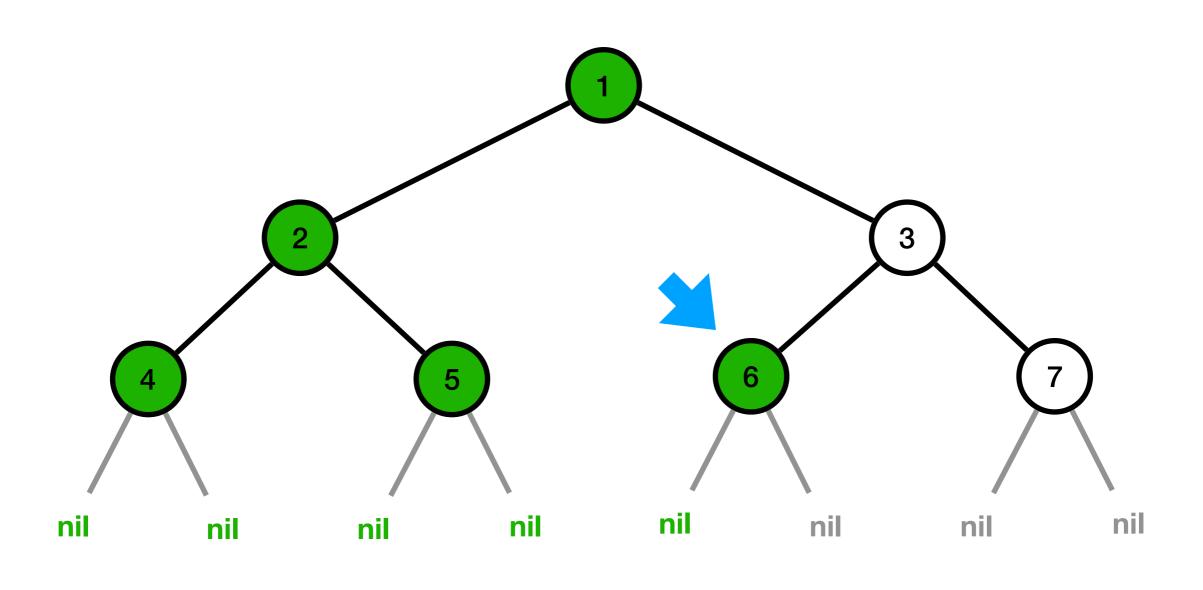


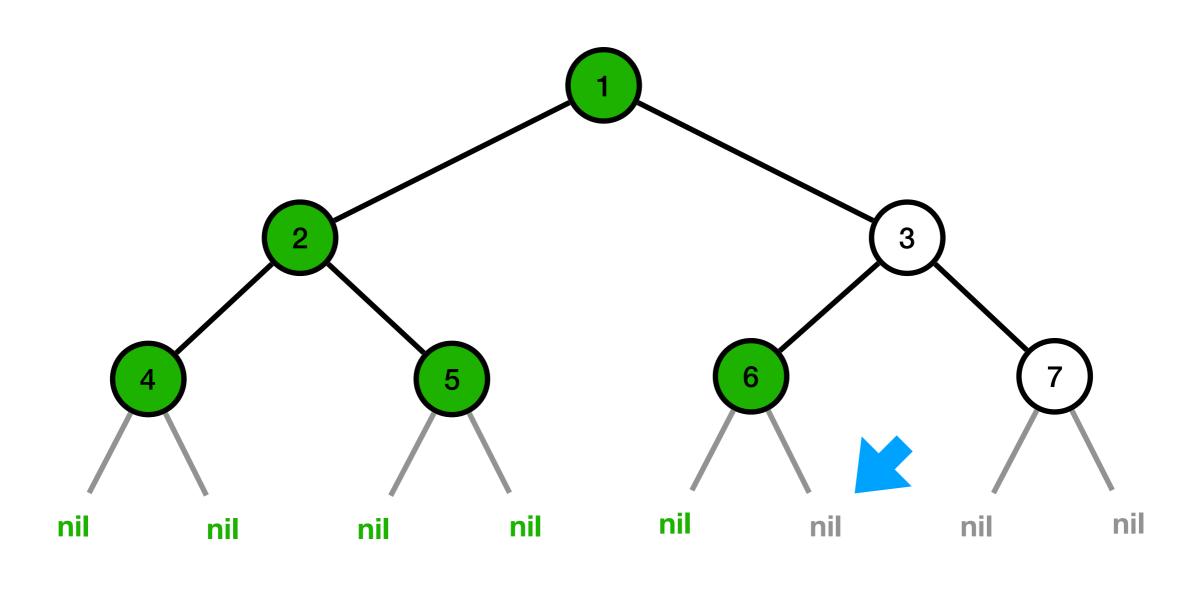


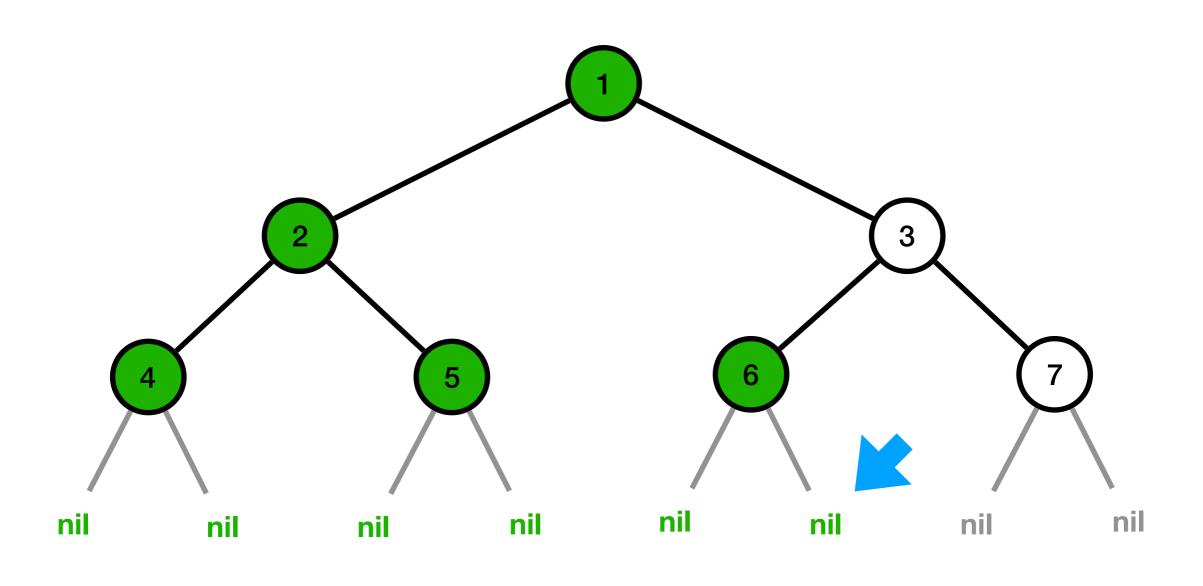


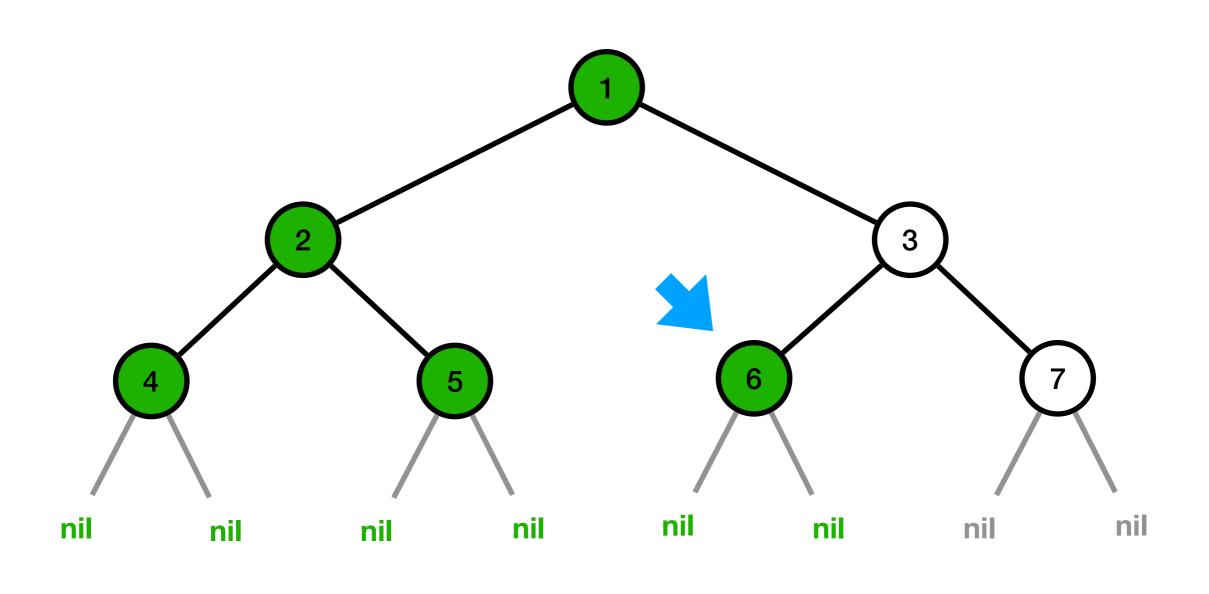


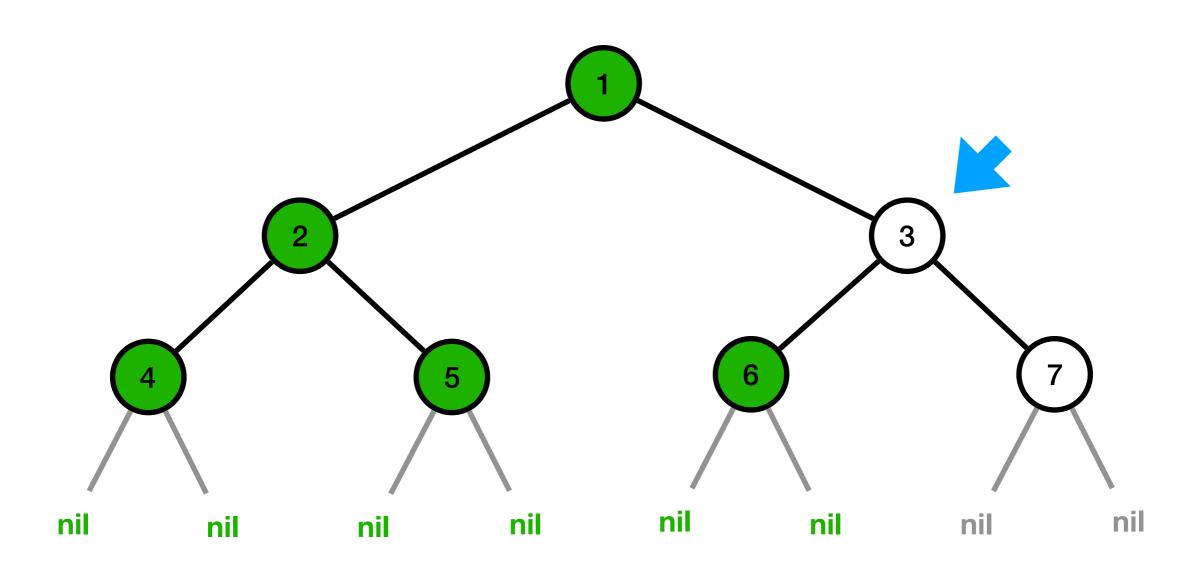


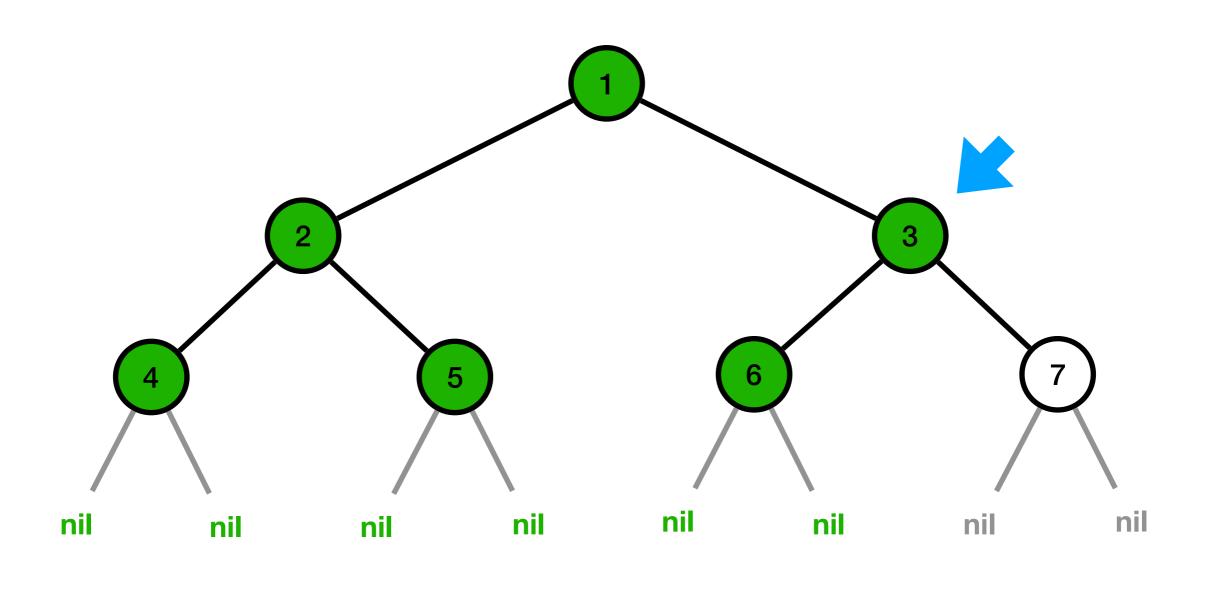




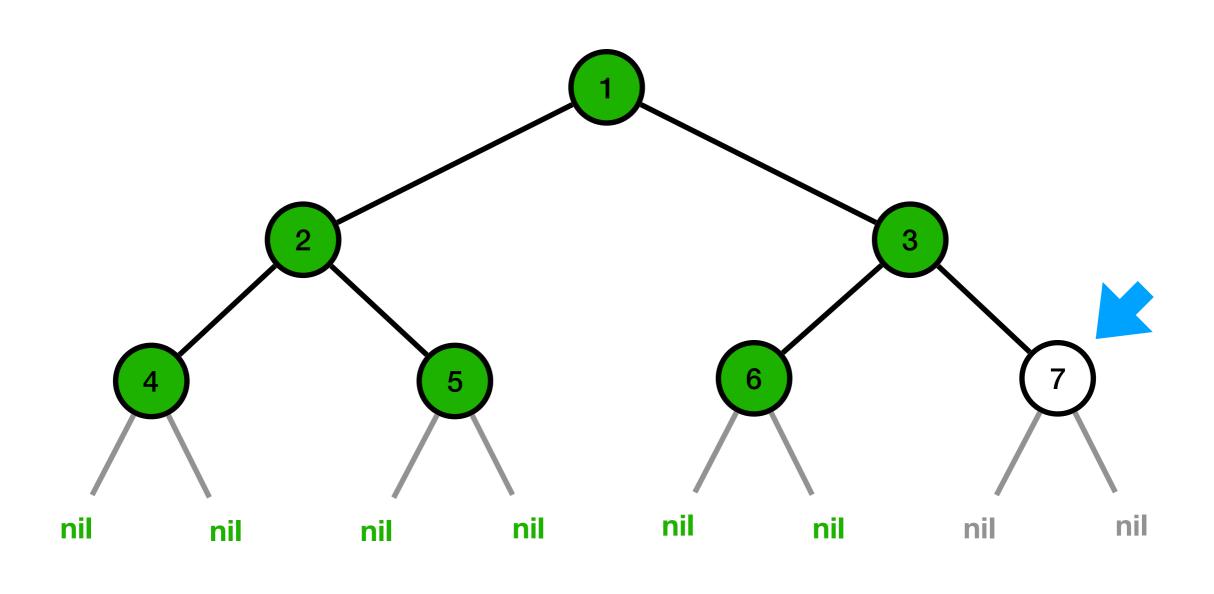




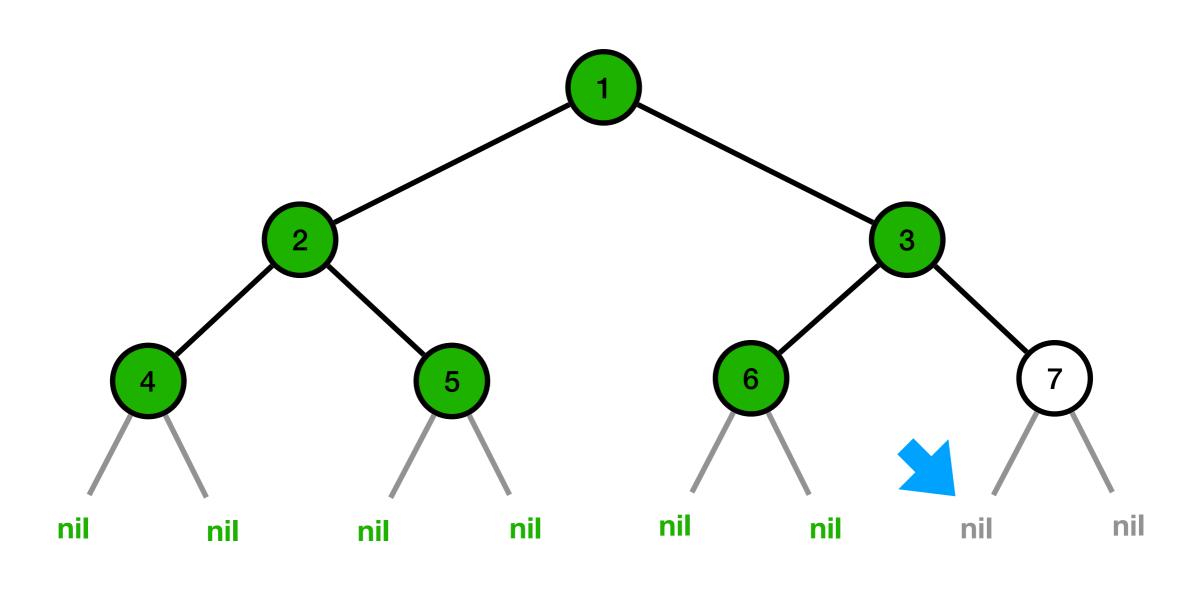




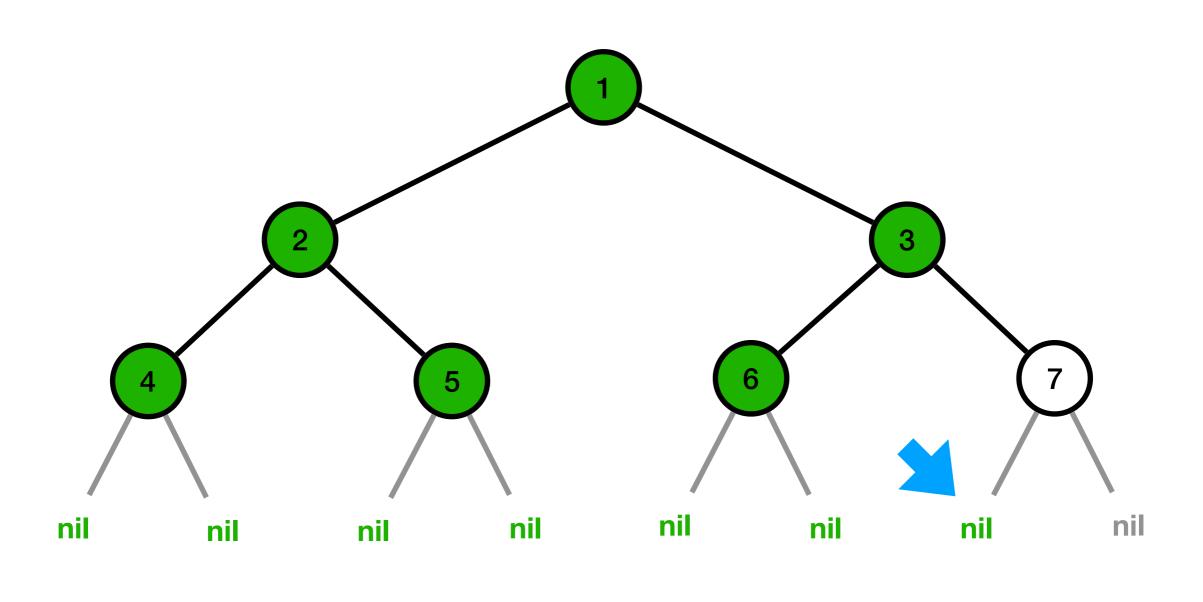
 $\left(5\right)$ $\left(1\right)$



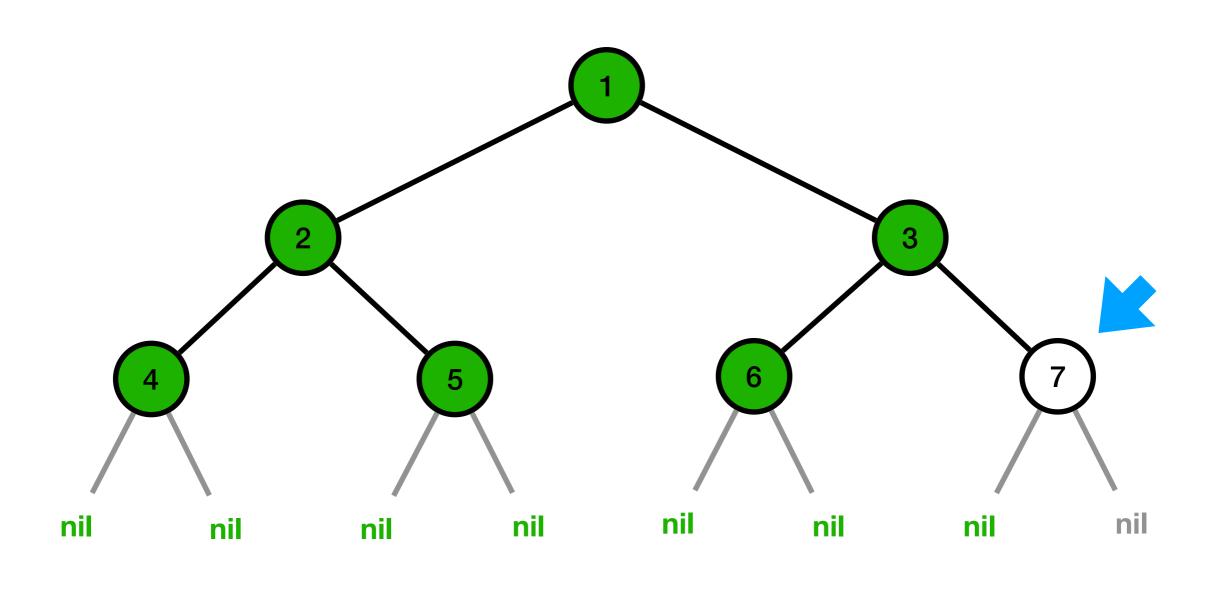
 $\left(5\right)$ $\left(1\right)$



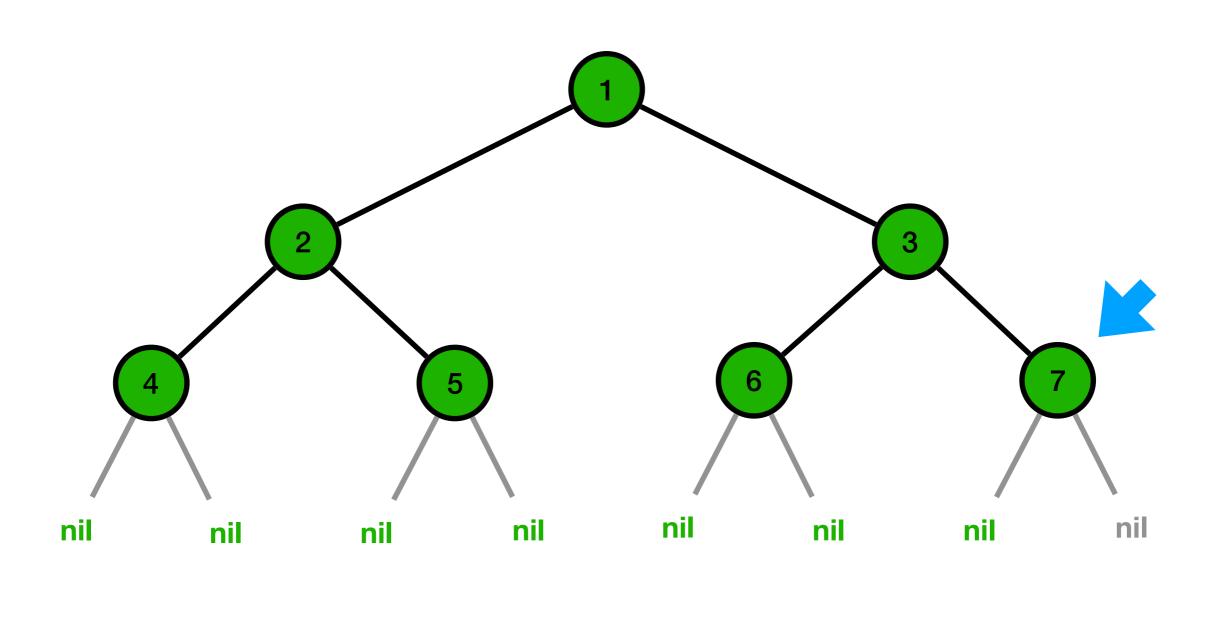
5 1



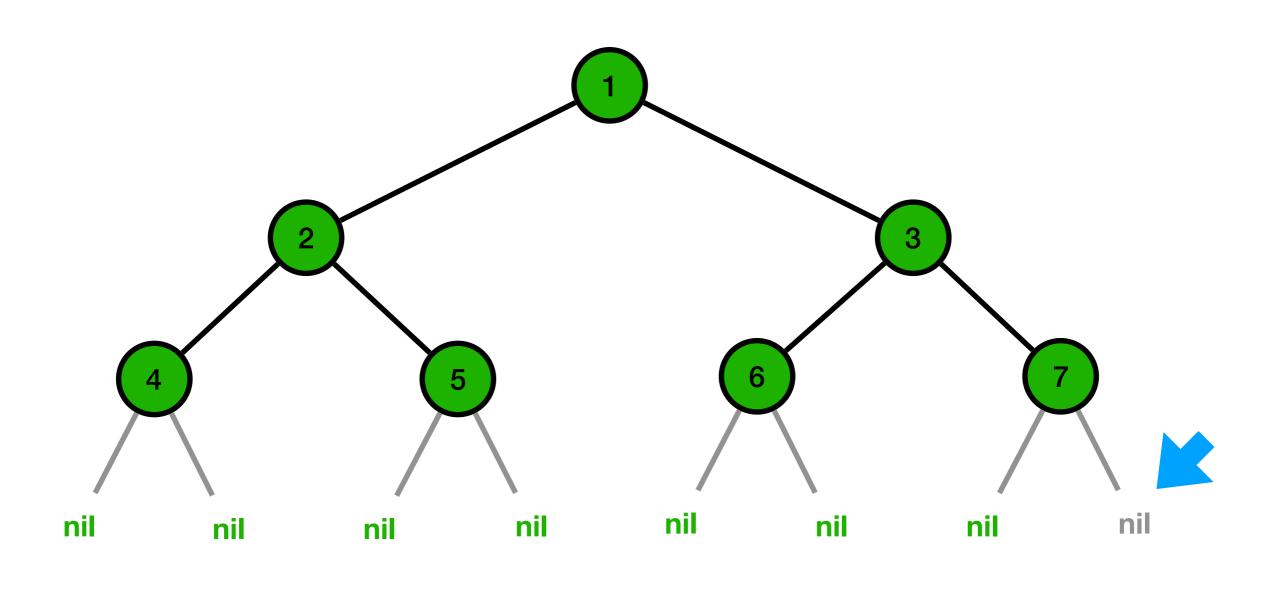
5 1

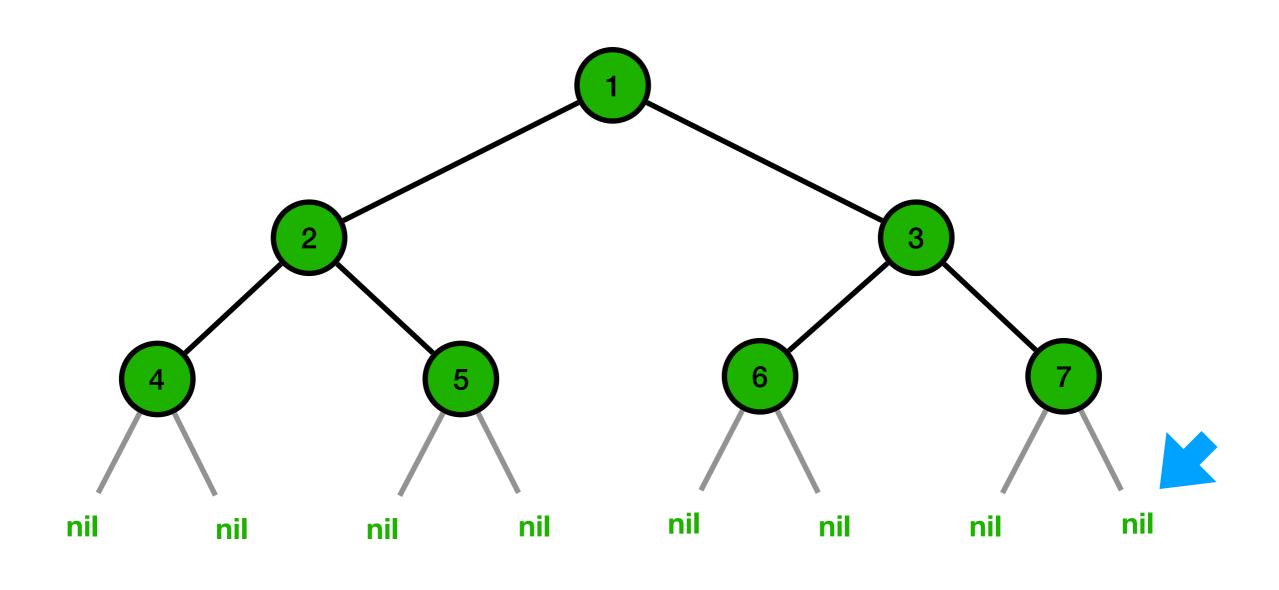


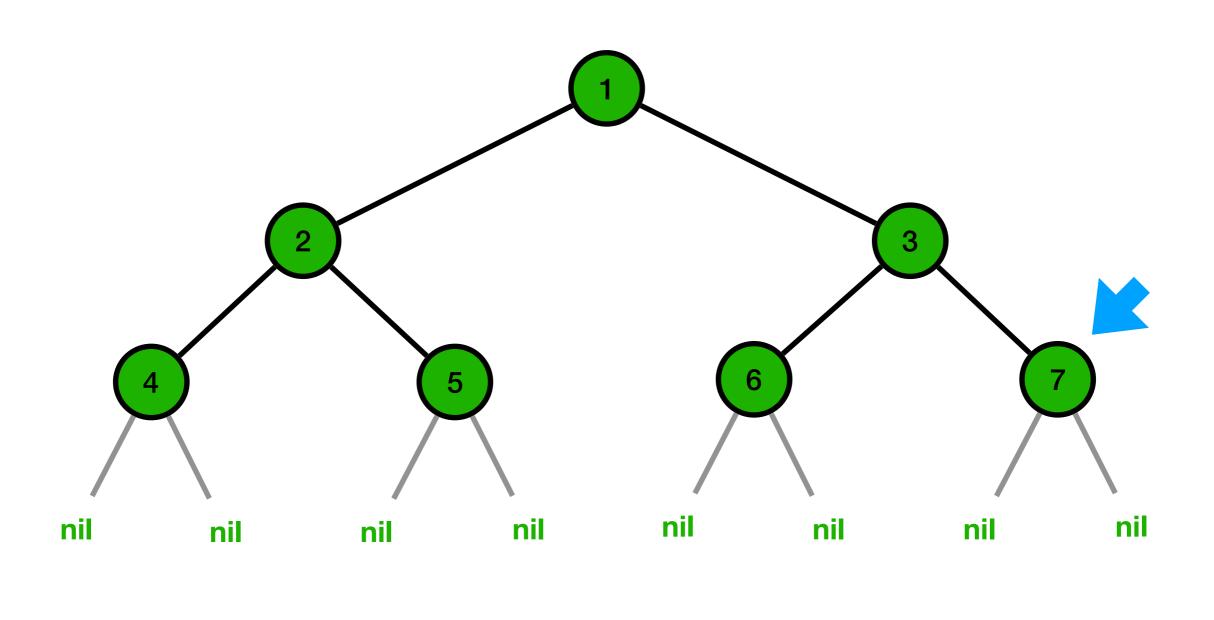
 $\left(5\right)$ $\left(1\right)$



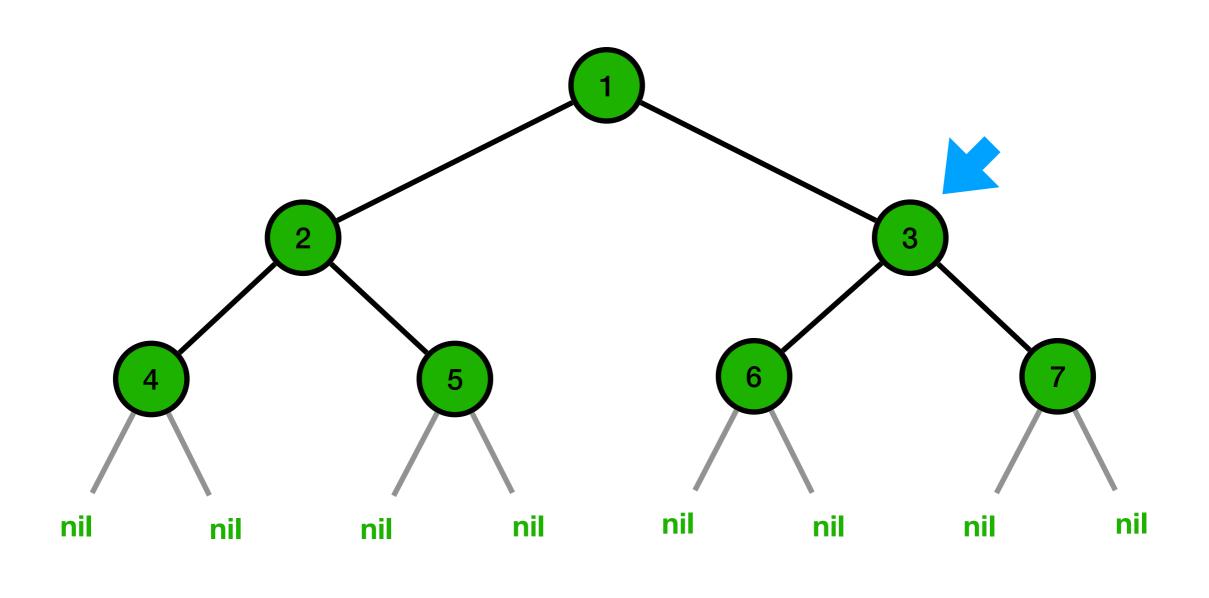
5 1 6



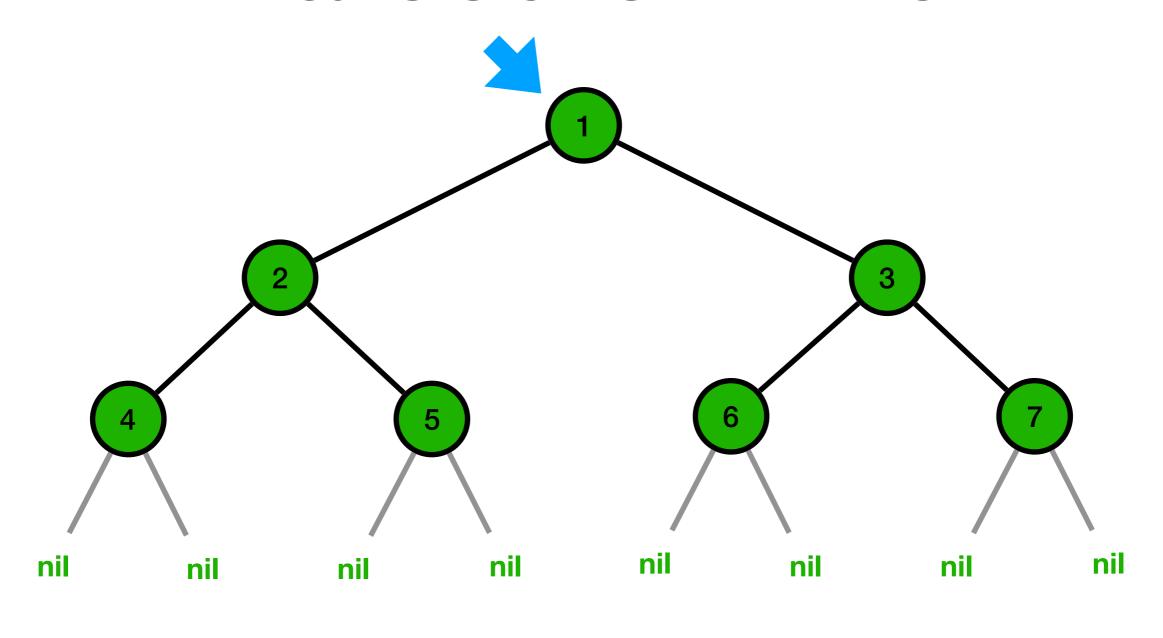


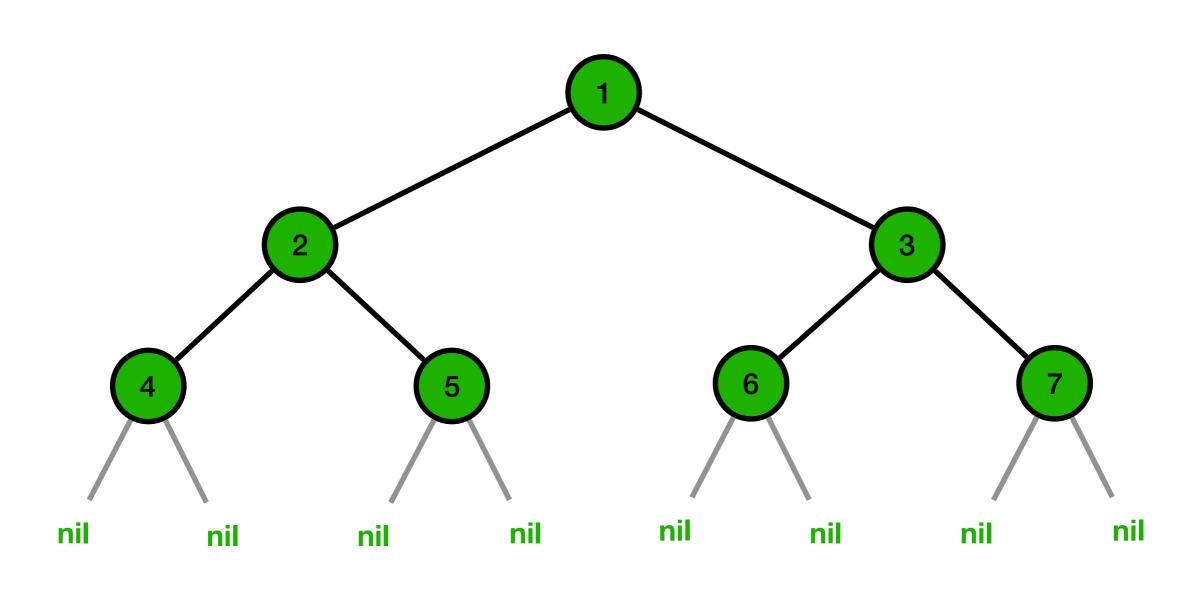


5 1 6



5 1 6





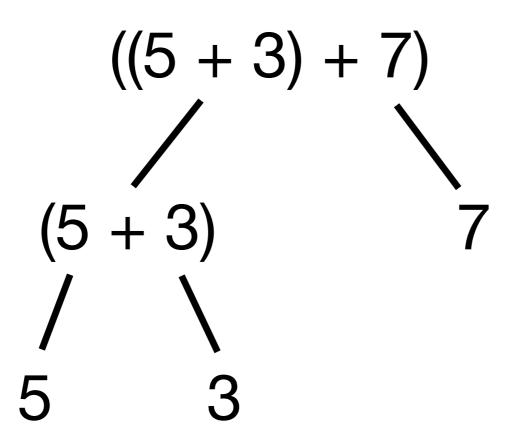
5 1 6

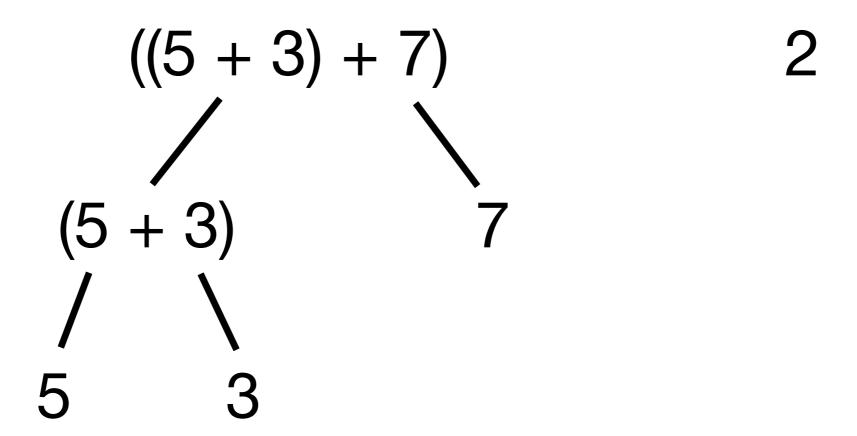
Expressions arithmétiques

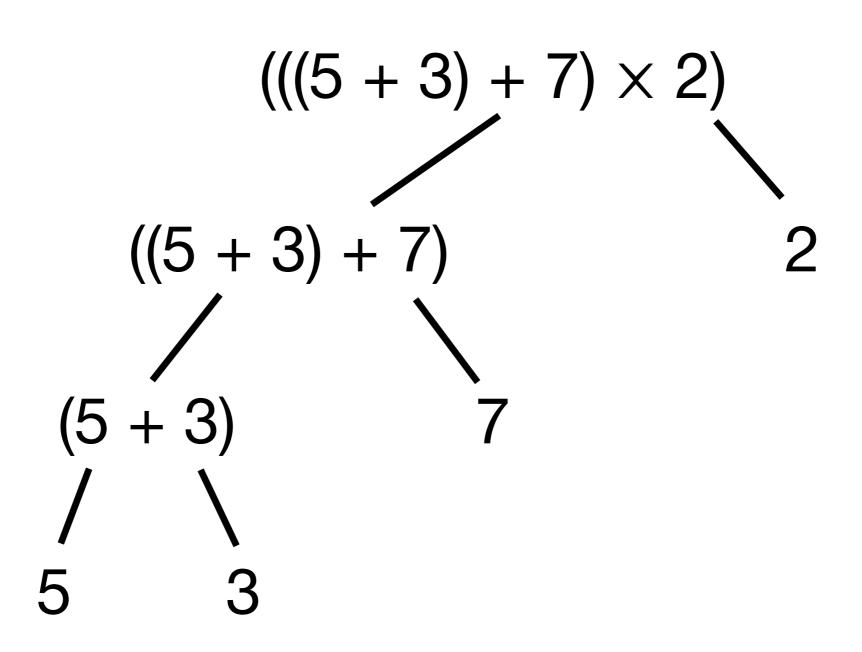
Une expression arithmétique (totalement parenthésée) est :

- soit un nombre entier n
- soit une somme parenthésée (e₁ + e₂), où e₁ et e₂
 sont deux sub-expressions (totalement parenthésée)
- soit une soustraction parenthésée (e₁ e₂)
- soit un produit parenthésé (e₁ × e₂)
- soit une division parenthésée (e₁ ÷ e₂)

5 3







5 3 7 2

$$(5 + 3)$$
 $(7 + 2)$ 1 1 1 1 1 2 2 2

$$((5 + 3) \times (7 + 2))$$

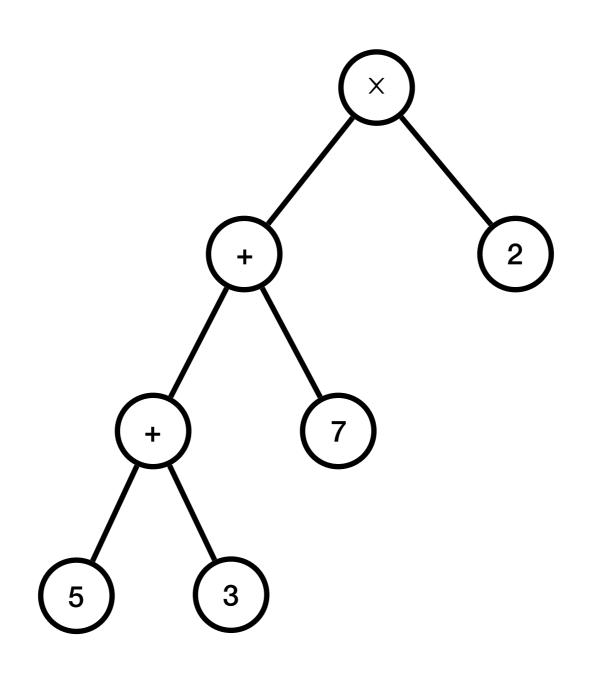
 $(5 + 3)$
 $(7 + 2)$
 $(5 + 3)$
 $(7 + 2)$
 $(5 + 3)$
 $(7 + 2)$
 $(5 + 3)$
 $(7 + 2)$
 $(7 + 2)$
 $(7 + 2)$

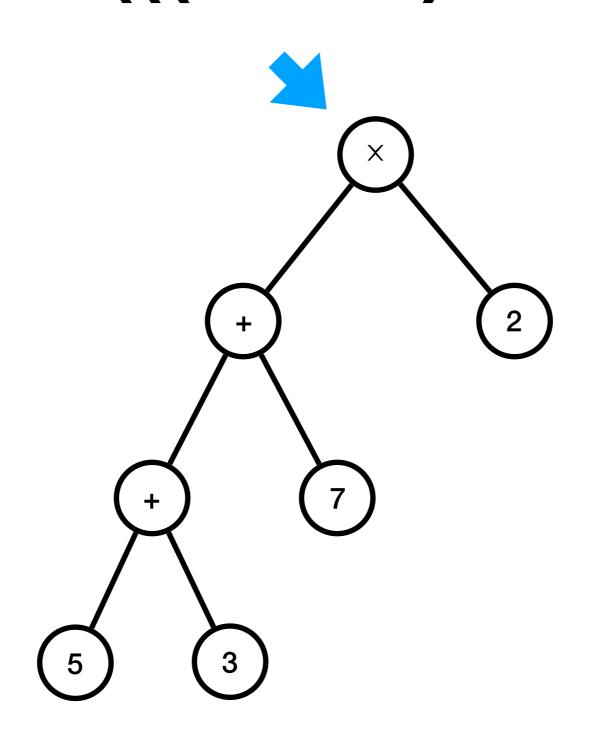
$$(((5+3)+7)\times 2)$$
 $((5+3)+7)\times 2$
 $((5+3)+7)\times 2$
 $(5+3)\times 7$
 $(5+3)\times 7$
 $(5+3)\times 7$
 $(5+3)\times 7$
 $(5+3)\times 7$
 $(5+3)\times 7$

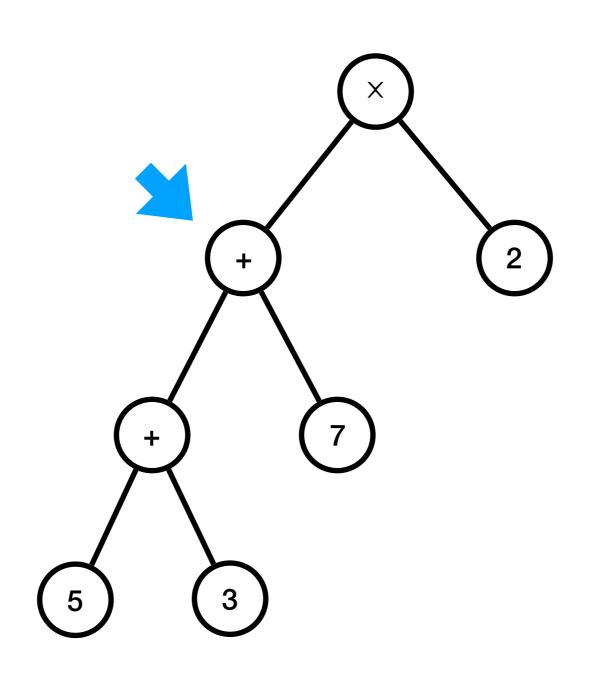
$$((5 + 3) \times (7 + 2))$$

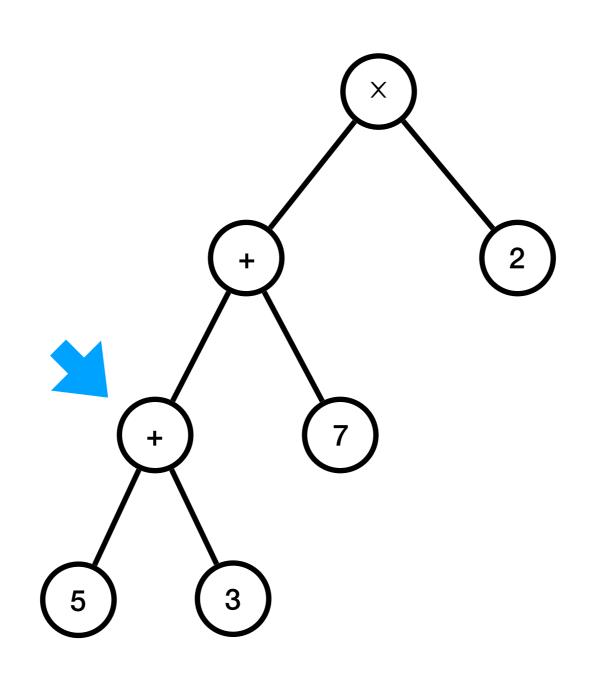
 $(5 + 3) \times (7 + 2)$
 $(5 + 3) \times (7 + 2)$

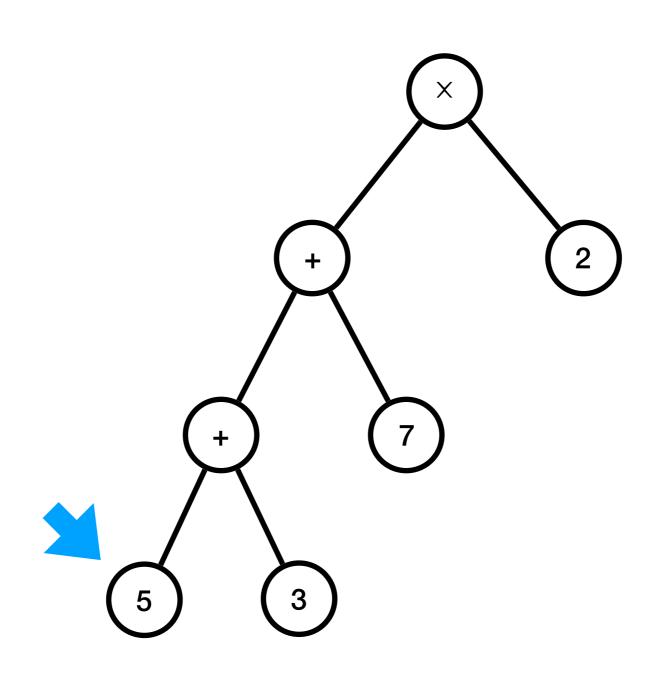
$$((5+3) \times (7+2))$$
 $(5+3) \times (7+2)$
 $(7+2) \times (7$

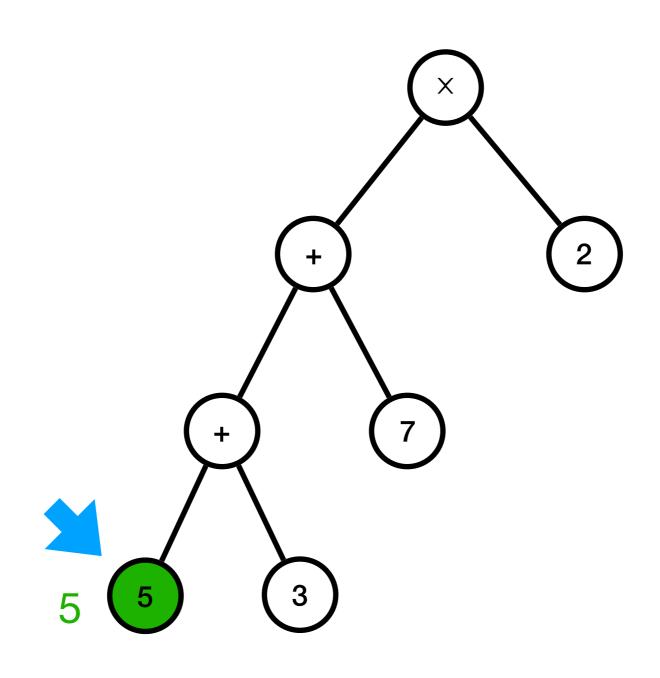


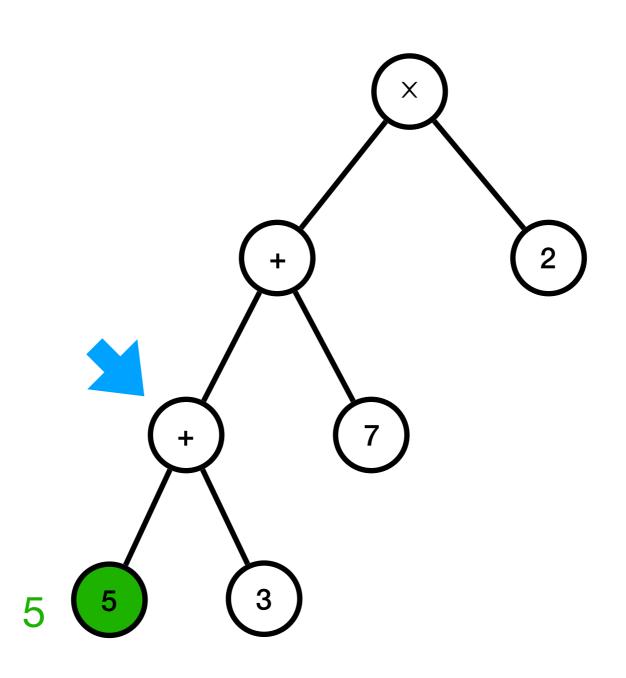


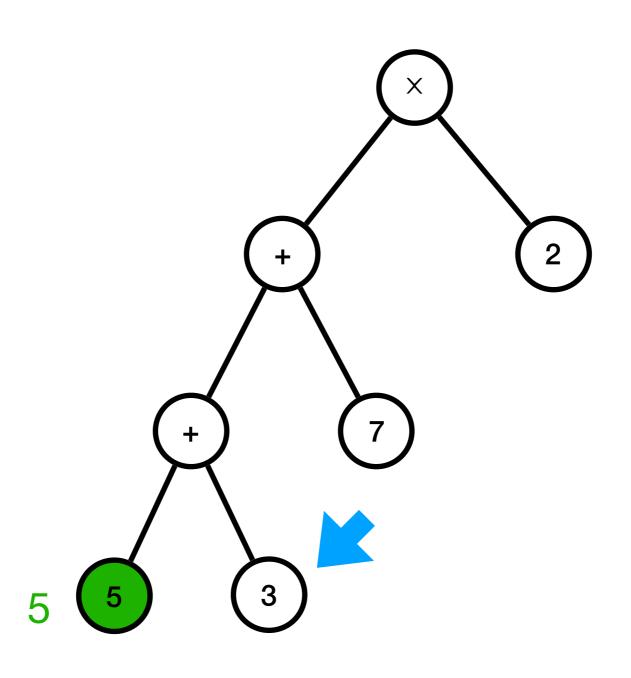


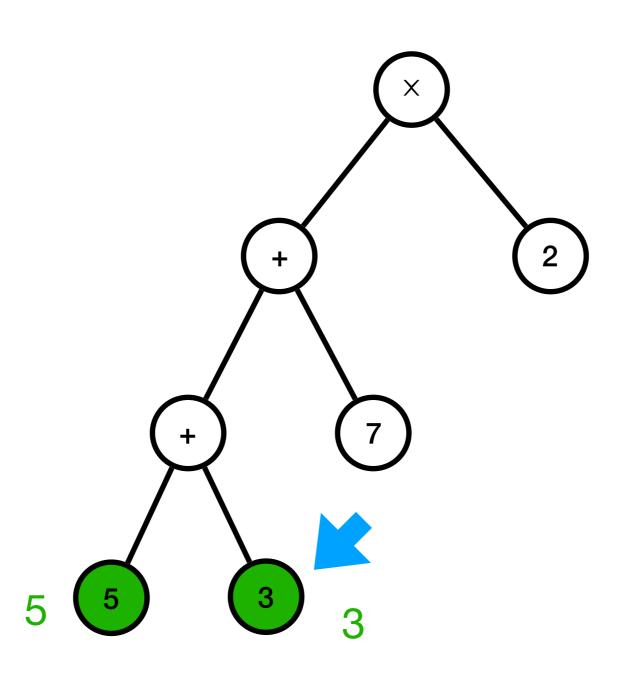


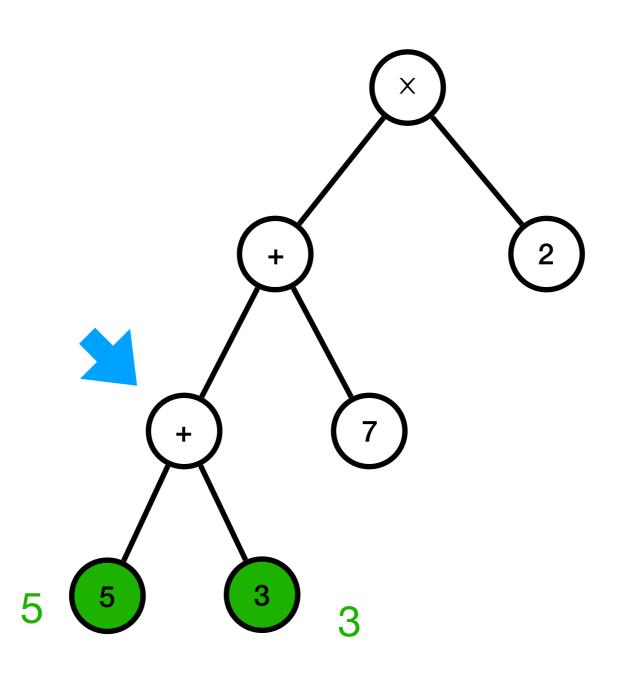


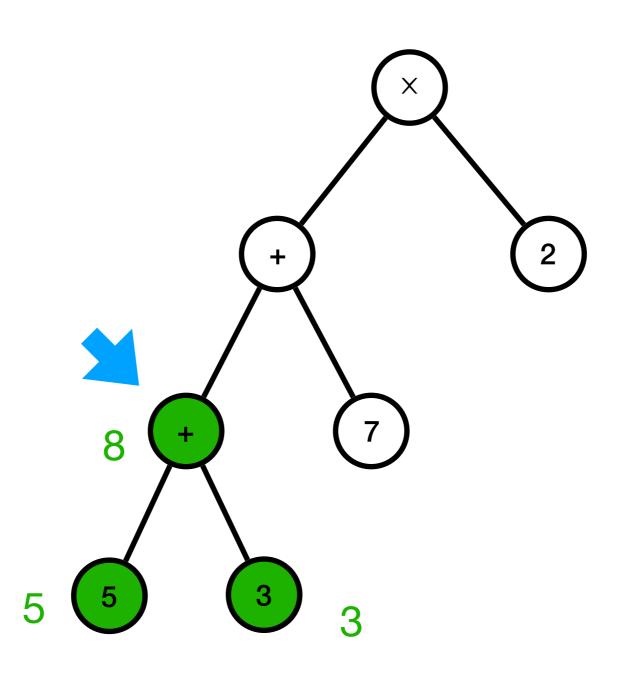


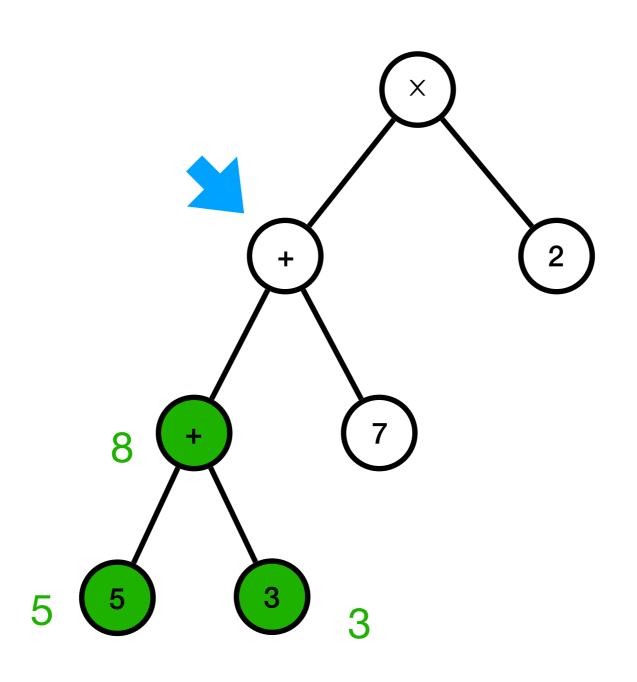


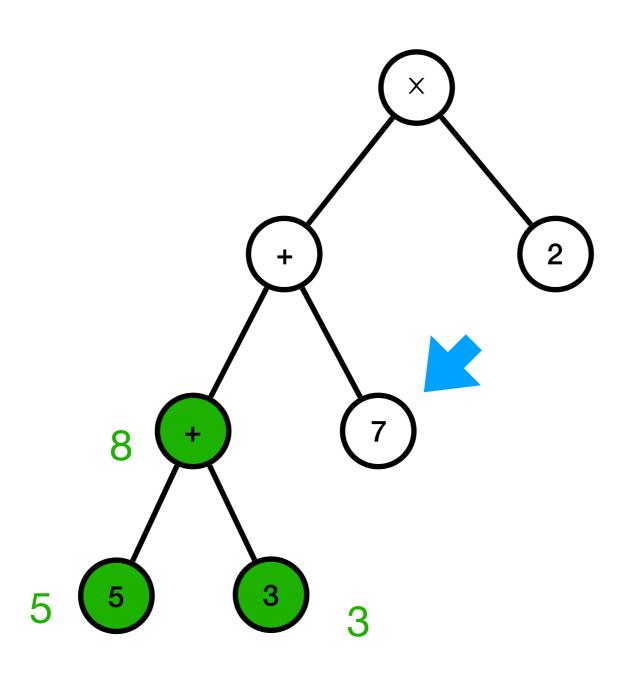


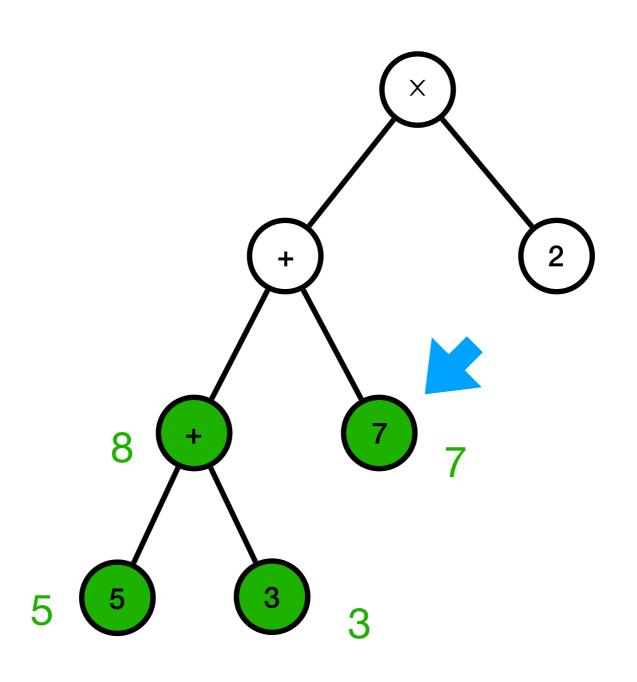


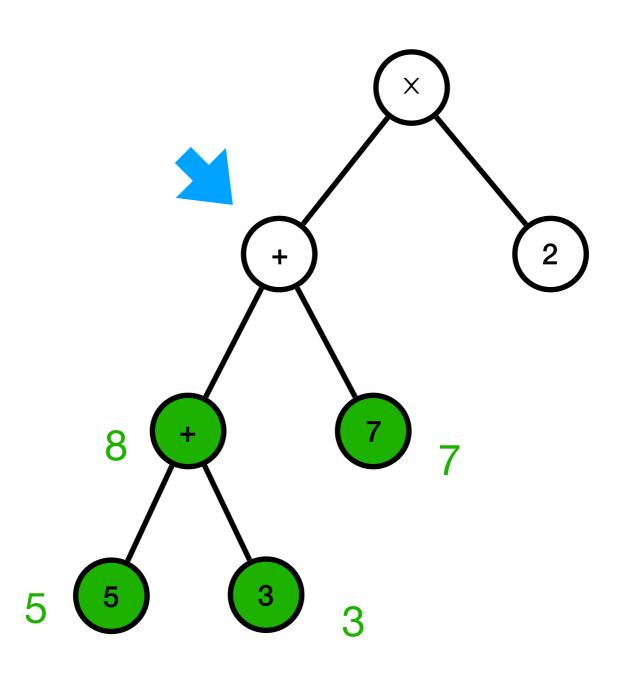


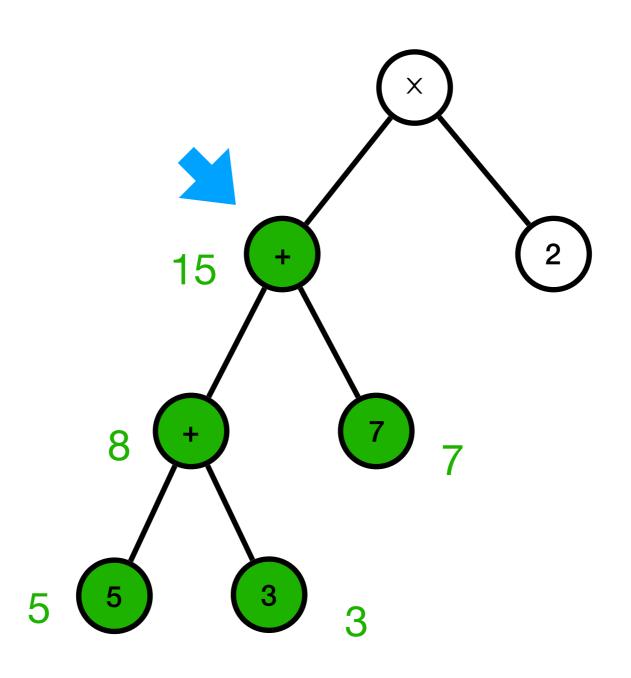


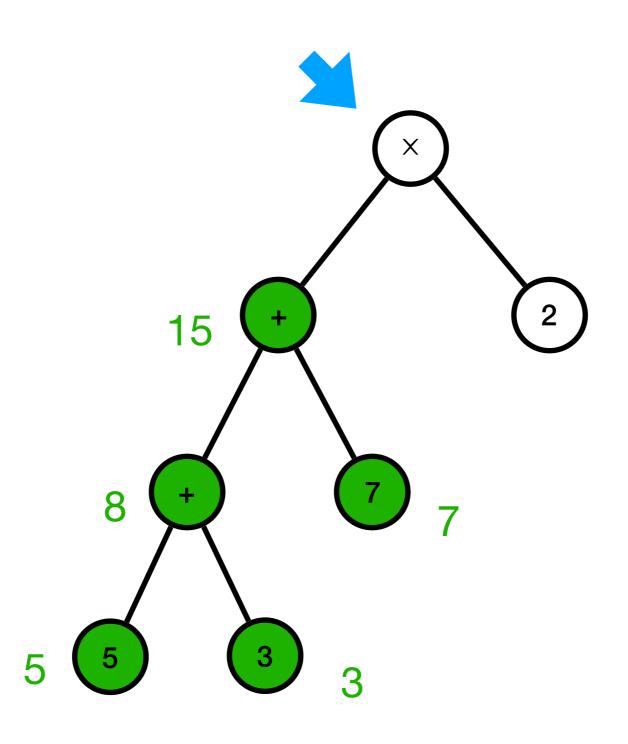


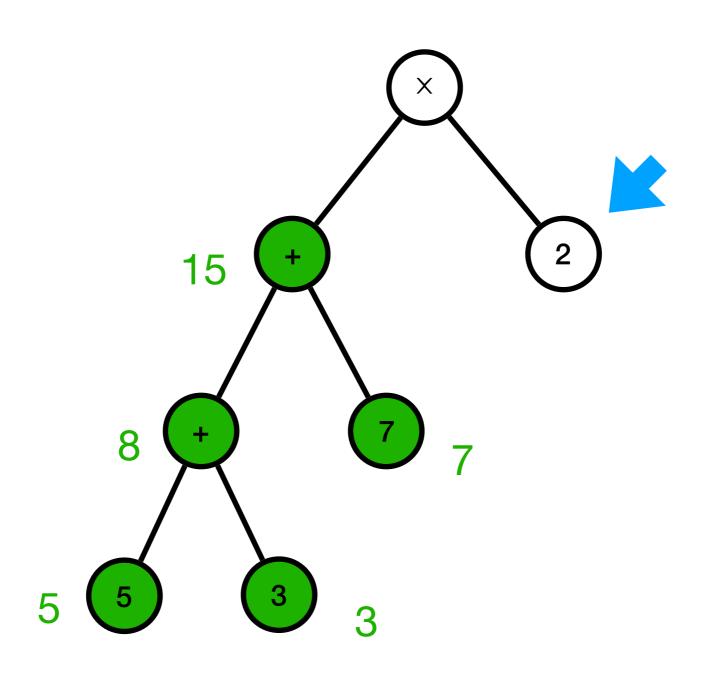


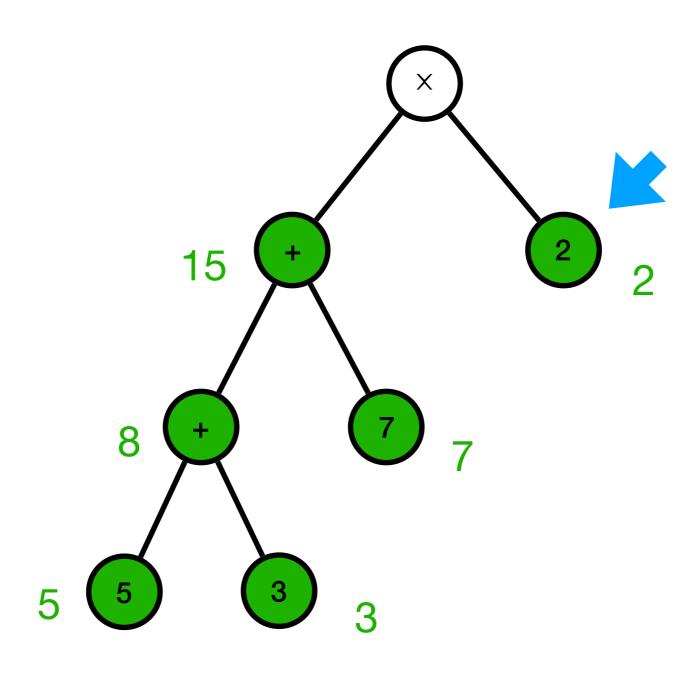


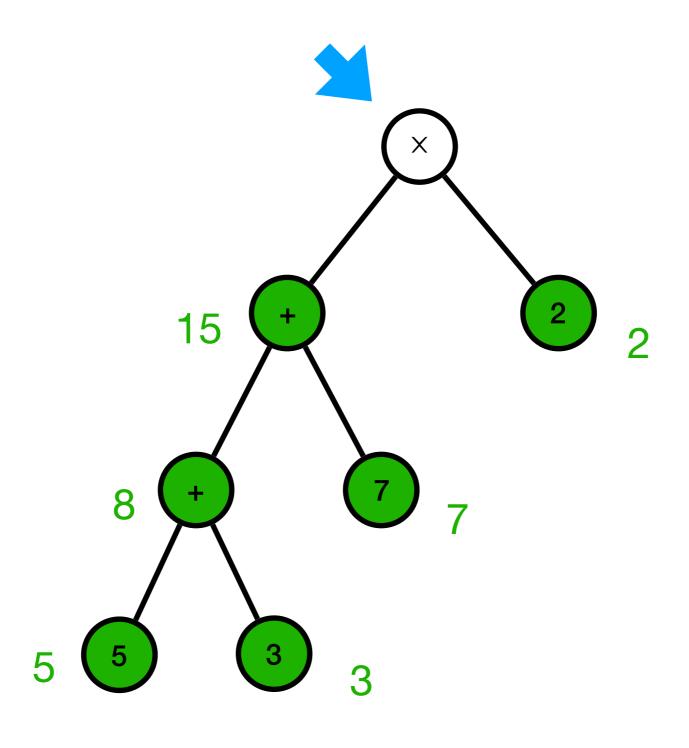


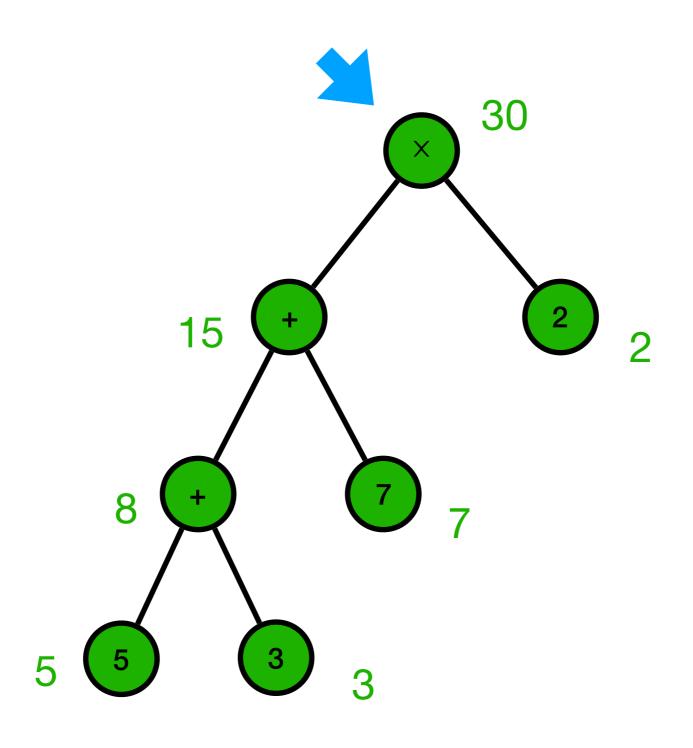


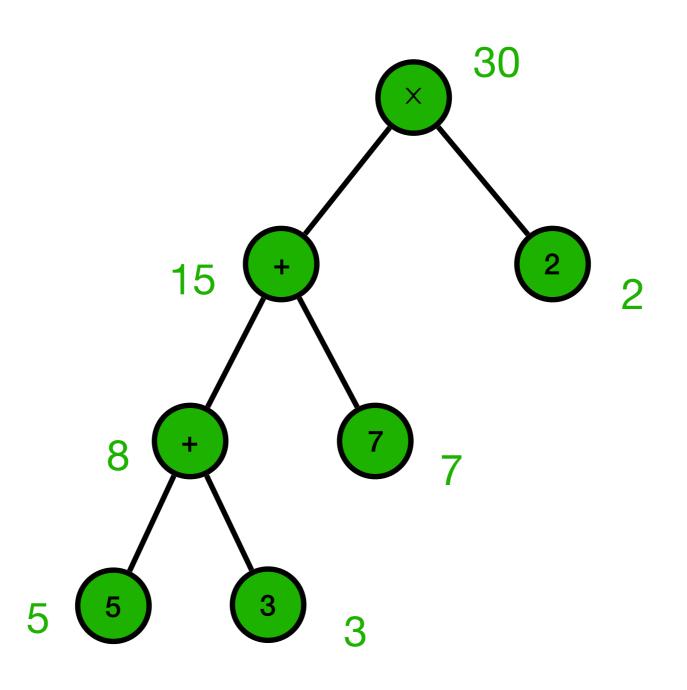












Évaluer une expression

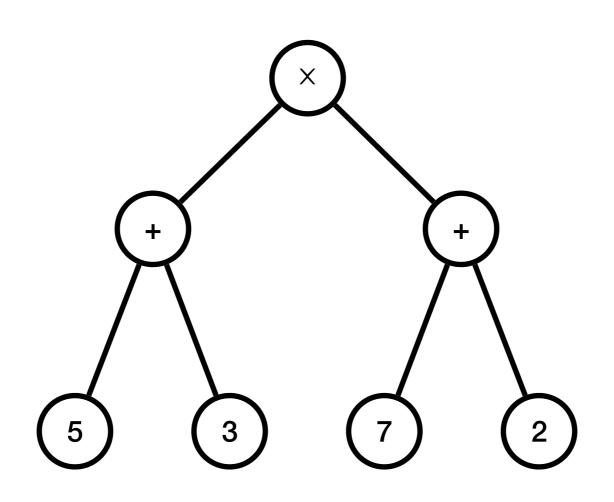
```
fonction évaluer-expression(nœud)
   si valeur[nœud] est un nombre alors
      retourner valeur[nœud]
   sinon
      m = évaluer-expression(gauche[nœud])
      n = évaluer-expression(droite[nœud])
      si valeur[nœud] = "+" alors
         retourner m + n
      sinon si valeur[nœud] = "-" alors
         retourner m - n
      sinon si valeur[nœud] = ... (etc.)
```

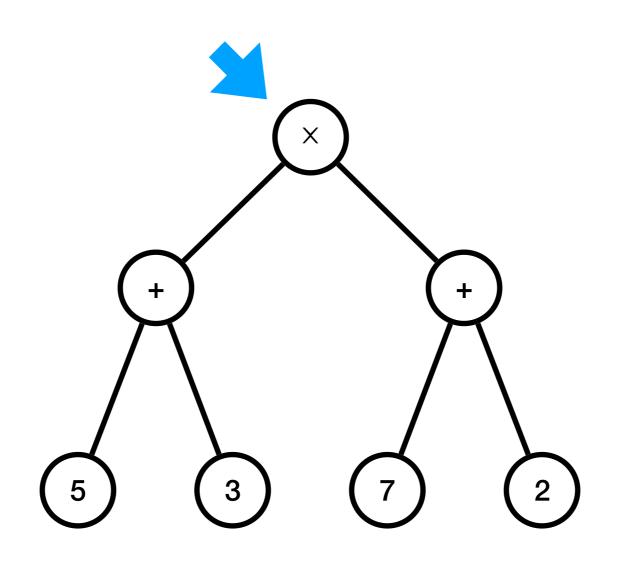
Évaluer une expression

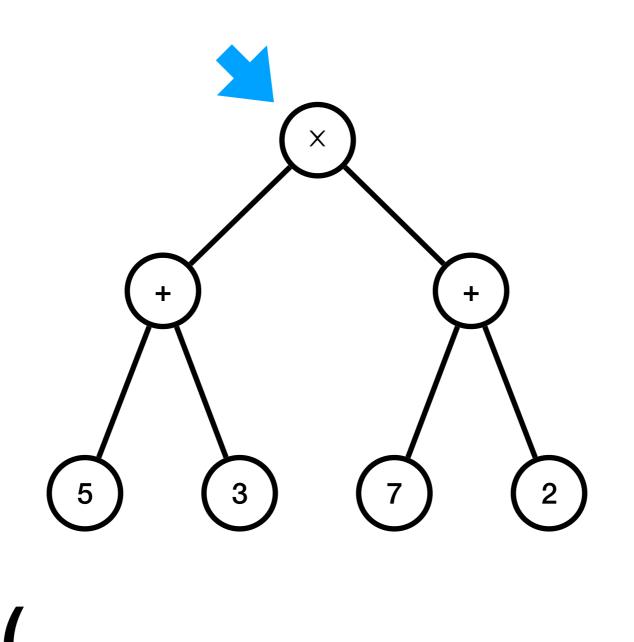
```
si valeur[nœud] est un nombre alors
   retourner valeur[nœud]
sinon
   m = évaluer-expression(gauche[nœud])
   n = évaluer-expression(droite[nœud])
   si valeur[nœud] = "+" alors
      retourner m + n
   sinon si valeur[nœud] = "-" alors
      retourner m - n
   sinon si valeur[nœud] = ... (etc.)
```

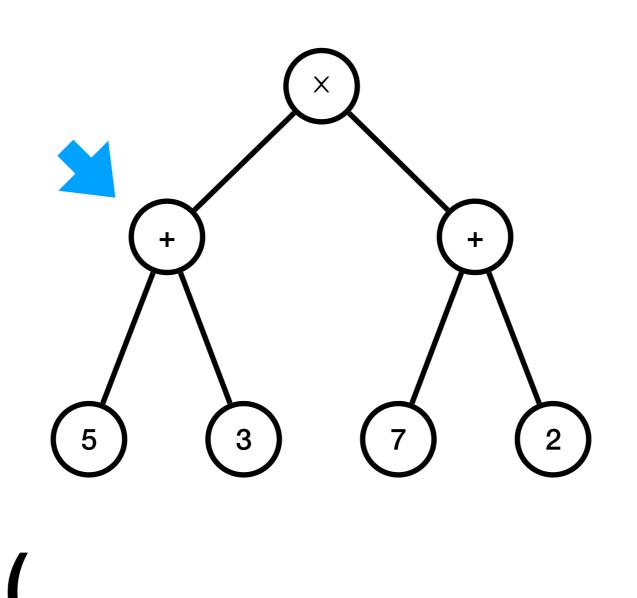
fonction évaluer-expression(nœud)

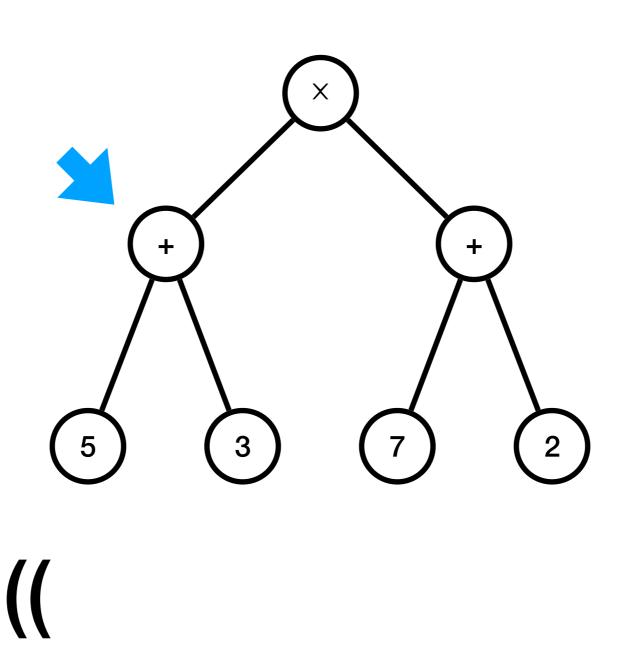
parcours postfixe!

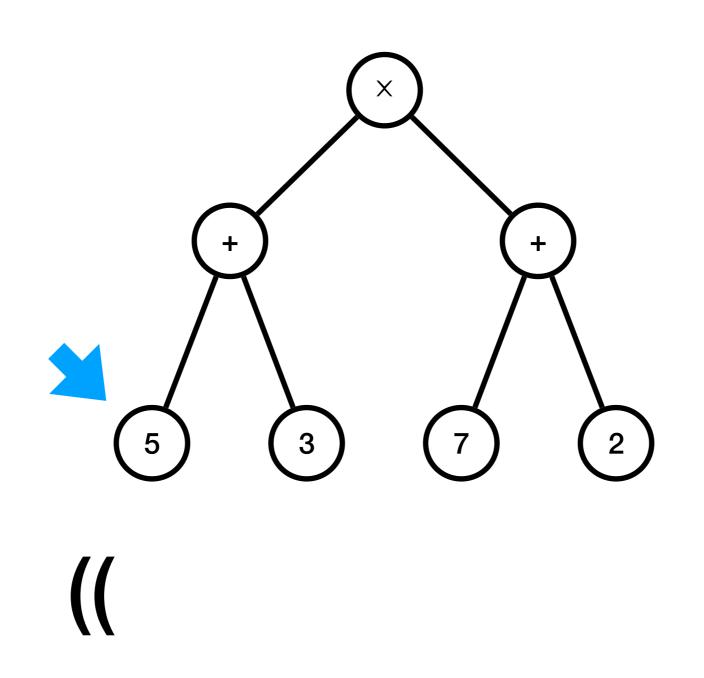


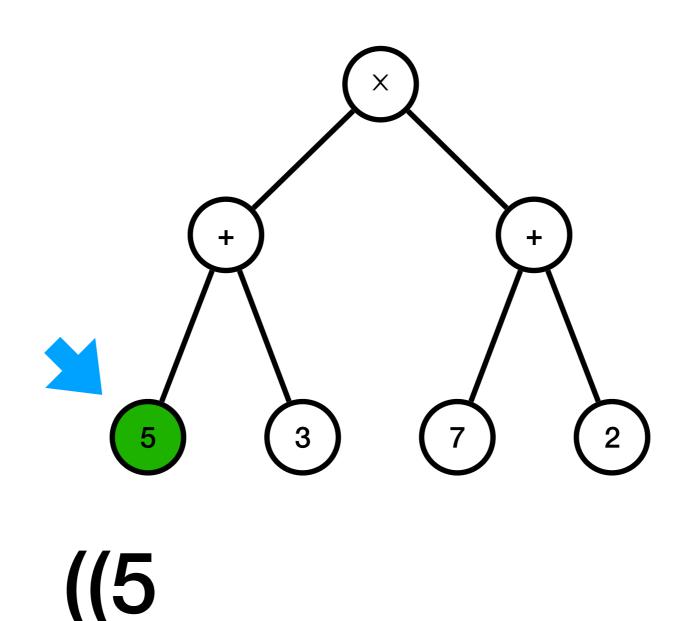


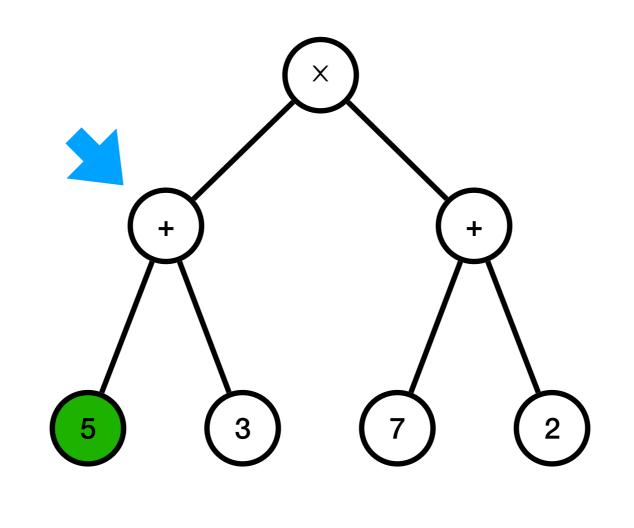




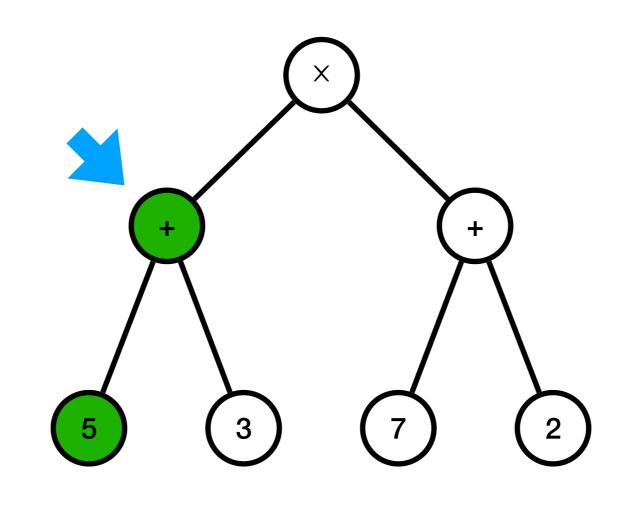




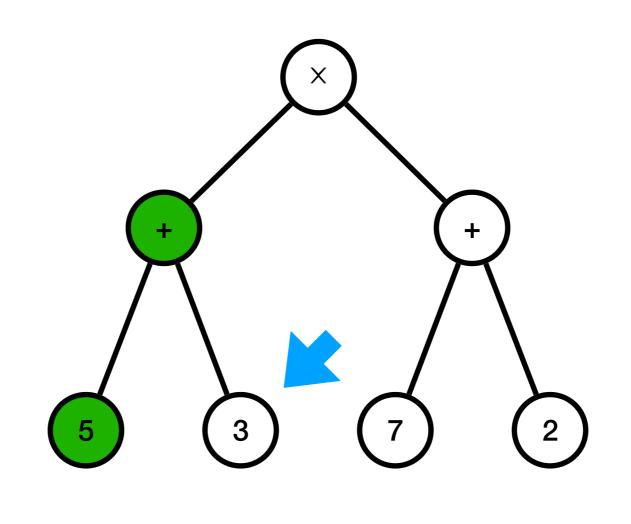




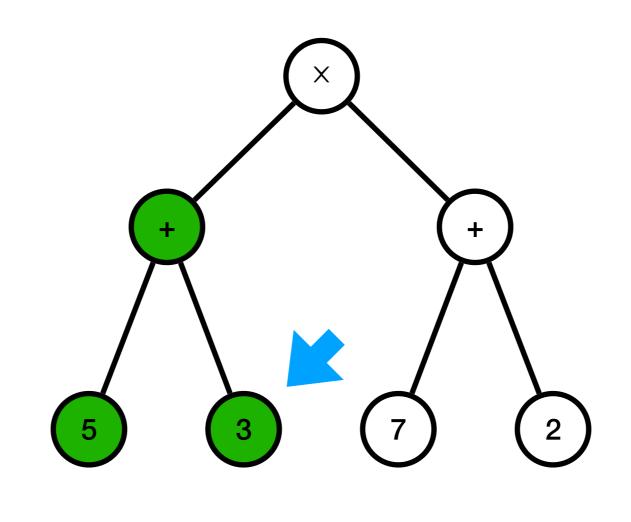
((5



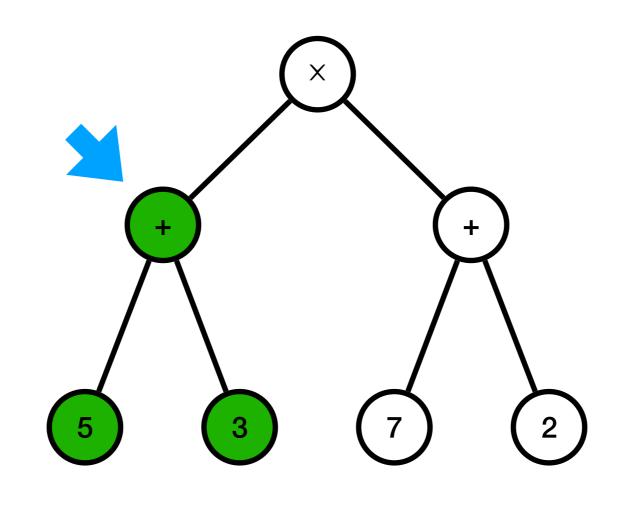
$$((5 +$$



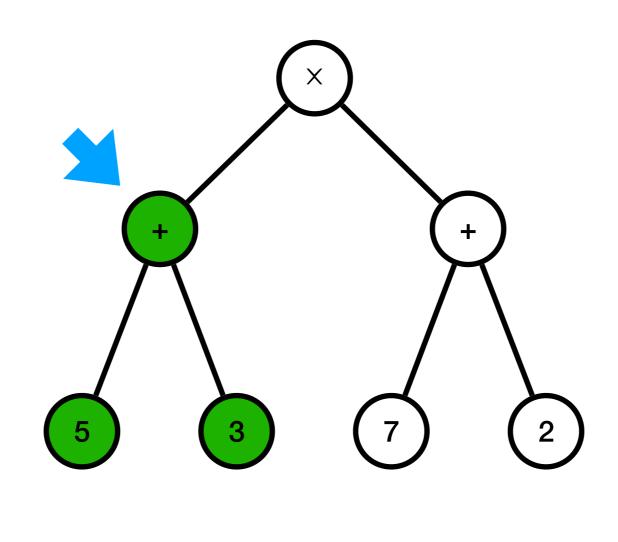
((5 +



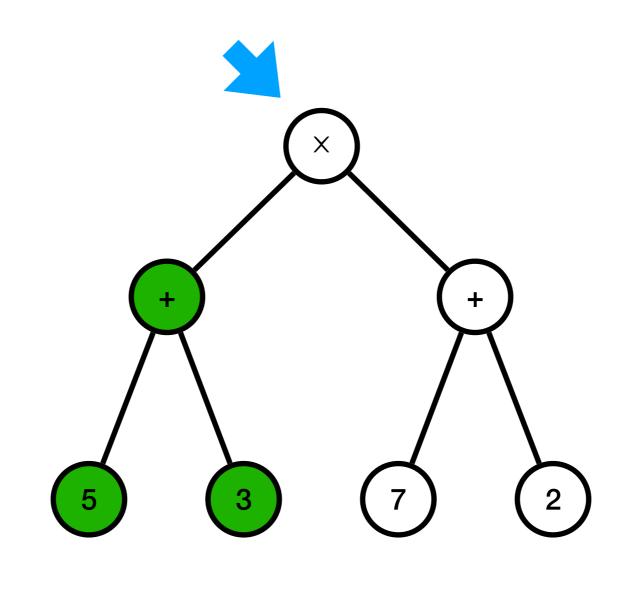
$$((5 + 3)$$



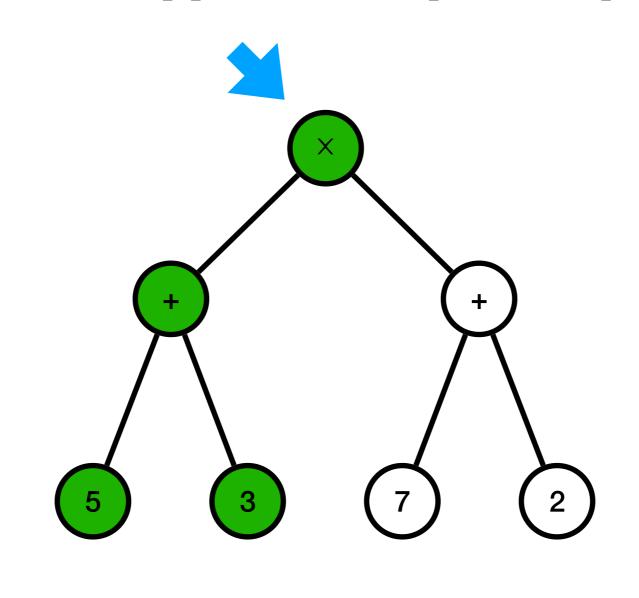
$$((5 + 3)$$



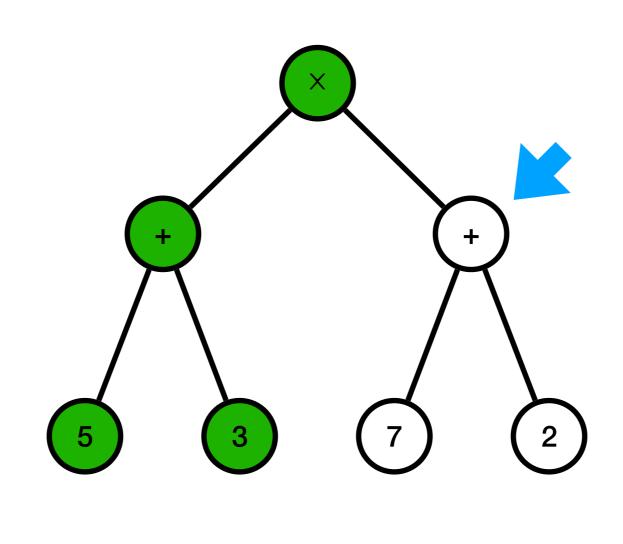
$$((5 + 3)$$



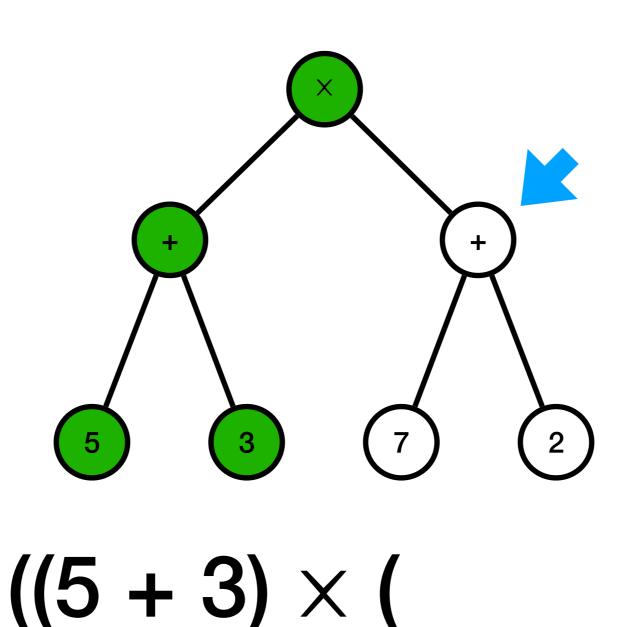
$$((5 + 3)$$

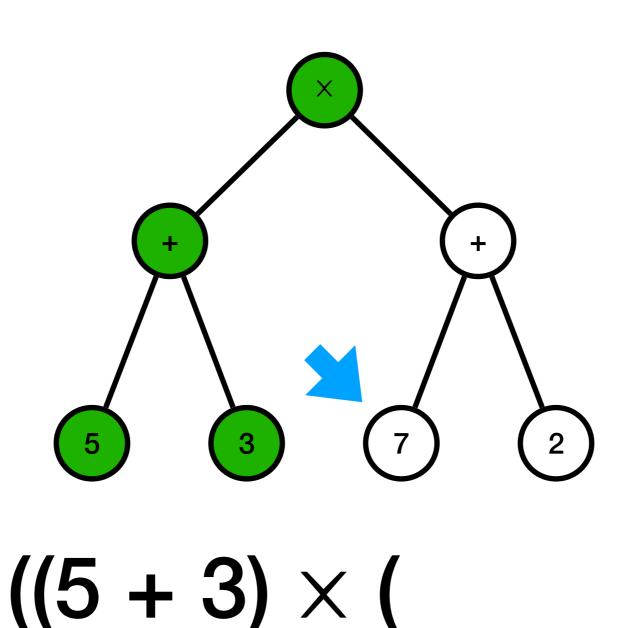


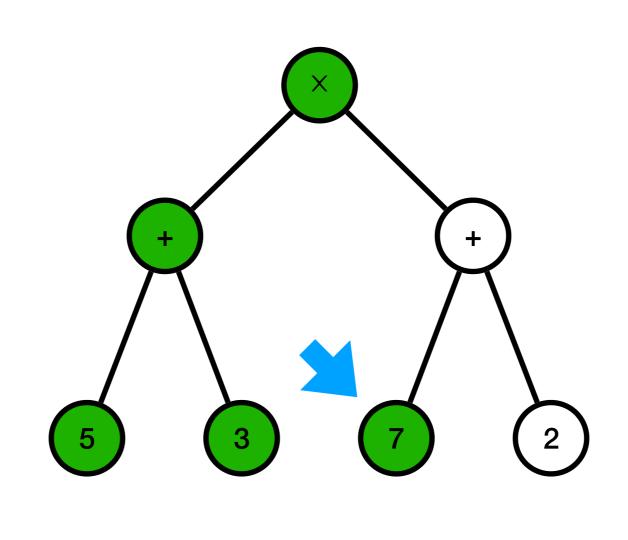
$$((5 + 3) \times$$



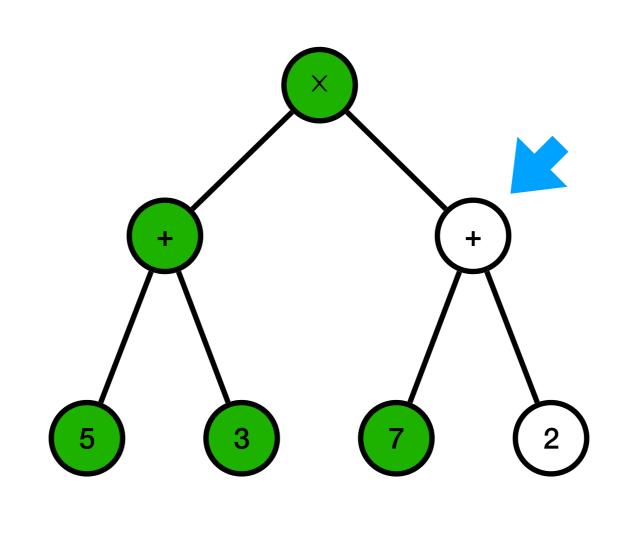
$$((5 + 3) \times$$



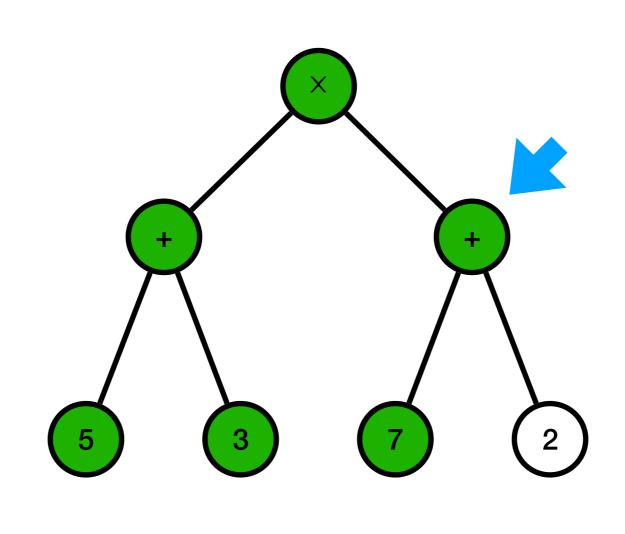




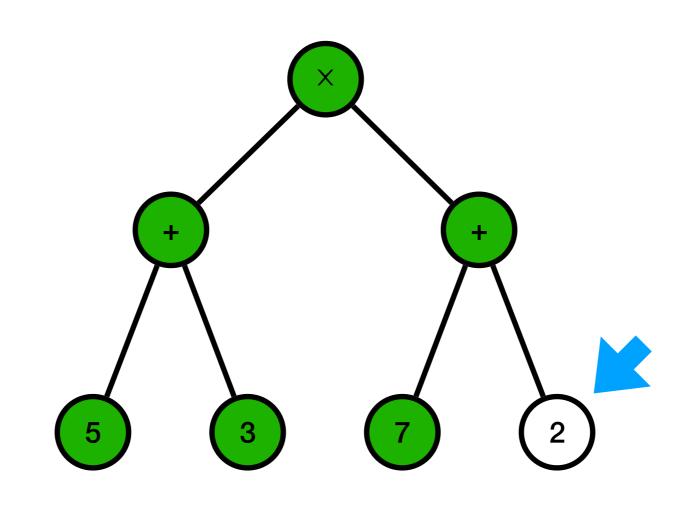
$$((5 + 3) \times (7 + 3))$$



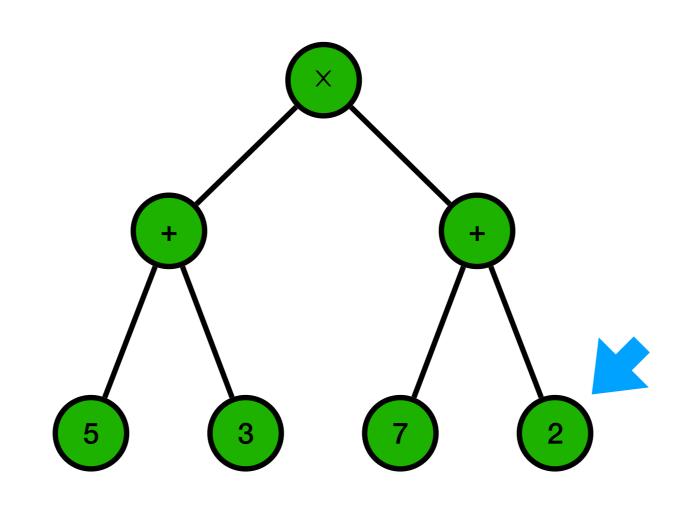
$$((5 + 3) \times (7 + 3))$$



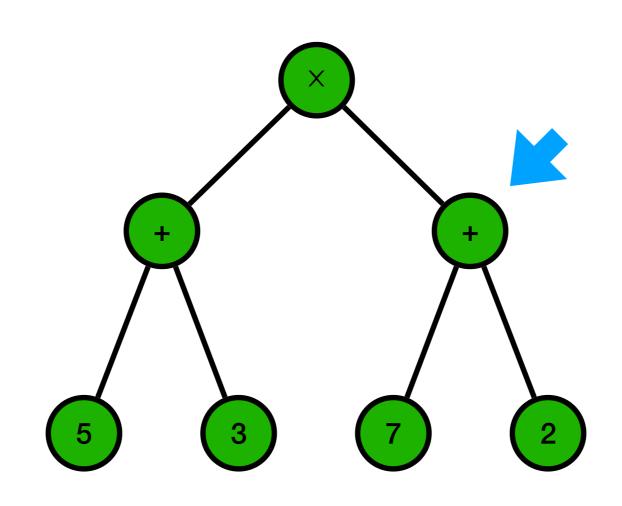
$$((5 + 3) \times (7 +$$



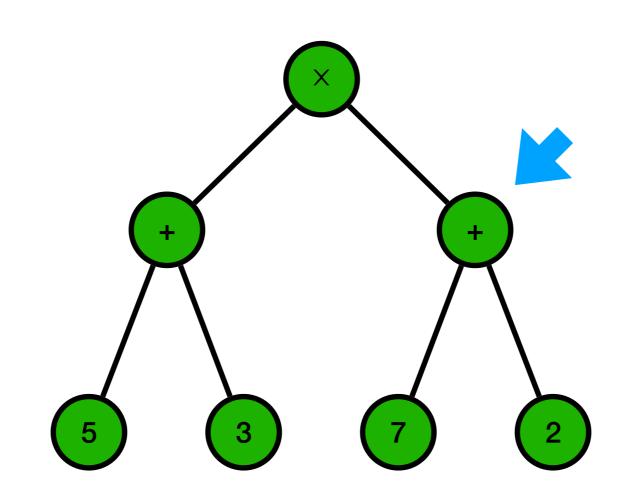
$$((5 + 3) \times (7 +$$



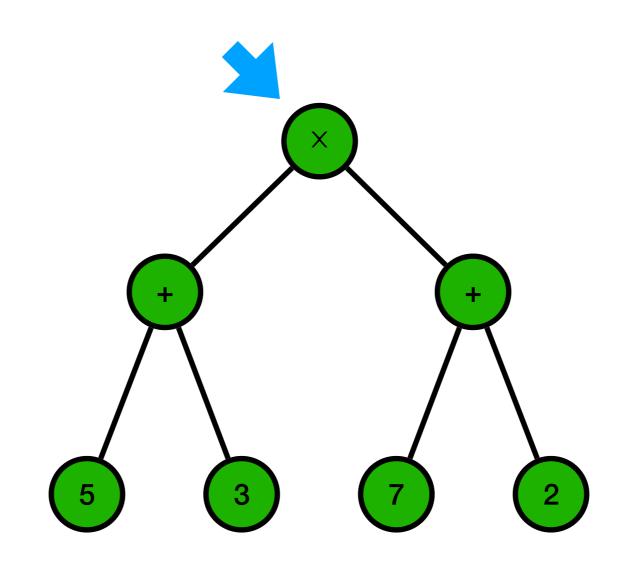
$$((5 + 3) \times (7 + 2)$$



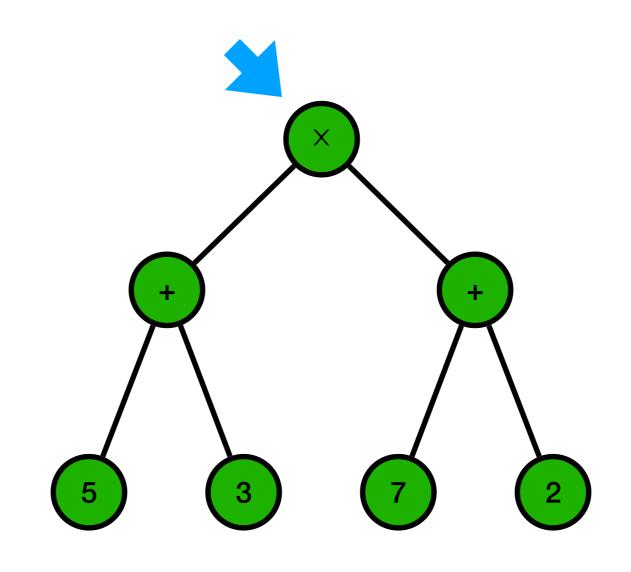
$$((5 + 3) \times (7 + 2)$$



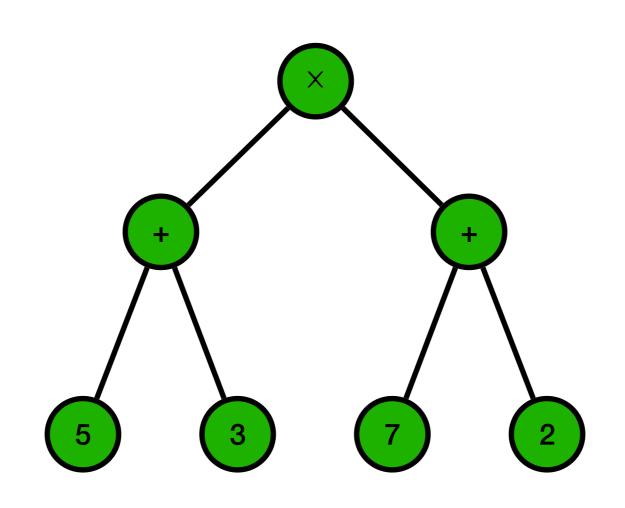
$$((5 + 3) \times (7 + 2)$$



$$((5 + 3) \times (7 + 2)$$



$$((5 + 3) \times (7 + 2))$$



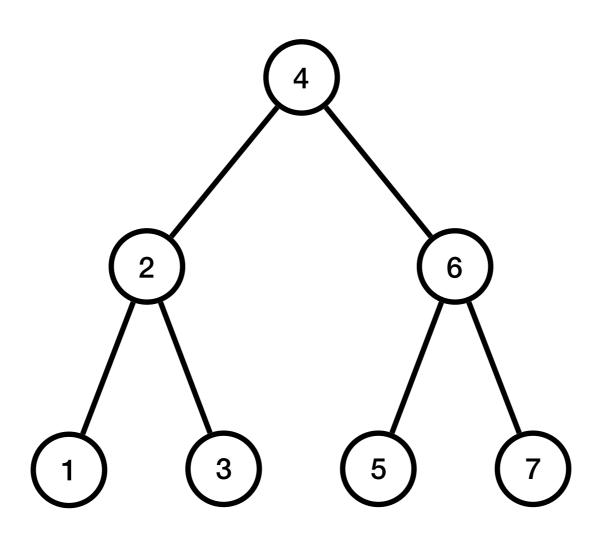
$$((5 + 3) \times (7 + 2))$$

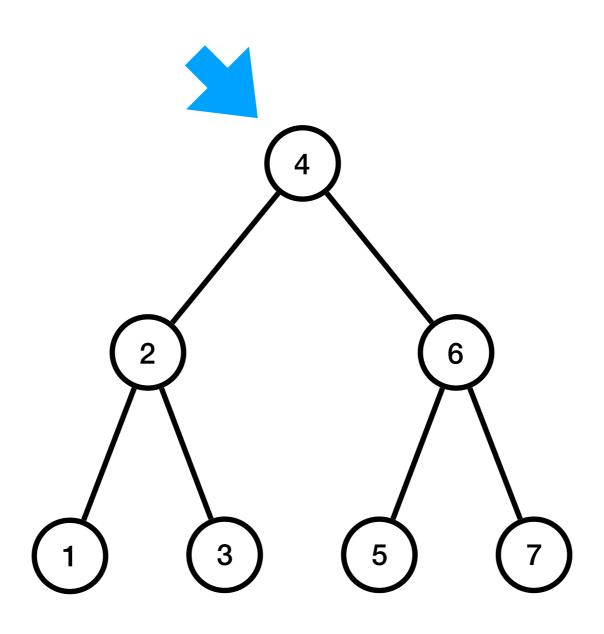
Afficher une expression

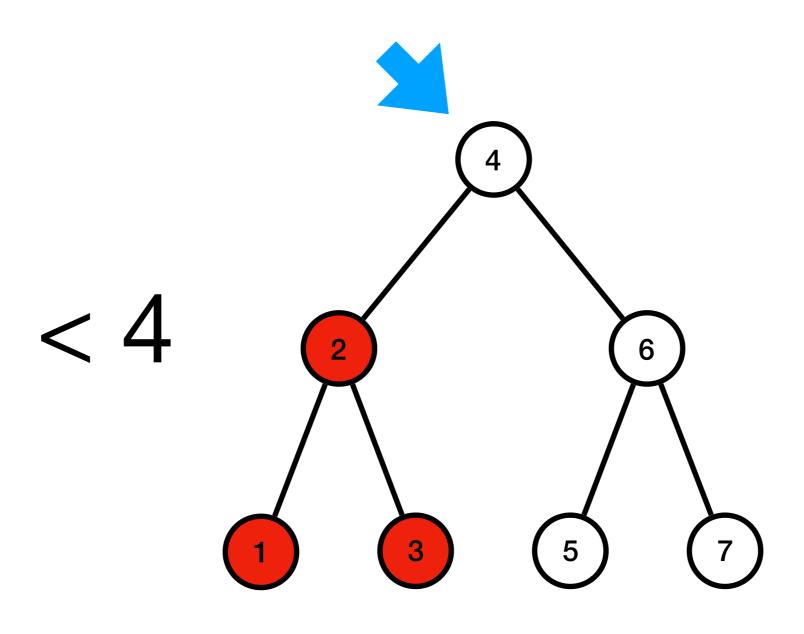
```
procédure afficher-expression(nœud)
    si valeur[nœud] est un nombre alors
        afficher valeur[nœud]
    sinon
        afficher "("
        afficher-expression(gauche[nœud])
        afficher valeur[nœud] (c'est le signe d'opération)
        afficher-expression(droite[nœud])
        afficher ")"
```

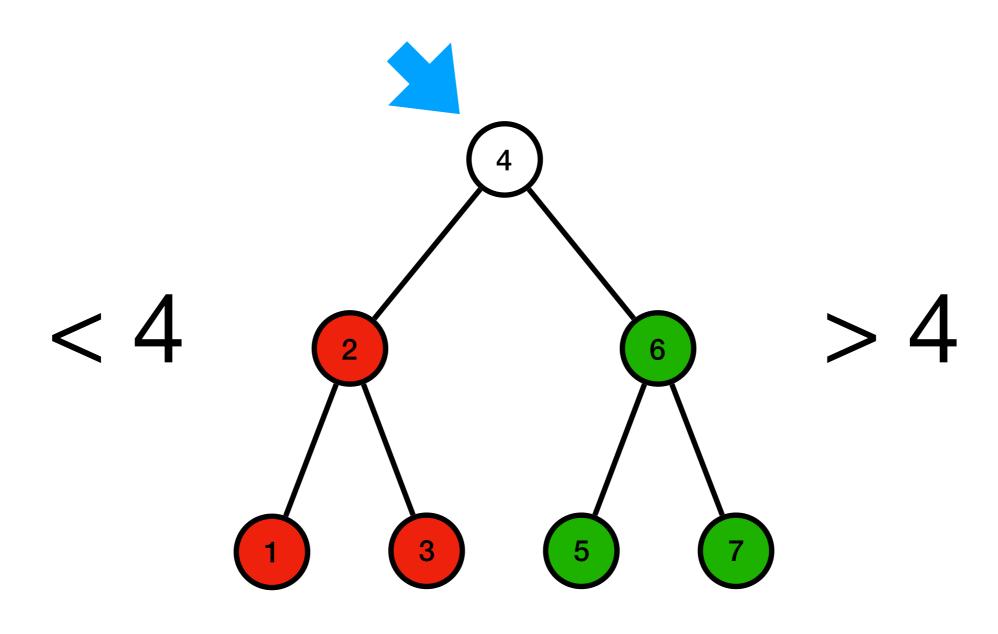
Afficher une expression

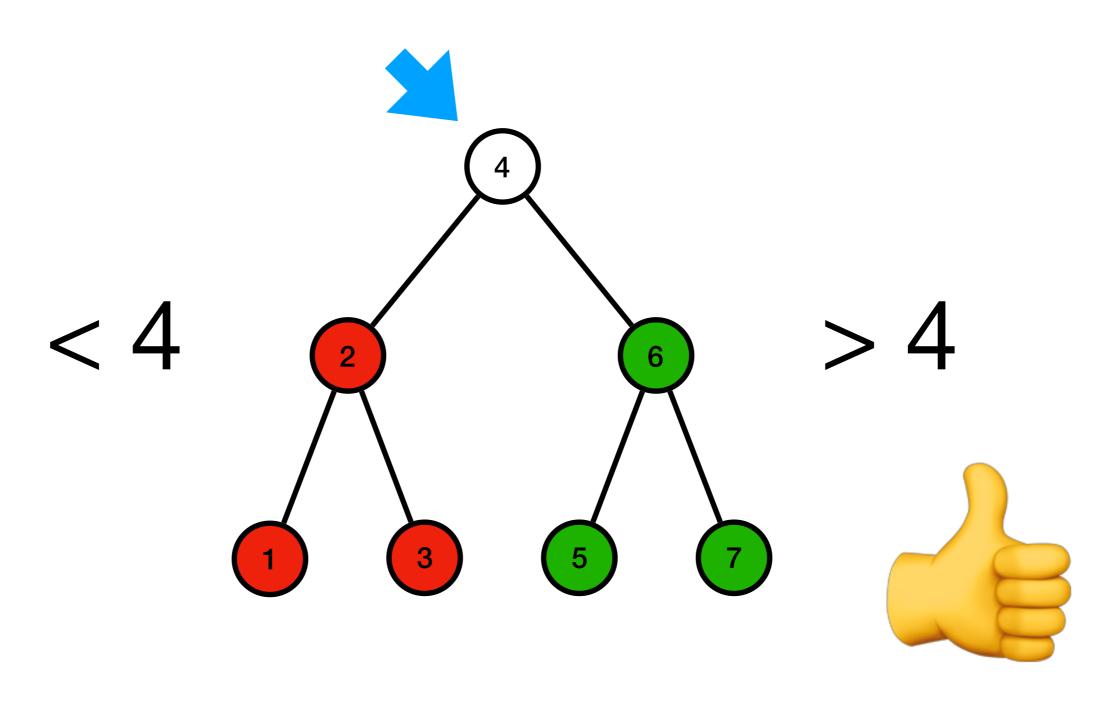
```
procédure afficher-expression(nœud)
    si valeur[nœud] est un nombre alors
        afficher valeur[nœud]
    sinon
        afficher "("
        afficher-expression(gauche[nœud])
        afficher valeur[nœud]
        afficher-expression(droite[nœud])
        afficher ")"
```

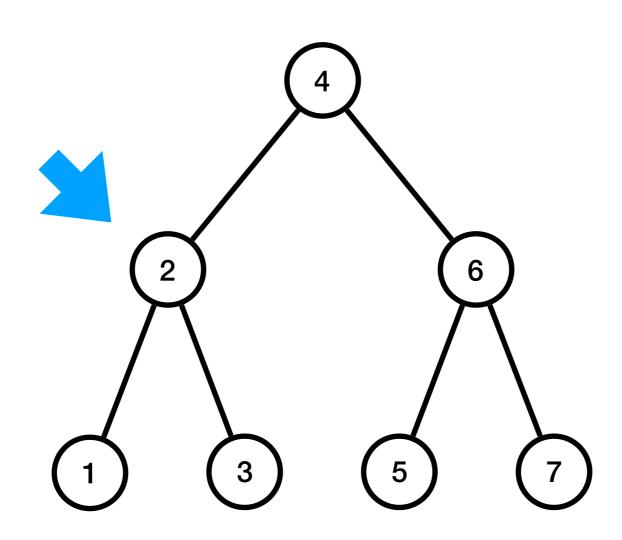


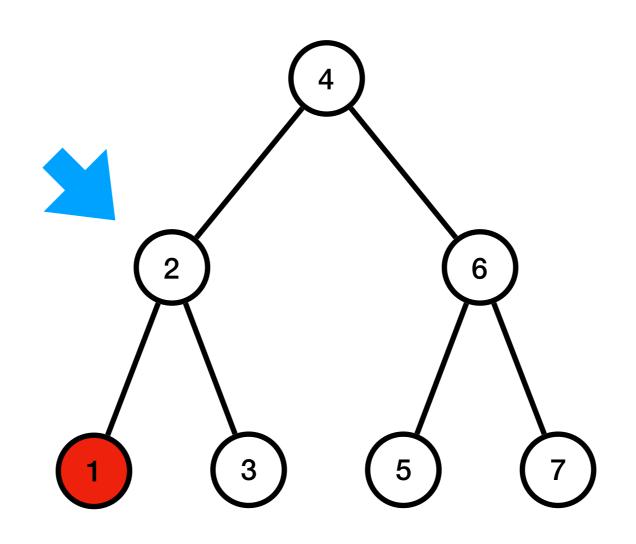


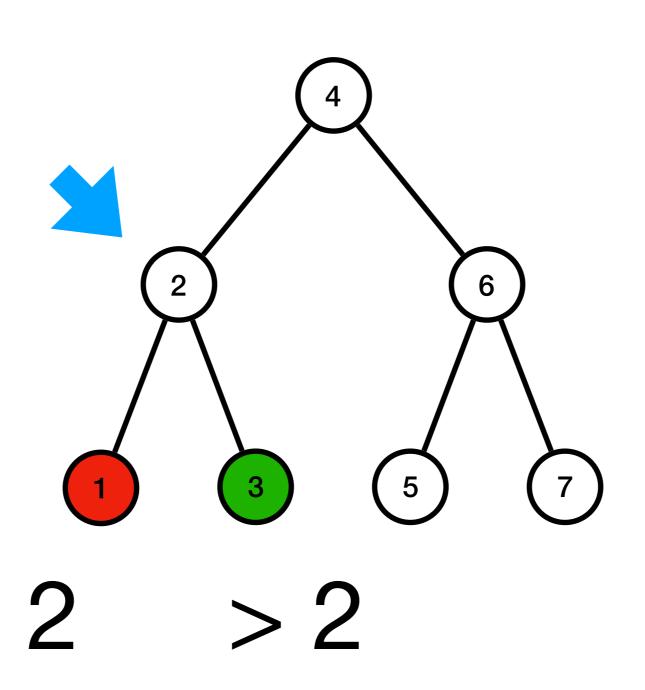


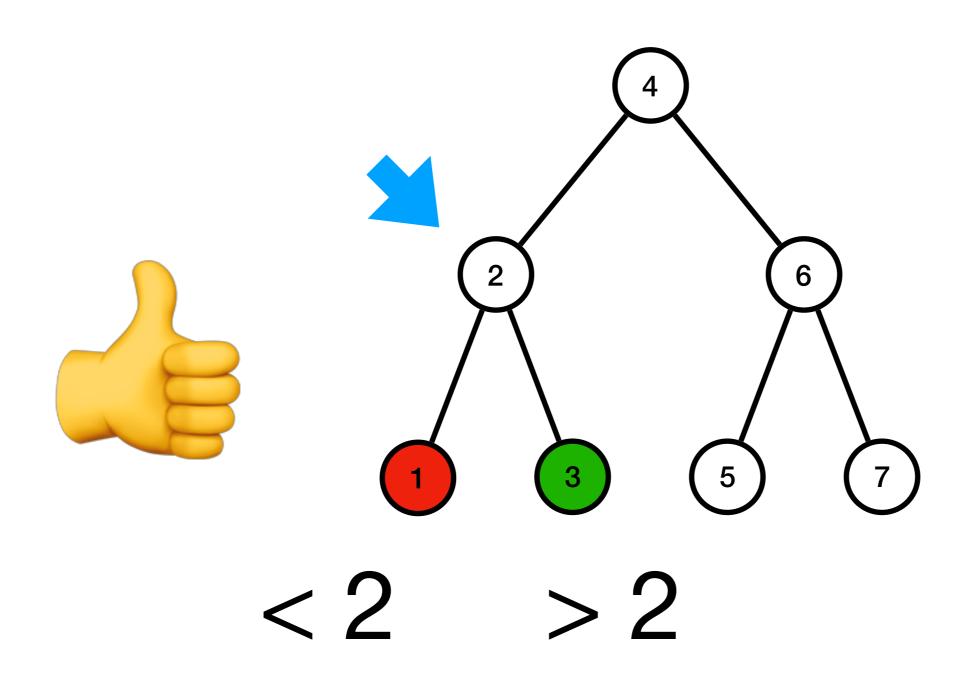


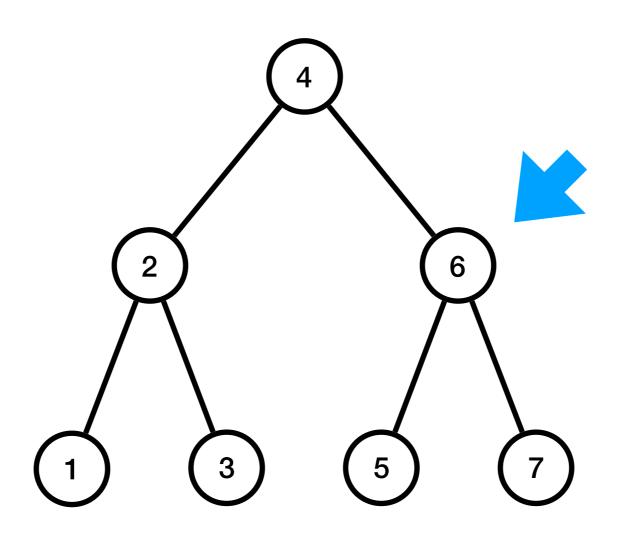


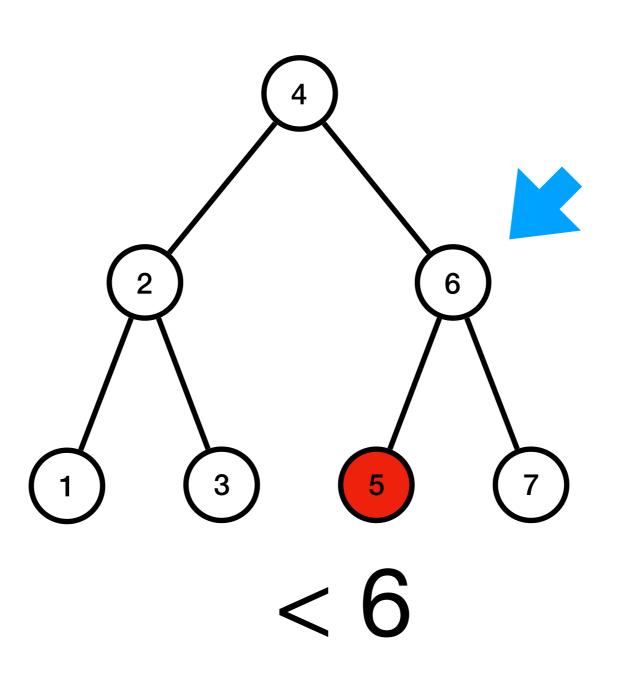


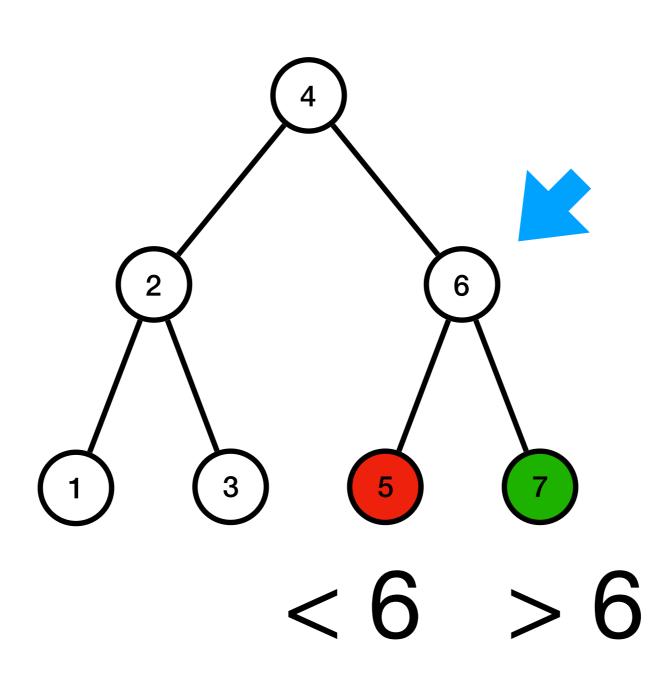


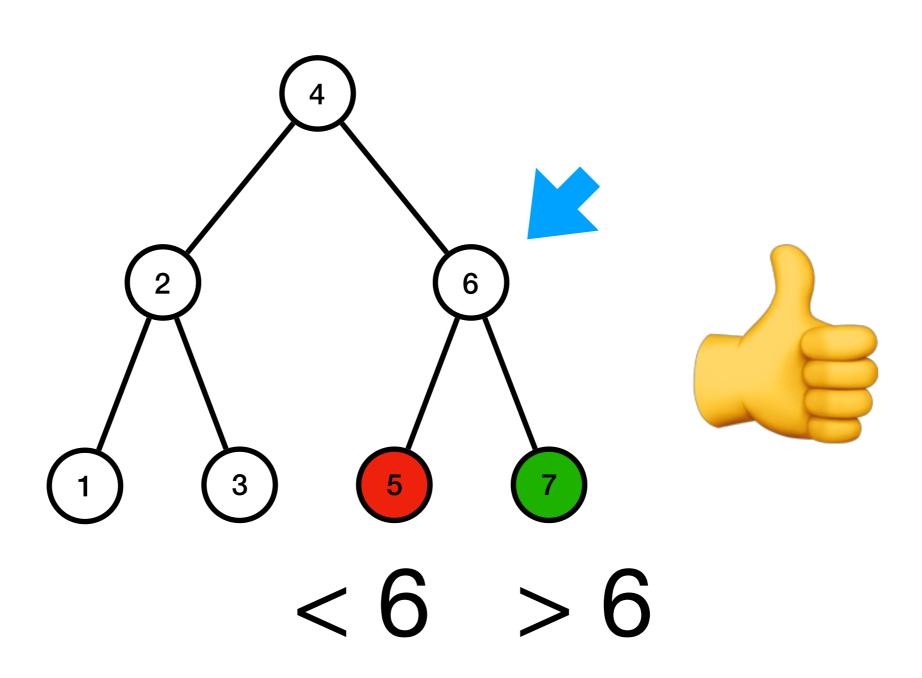


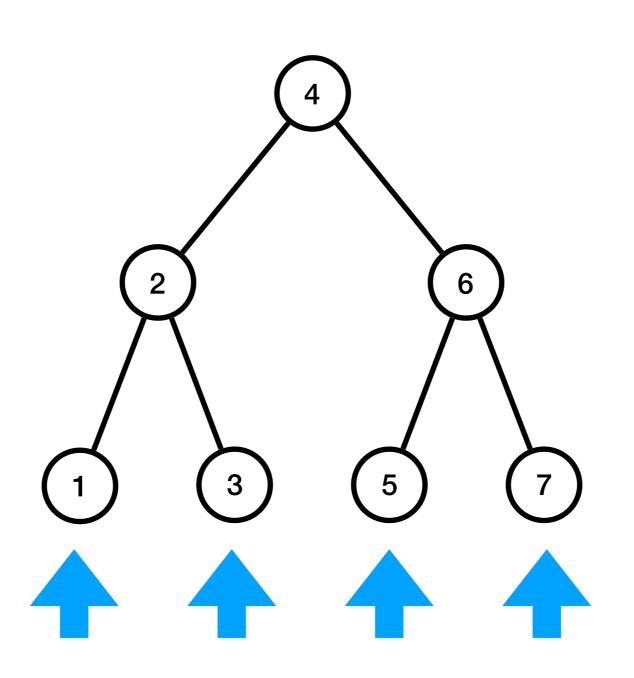


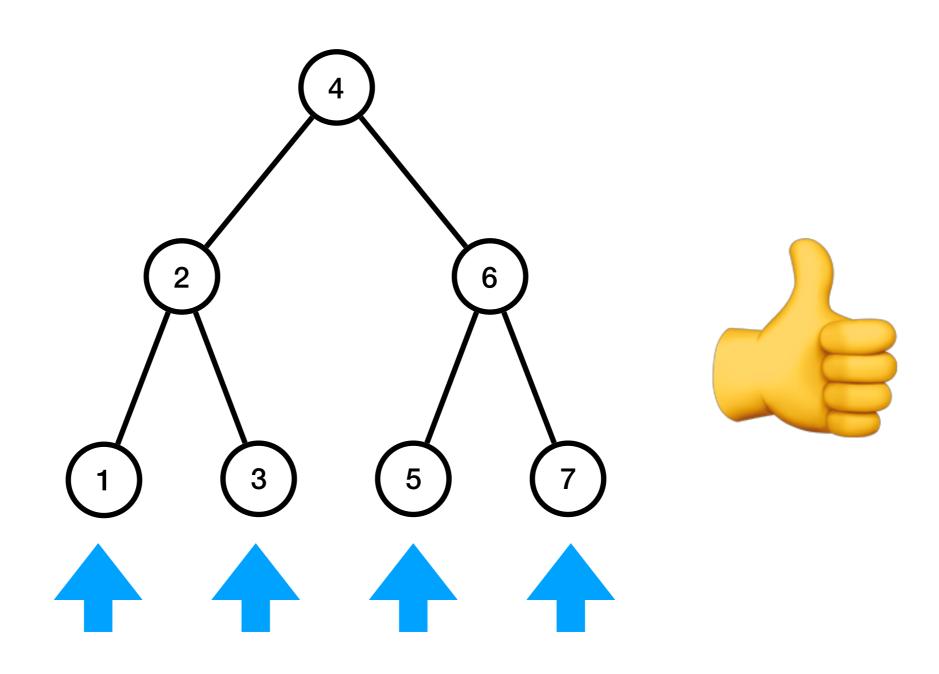


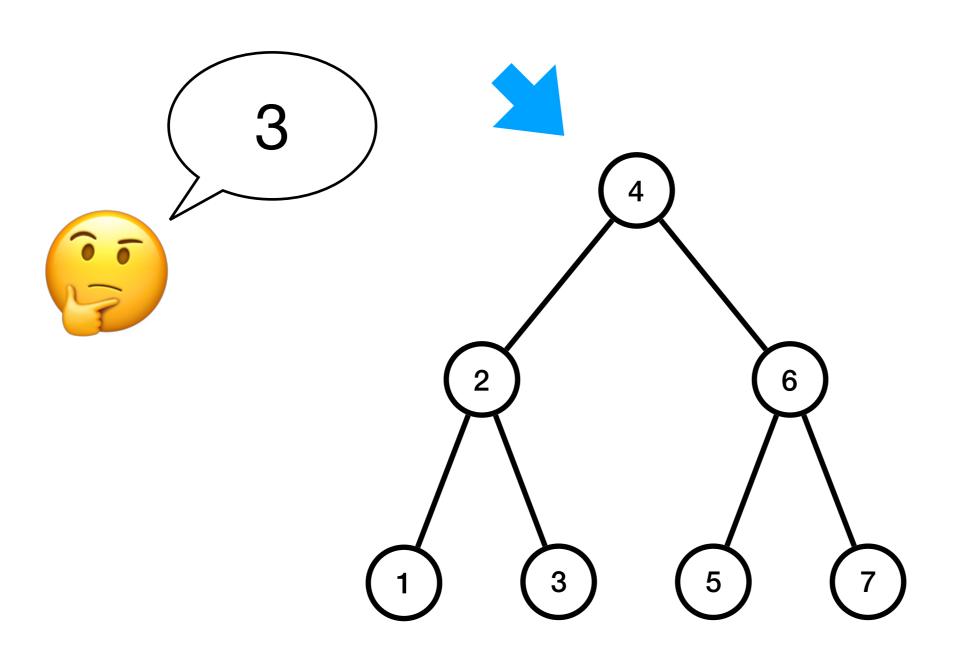


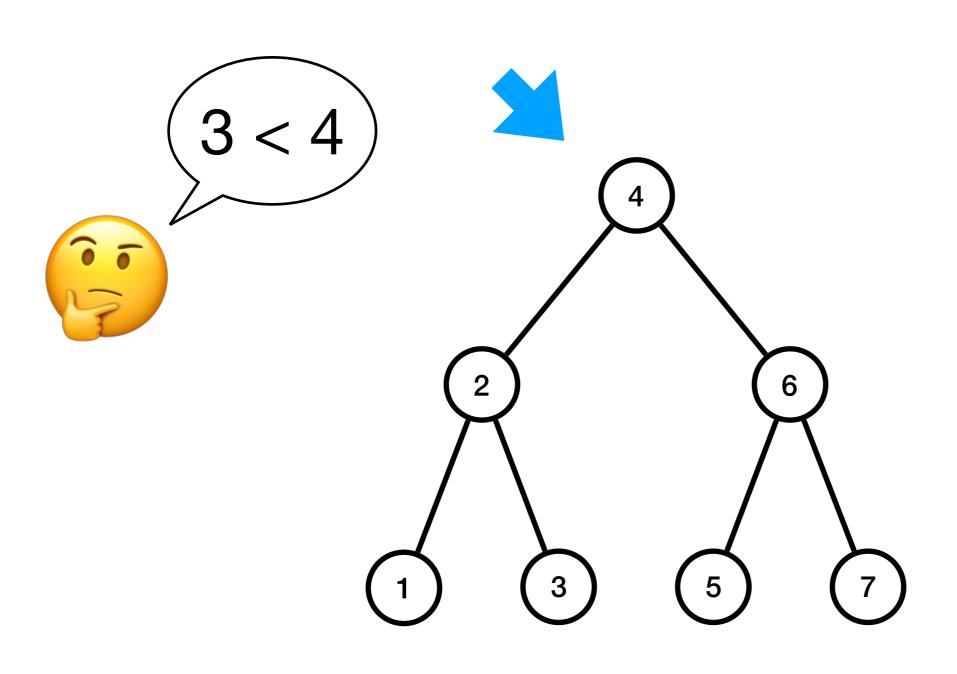


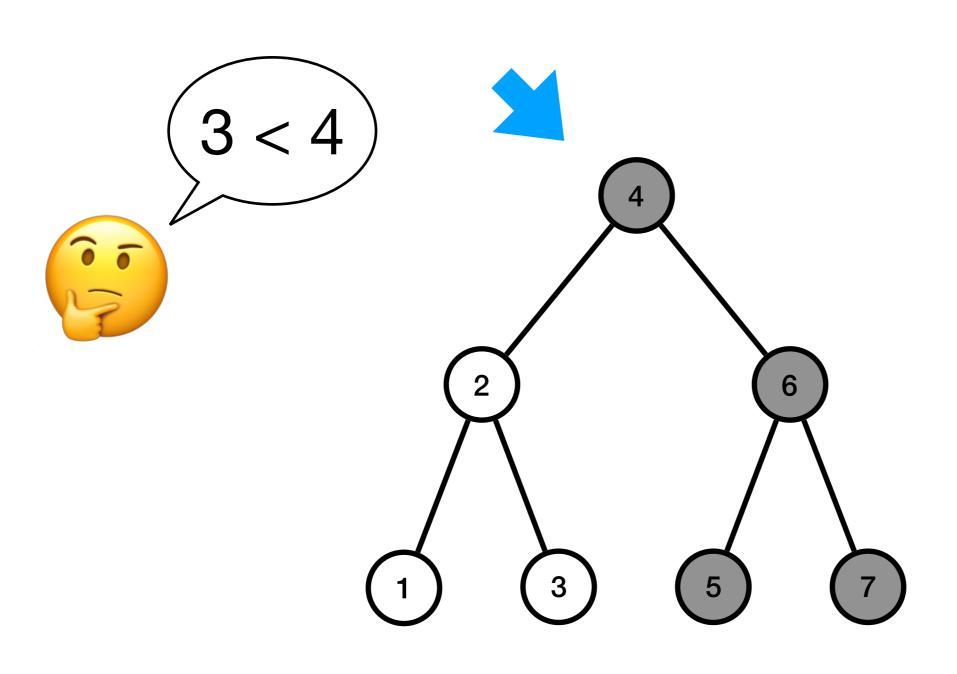


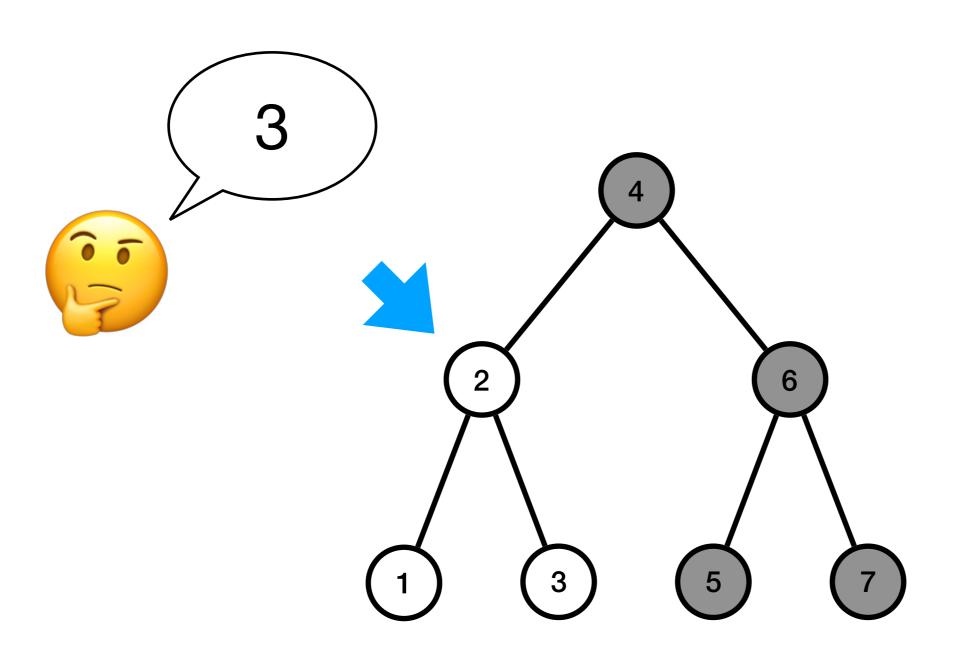


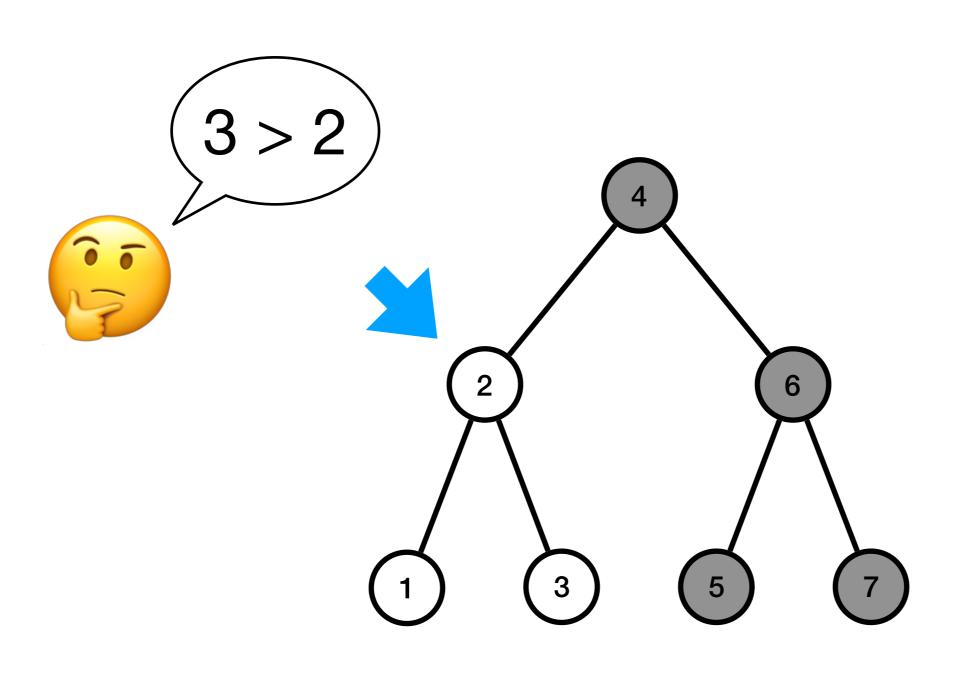


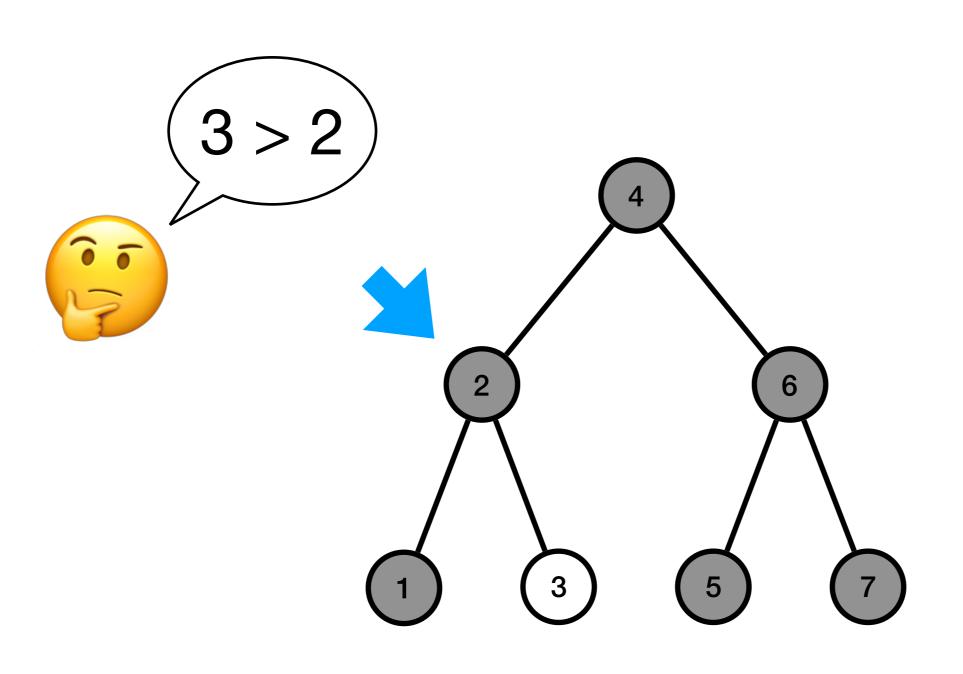


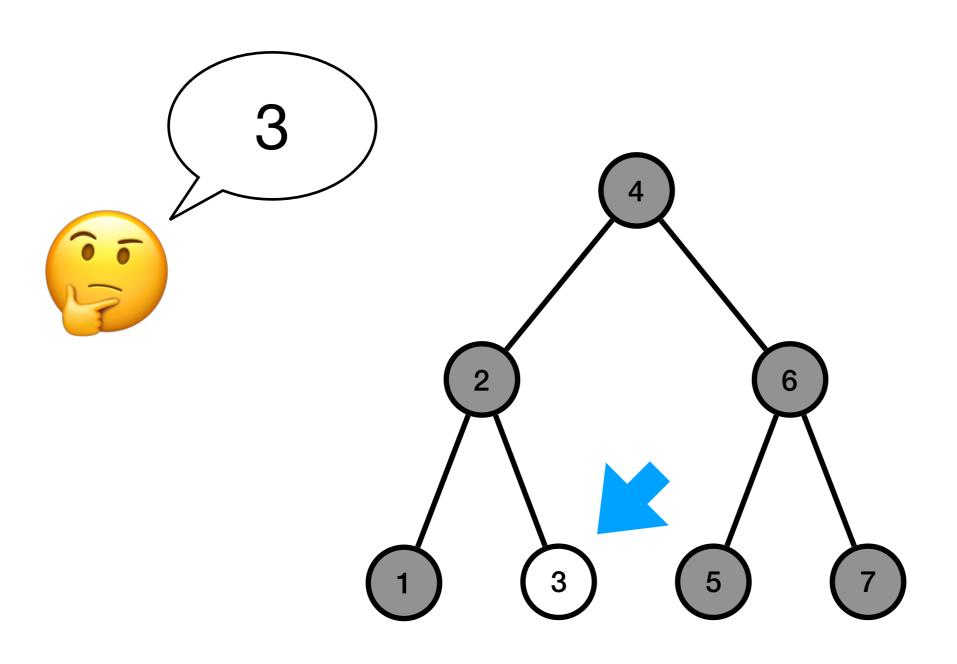


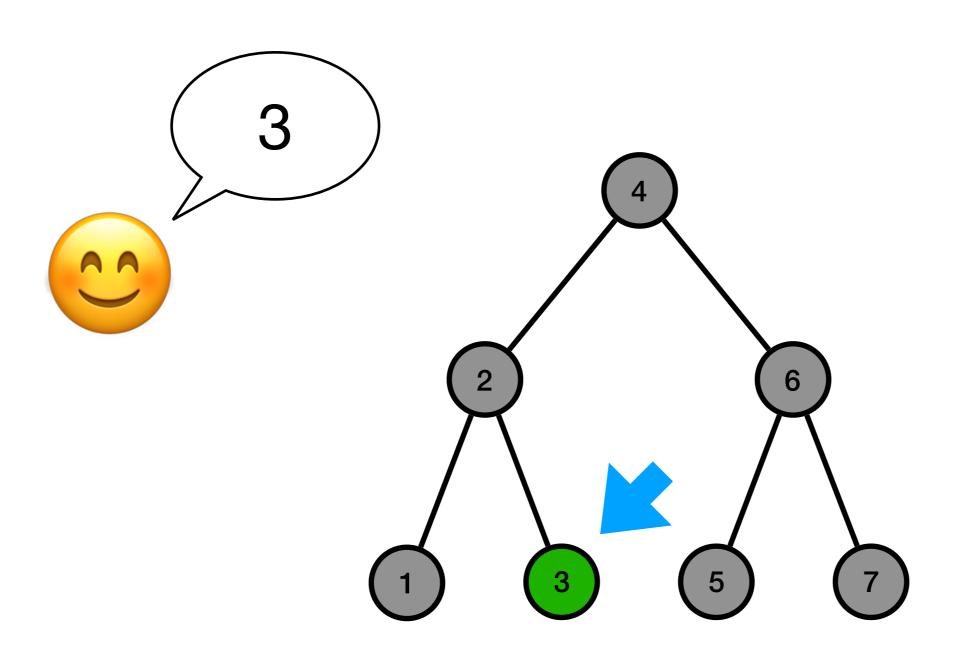


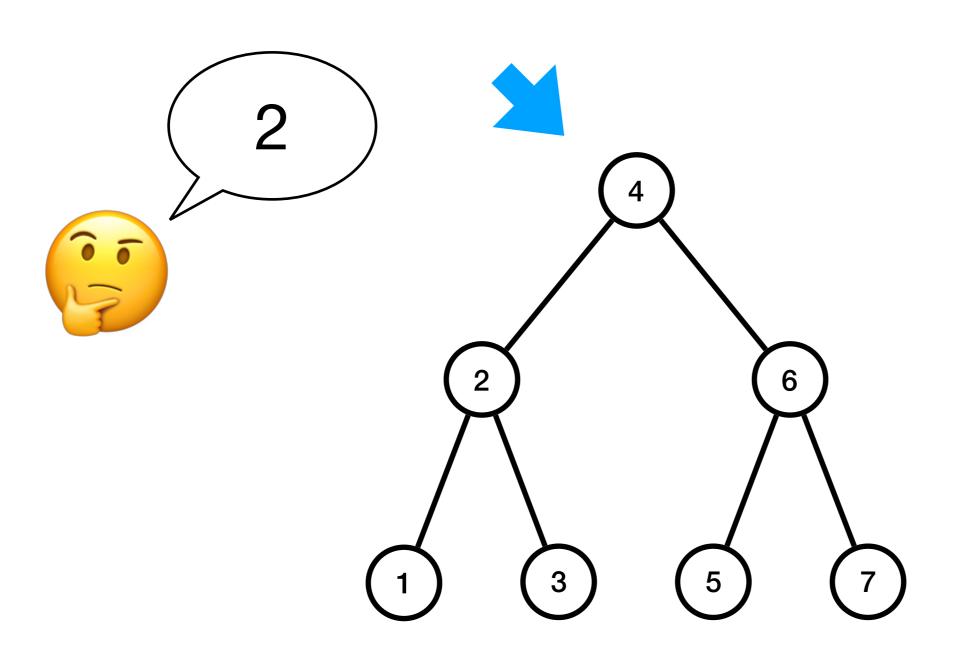


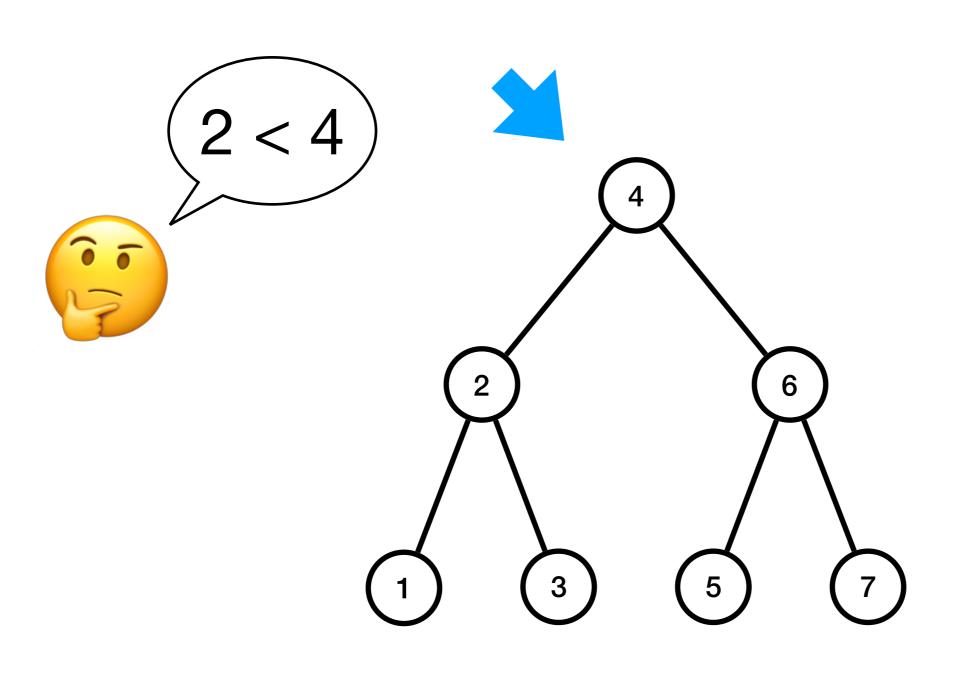


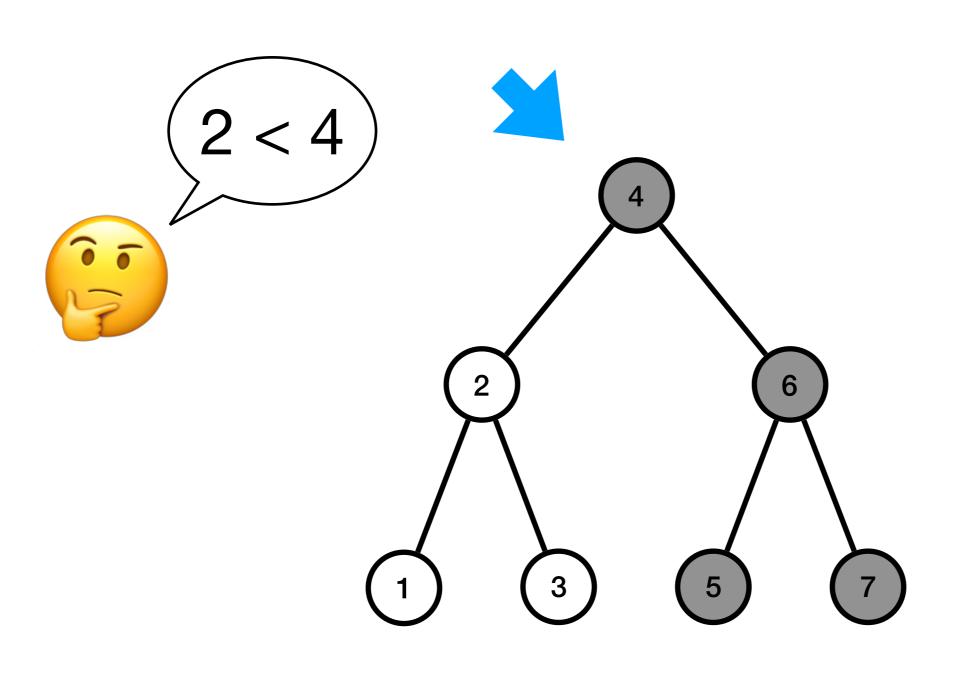


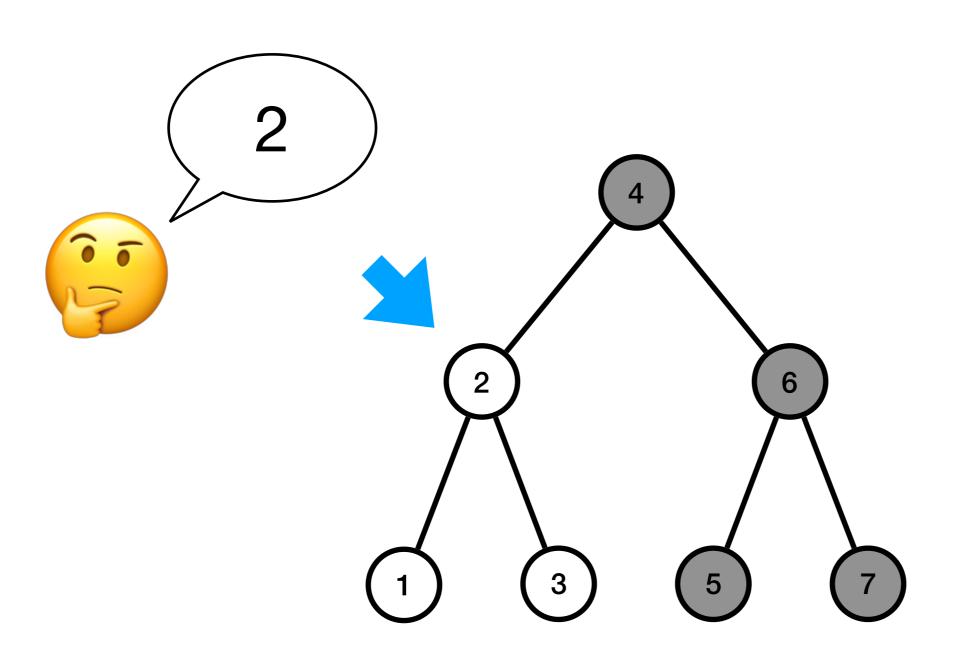


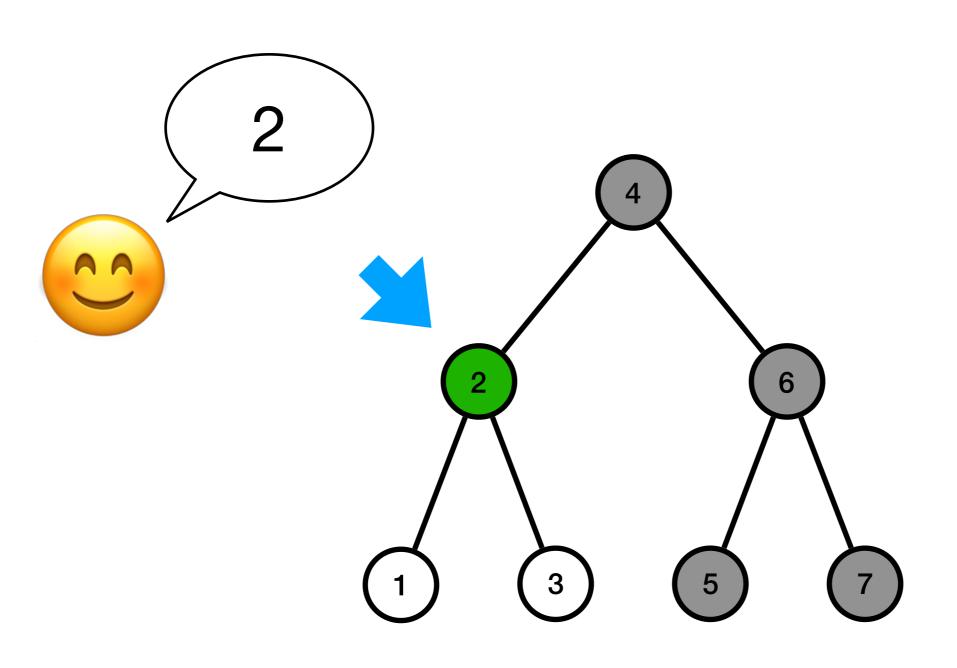


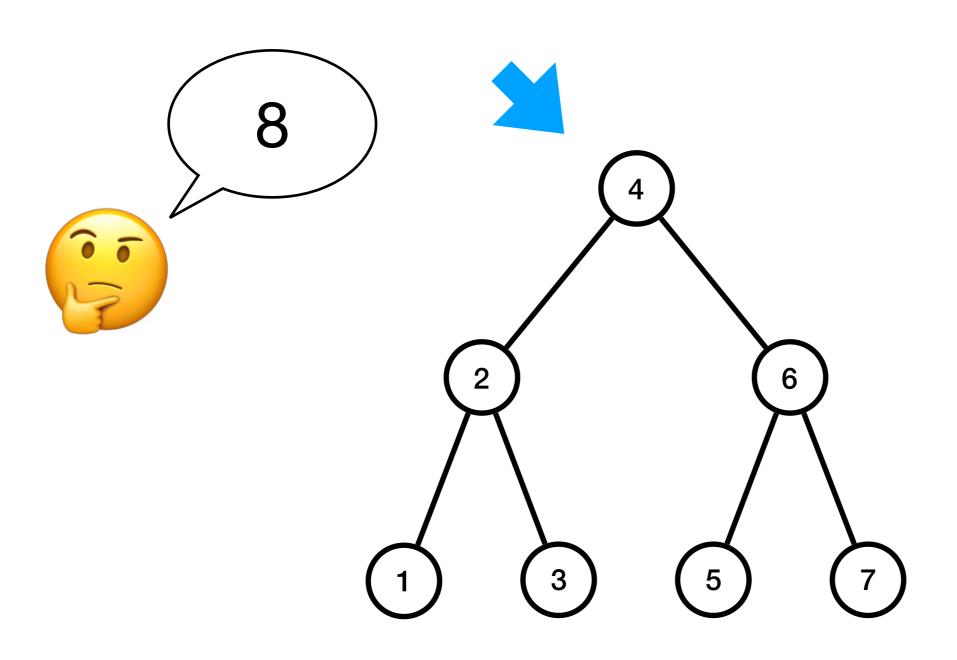


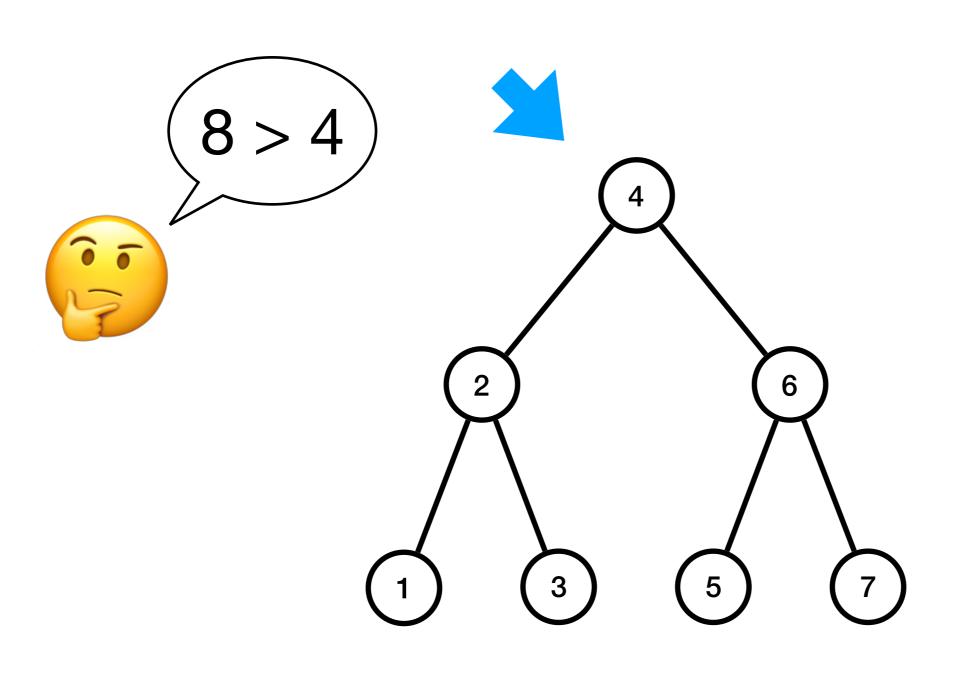


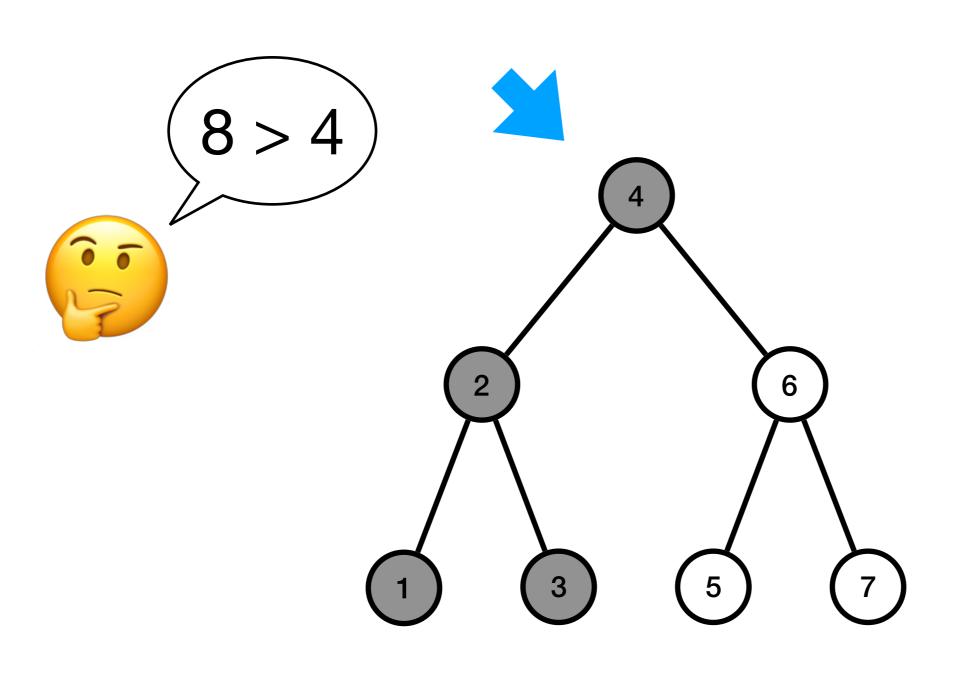


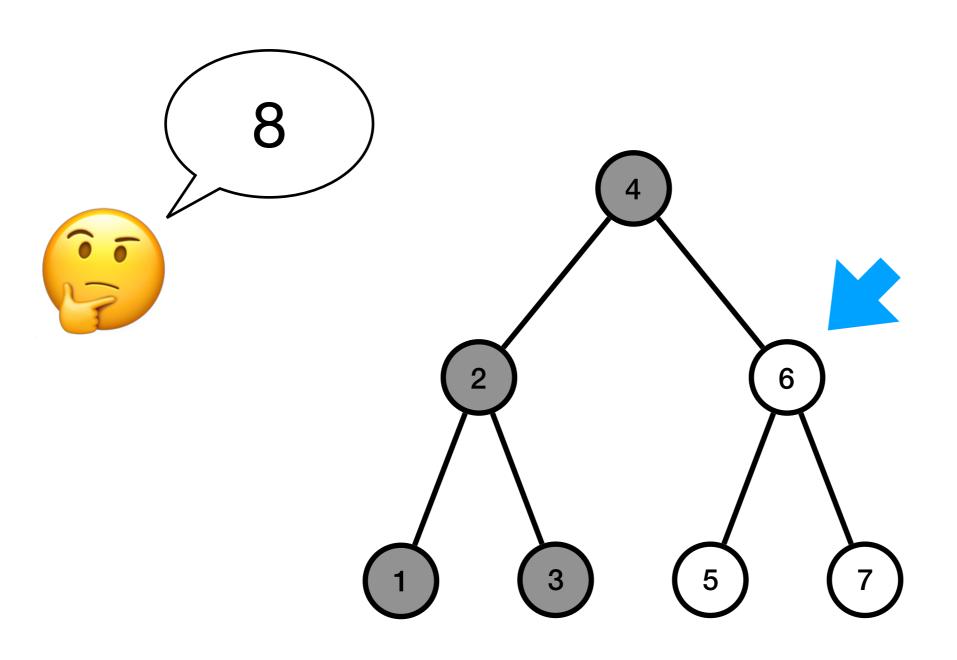


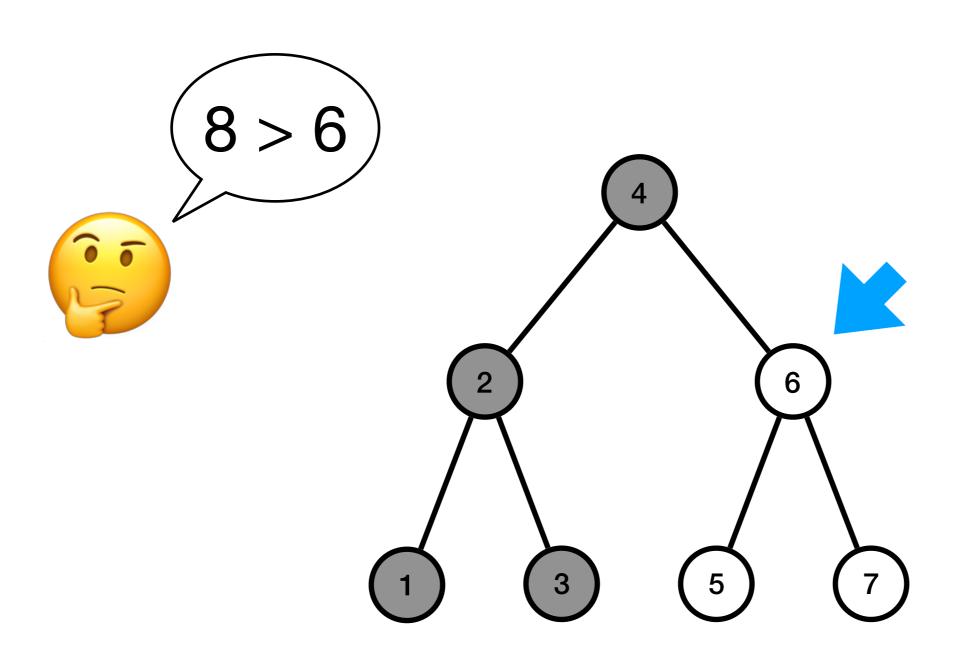


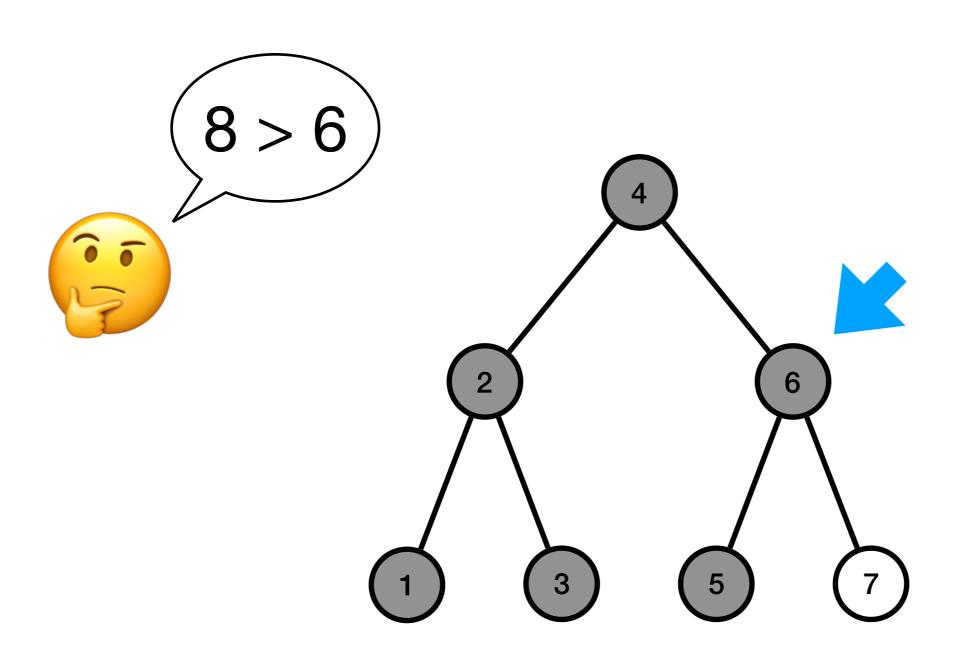


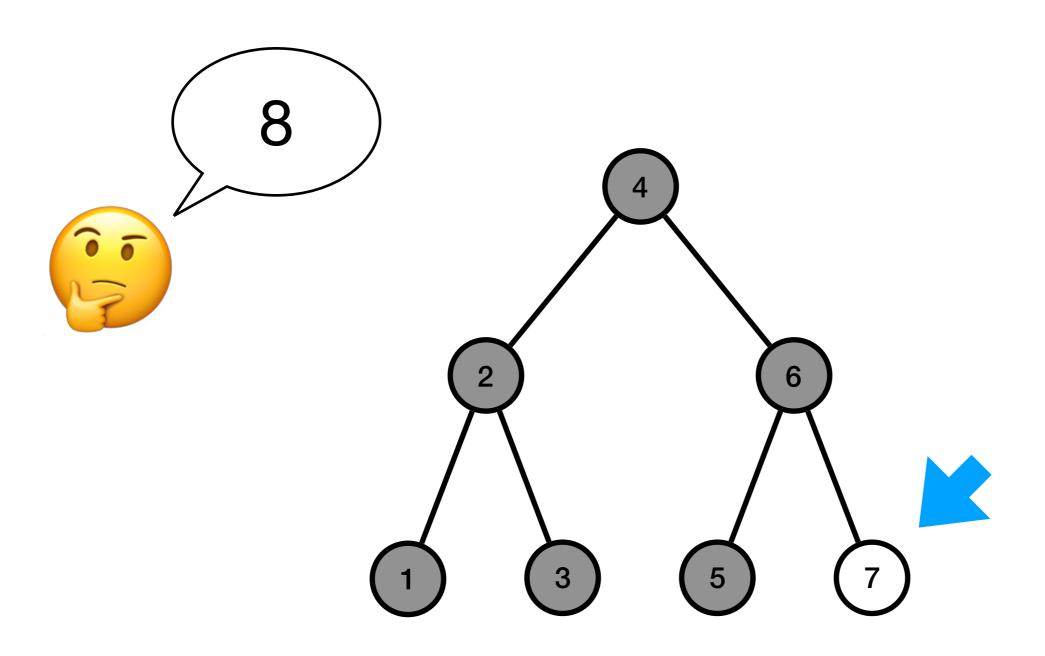


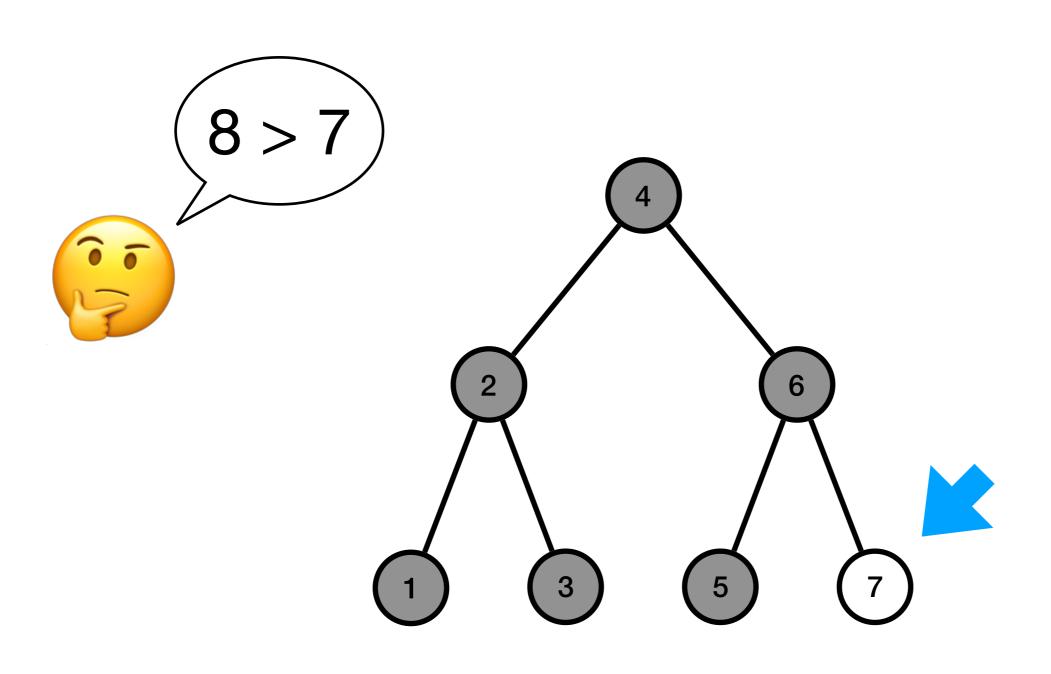


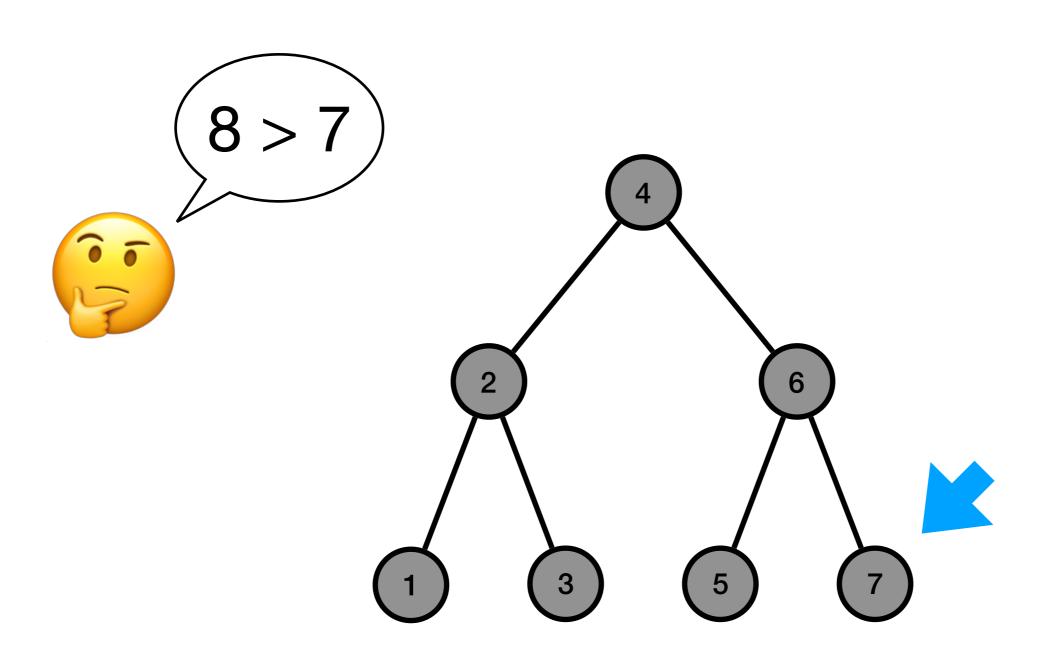


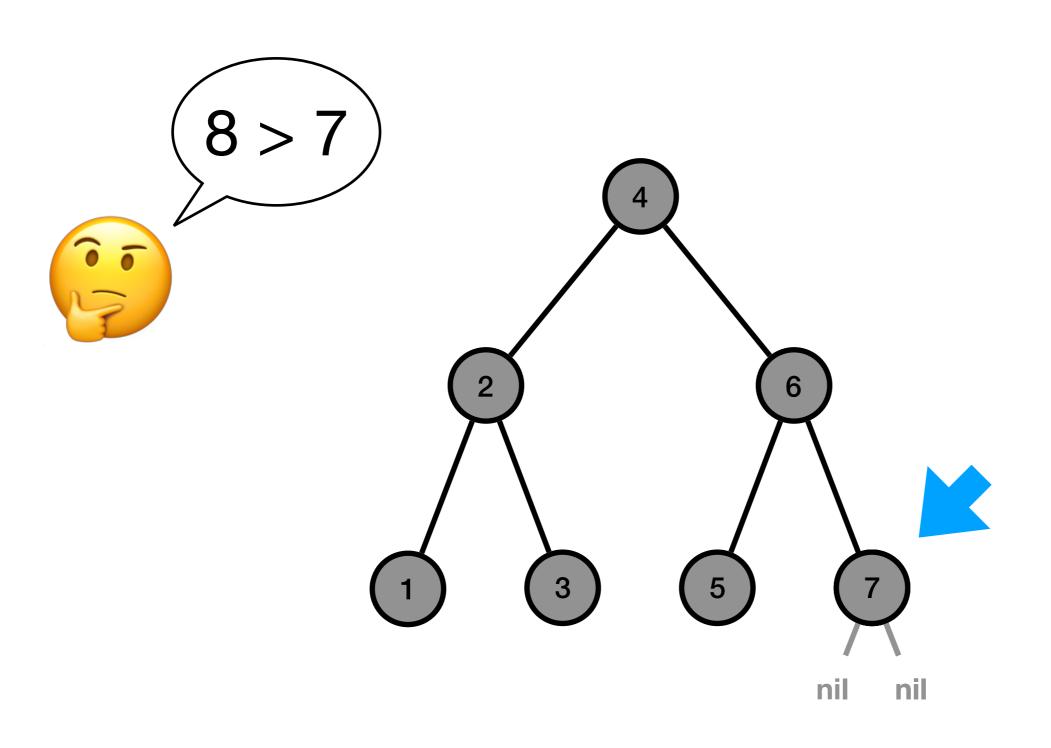


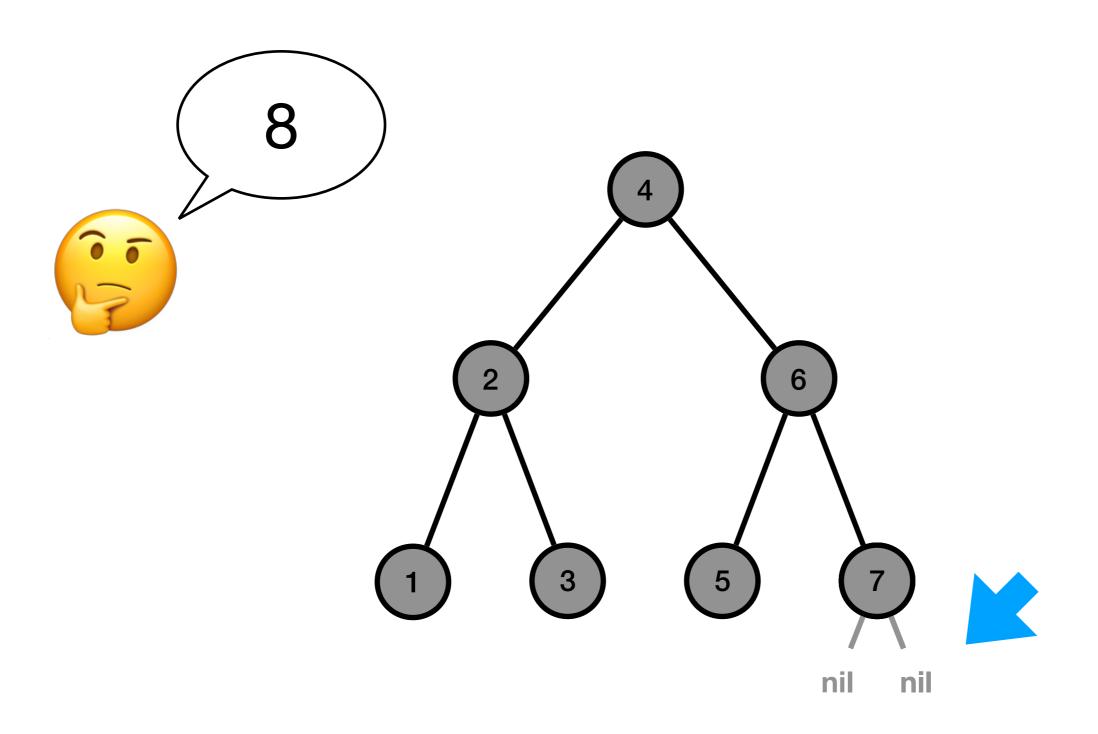


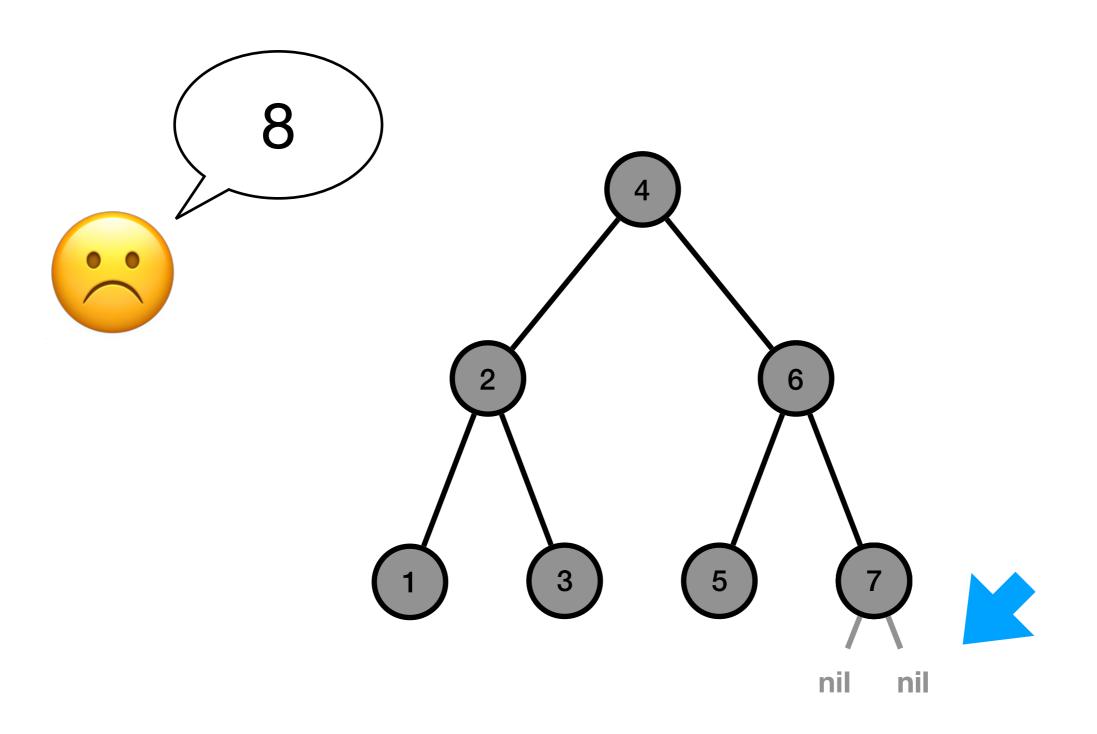






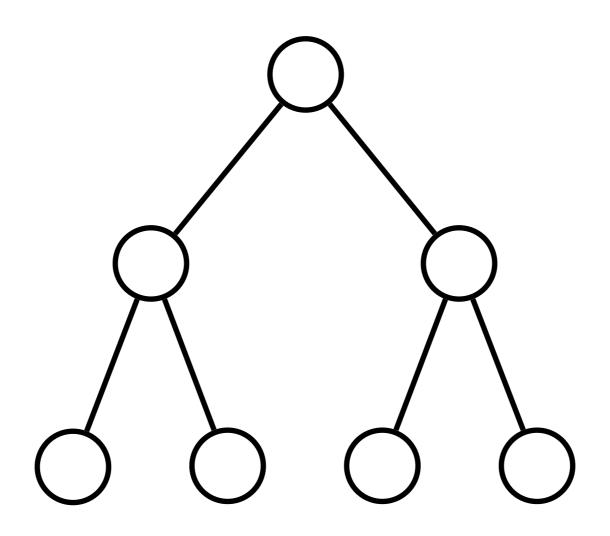




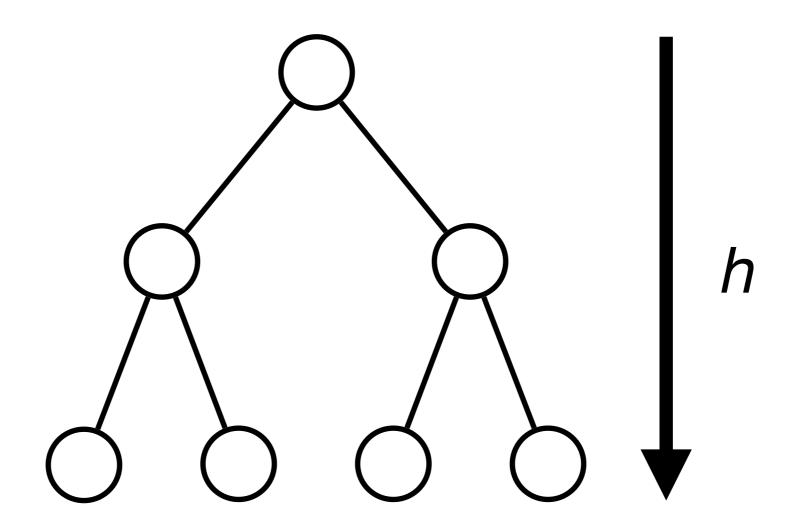


```
fonction rechercher-abr(nœud, x)
    si nœud = nil alors
    retourner faux
    sinon si x = valeur[nœud] alors
    retourner vrai
    sinon si x < valeur[nœud] alors
    return rechercher-abr(gauche[nœud], x)
    sinon
    return rechercher-abr(droite[nœud], x)</pre>
```

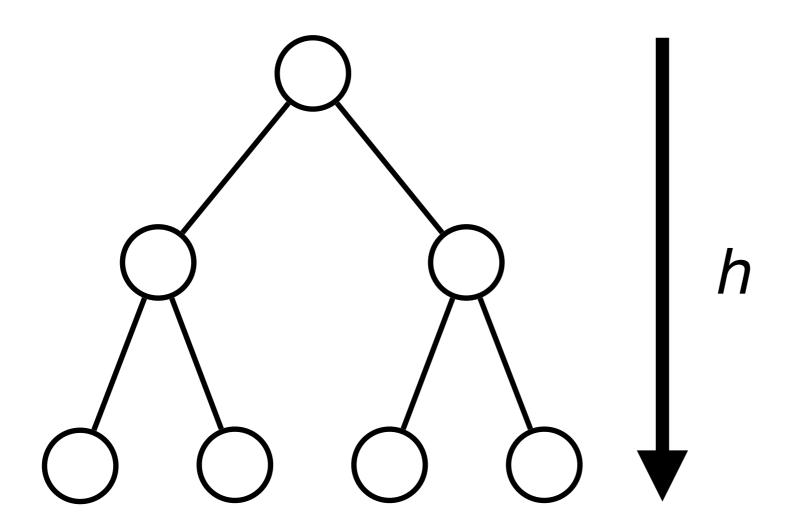
Temps de recherche



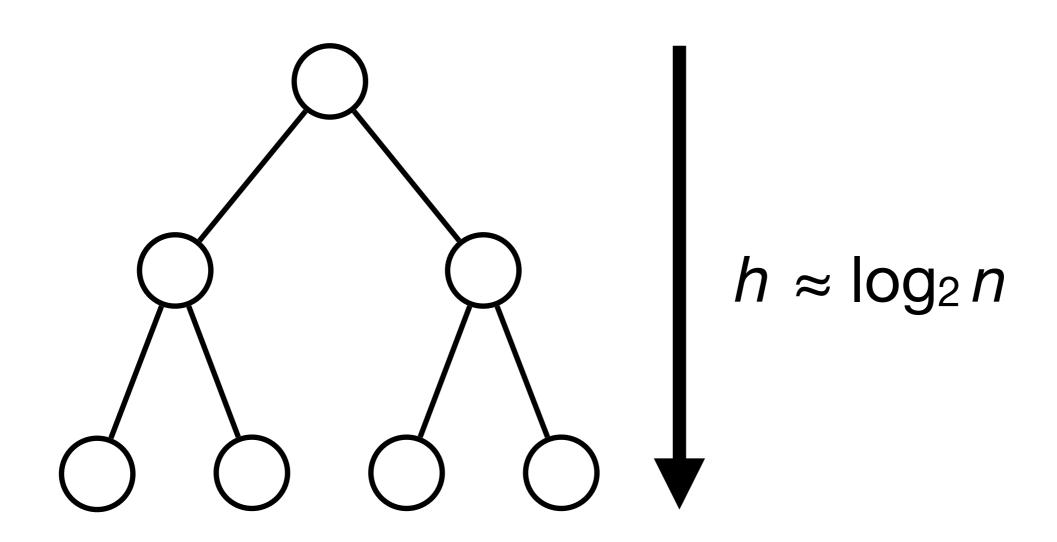
Temps de recherche



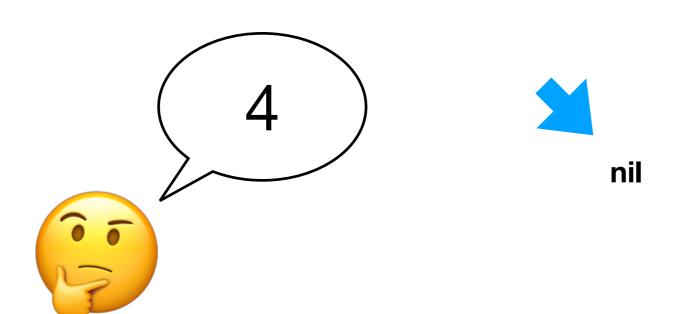
Pour un ABR balancé

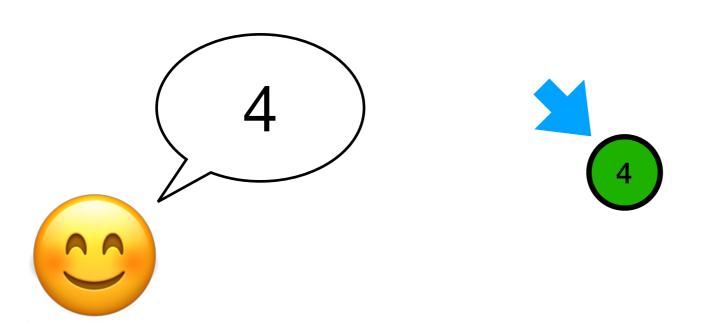


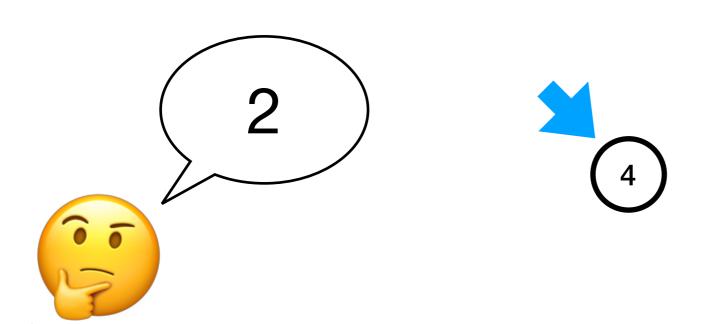
Pour un ABR balancé

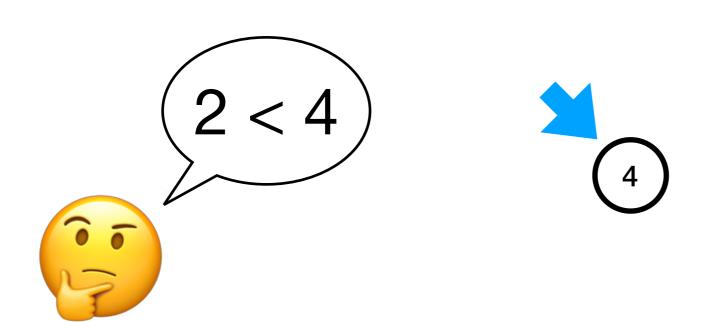


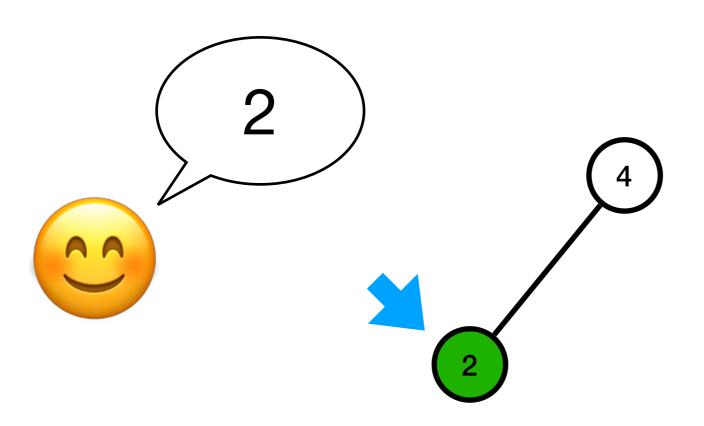
nil

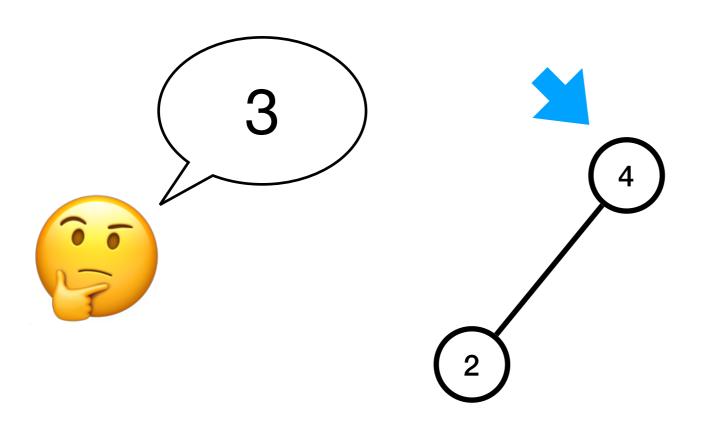


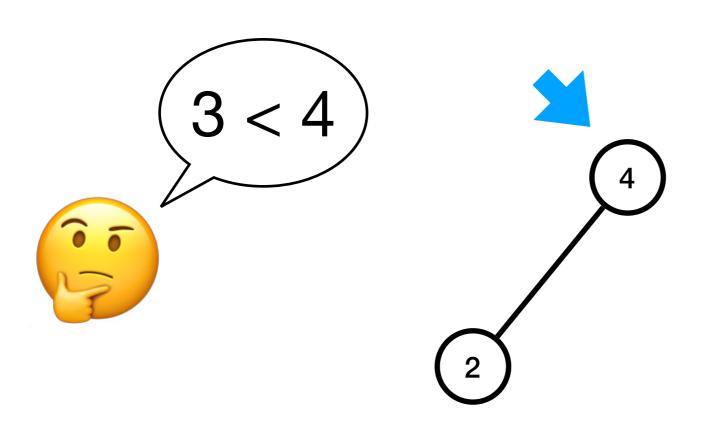


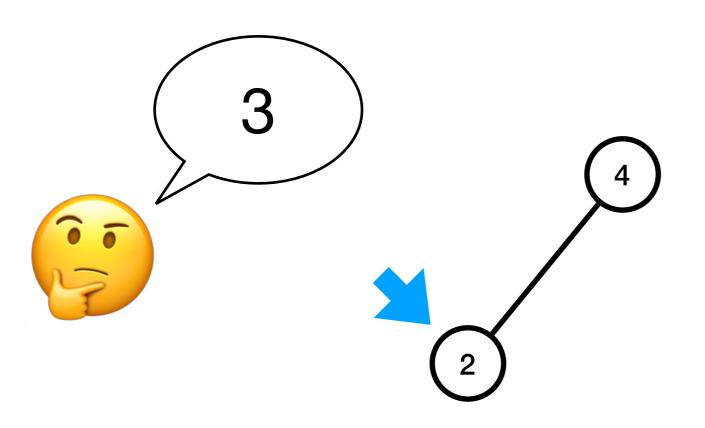


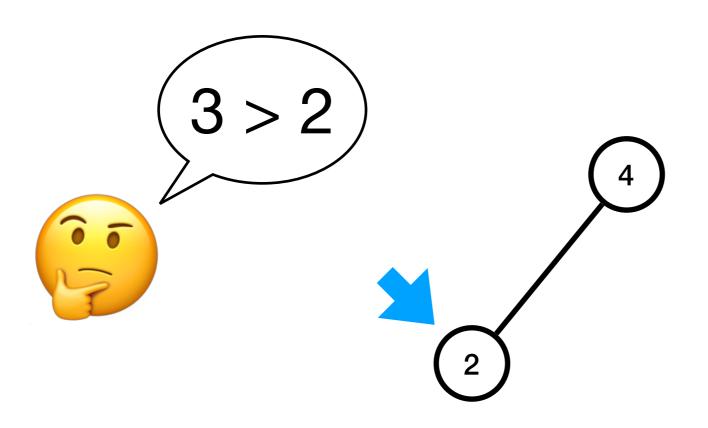


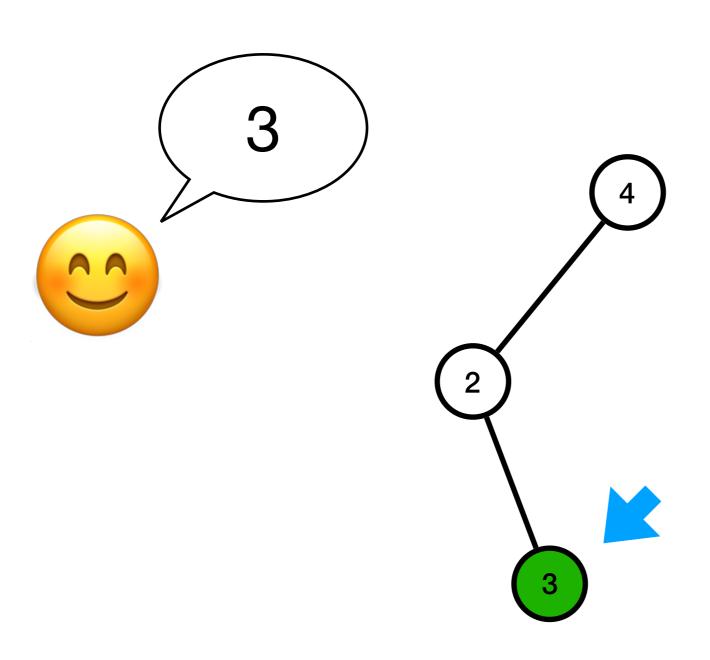


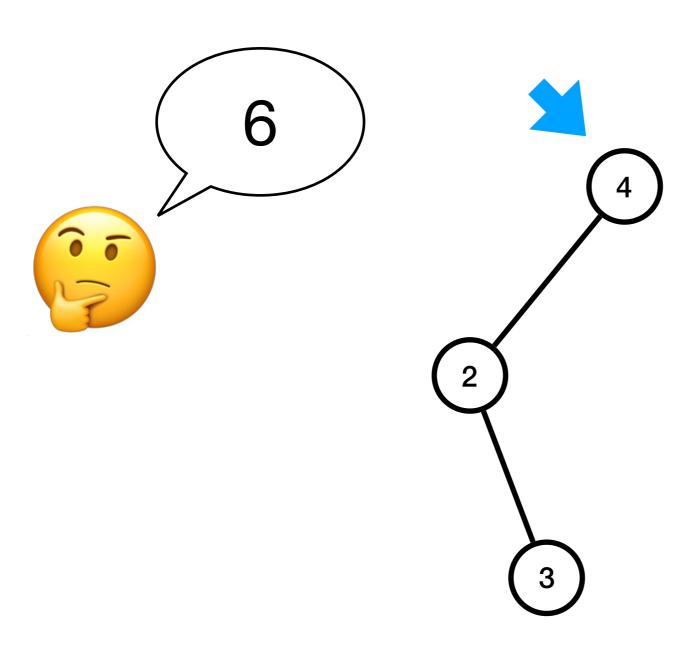


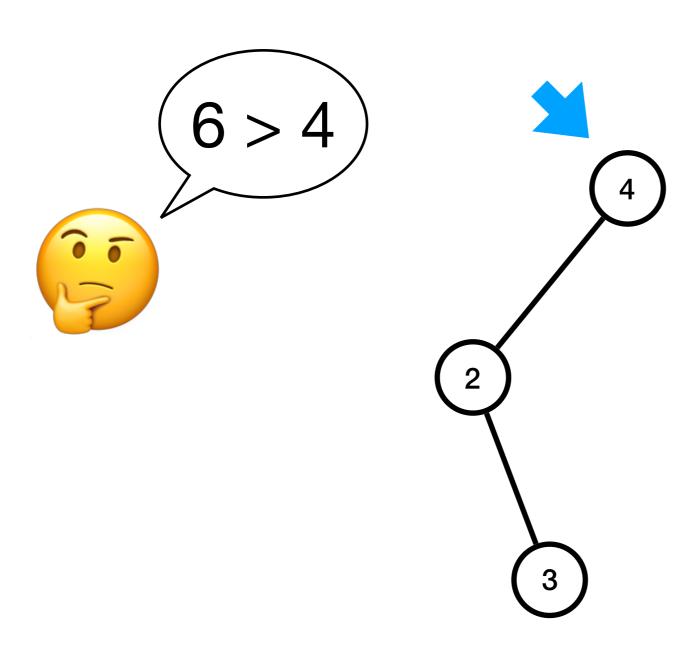


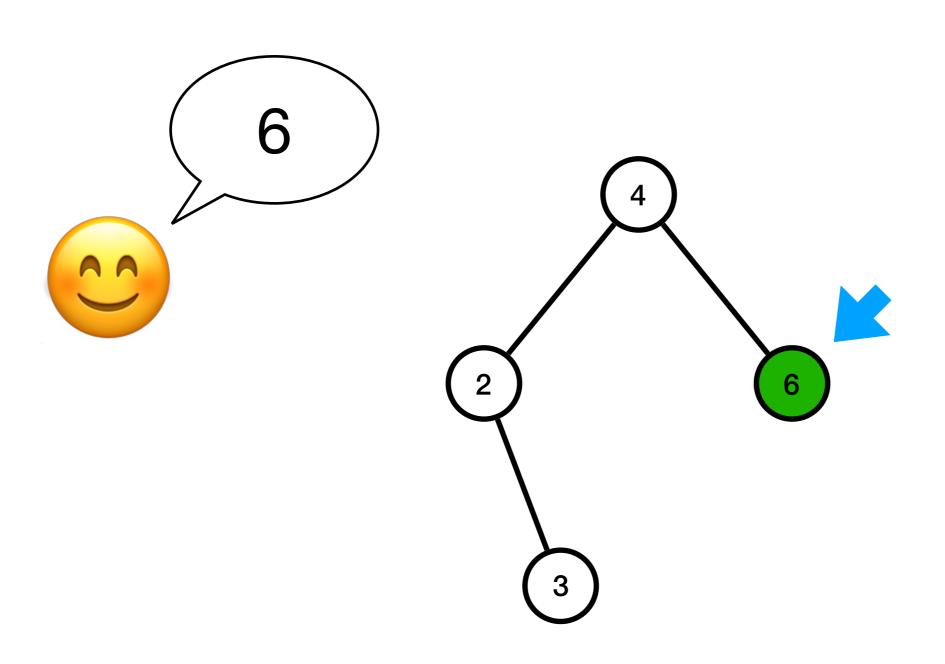


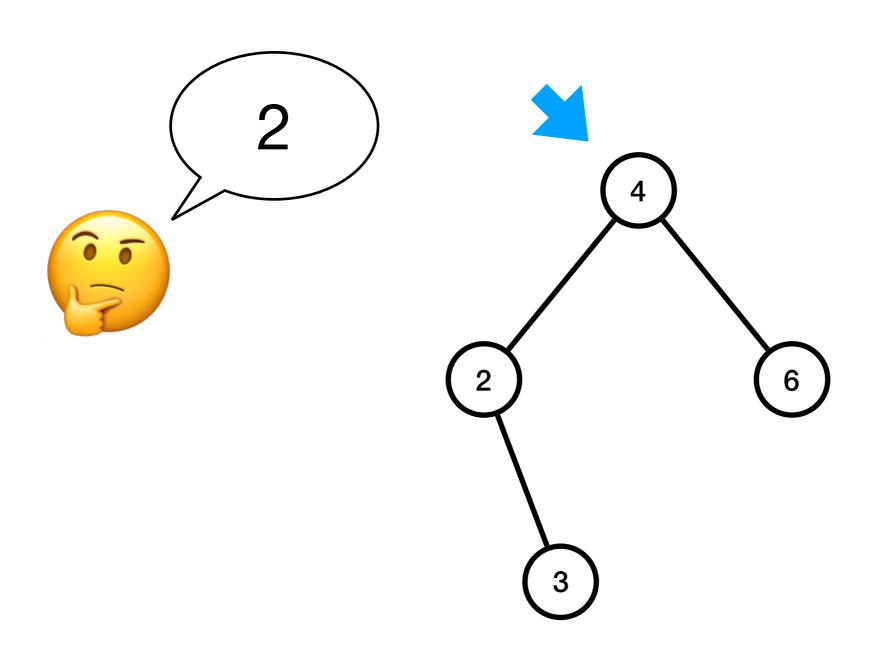


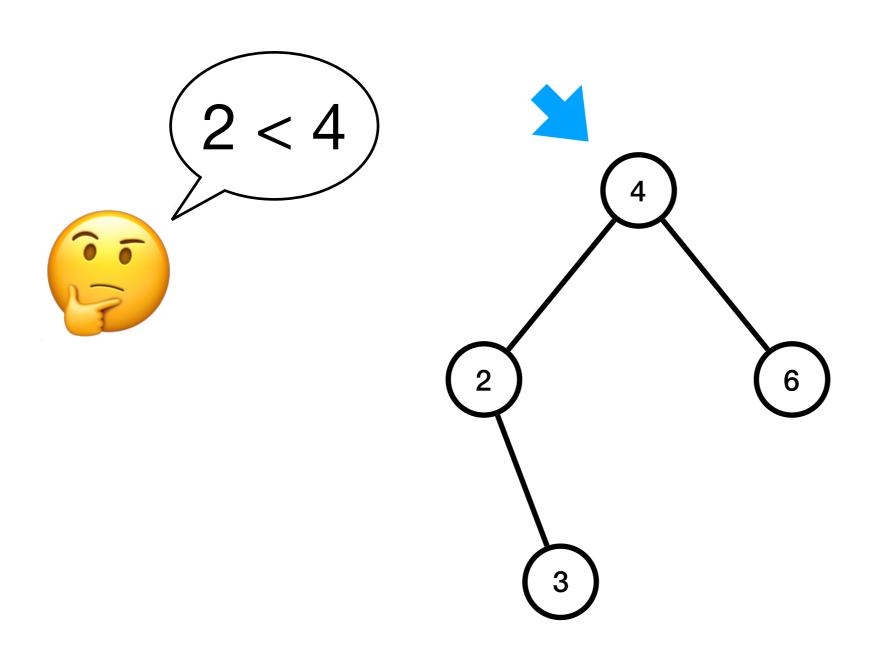


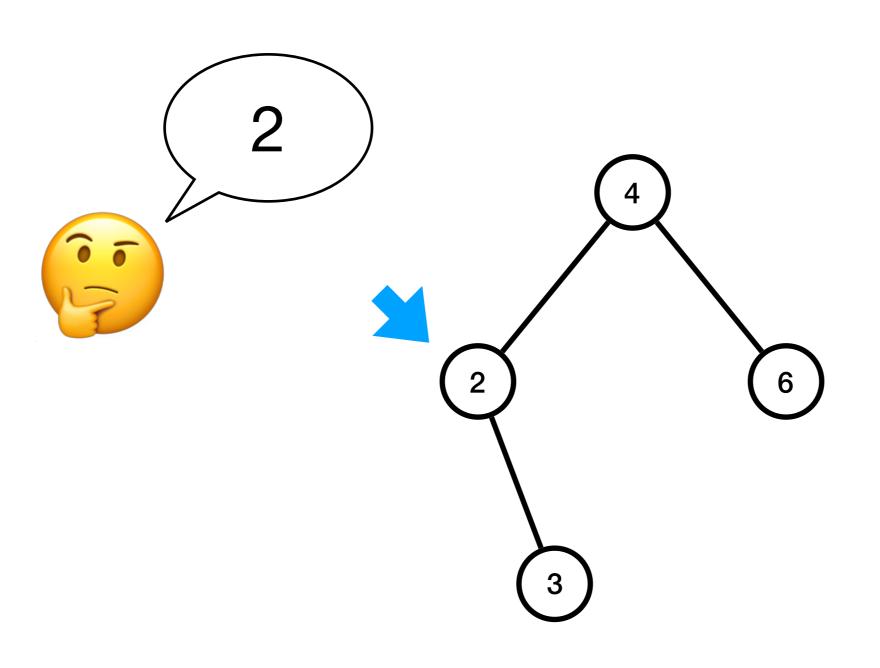


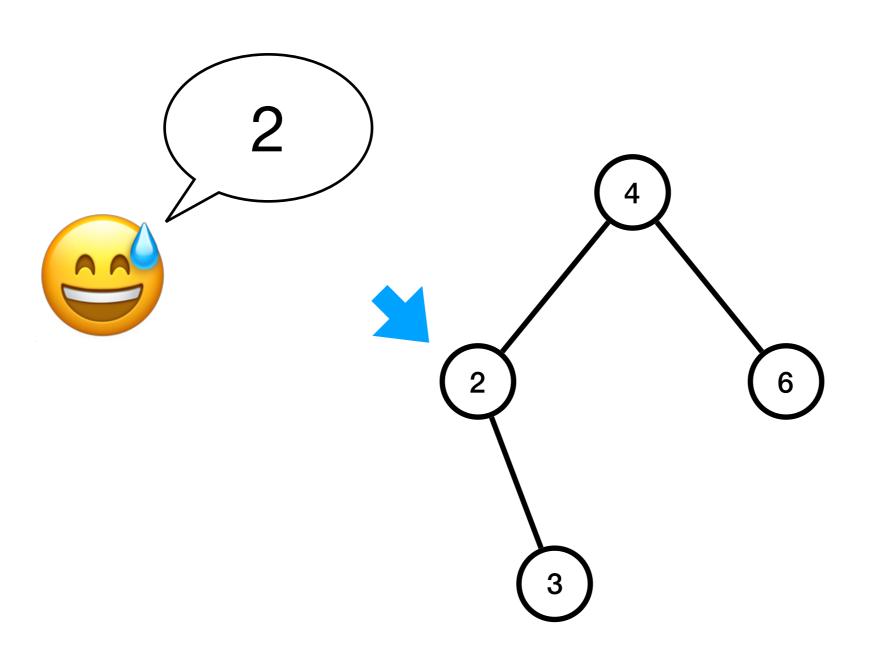


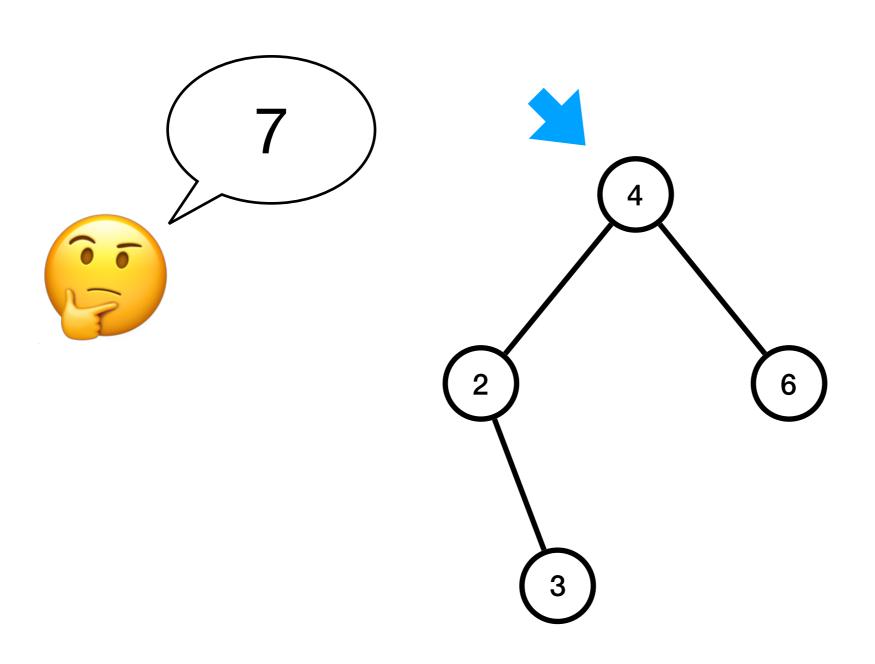


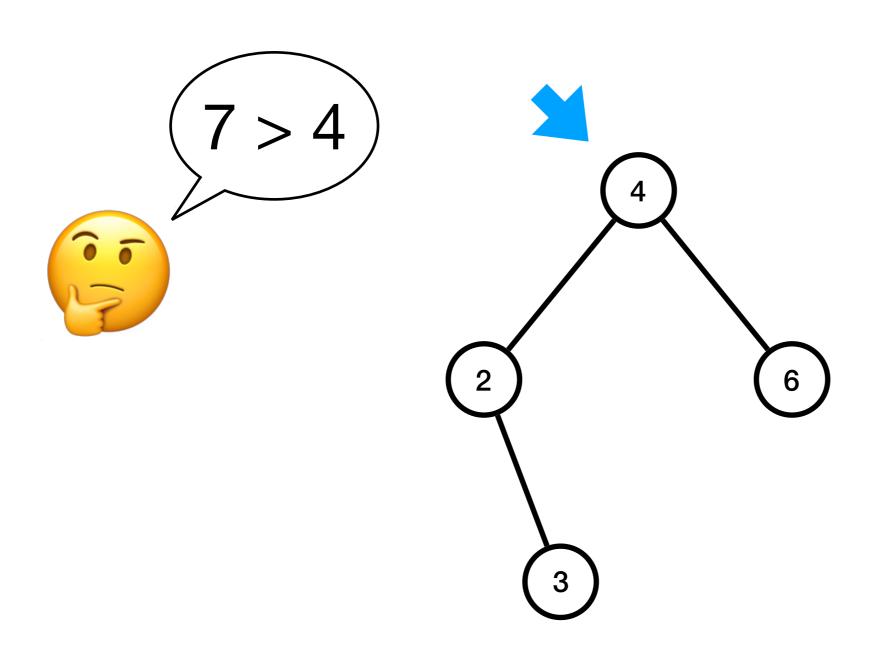


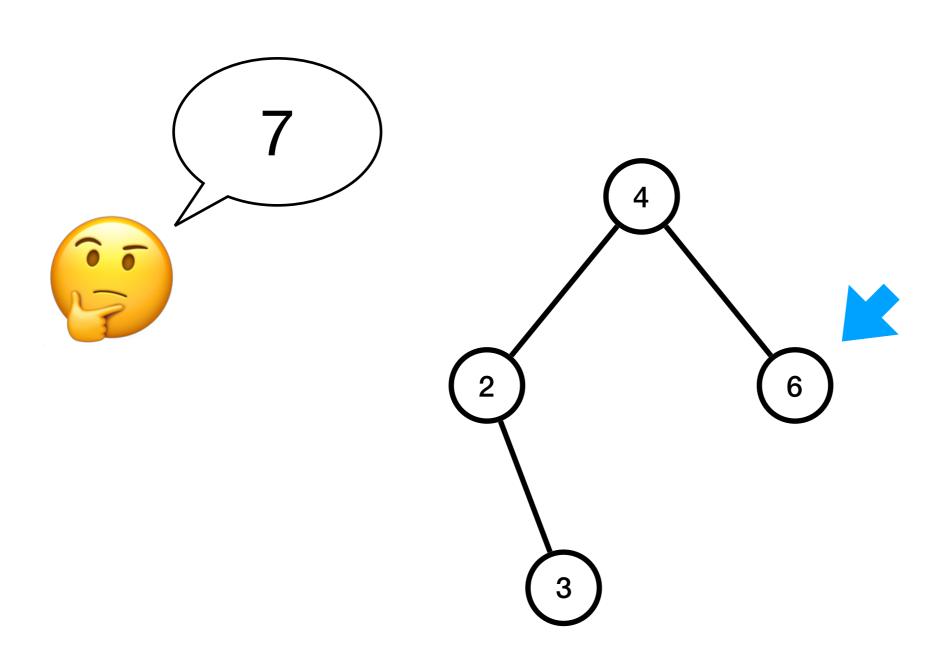


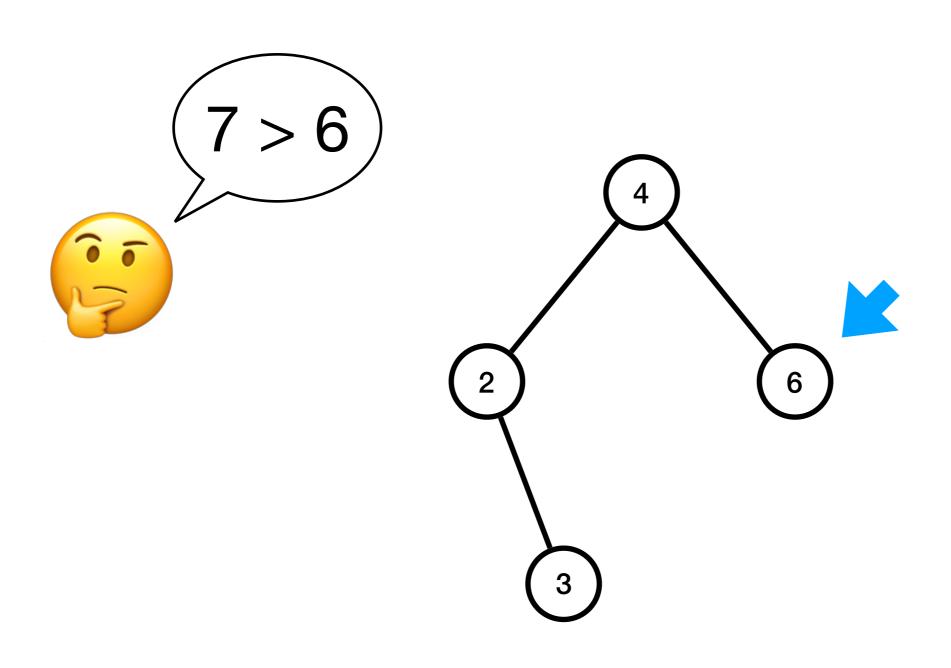


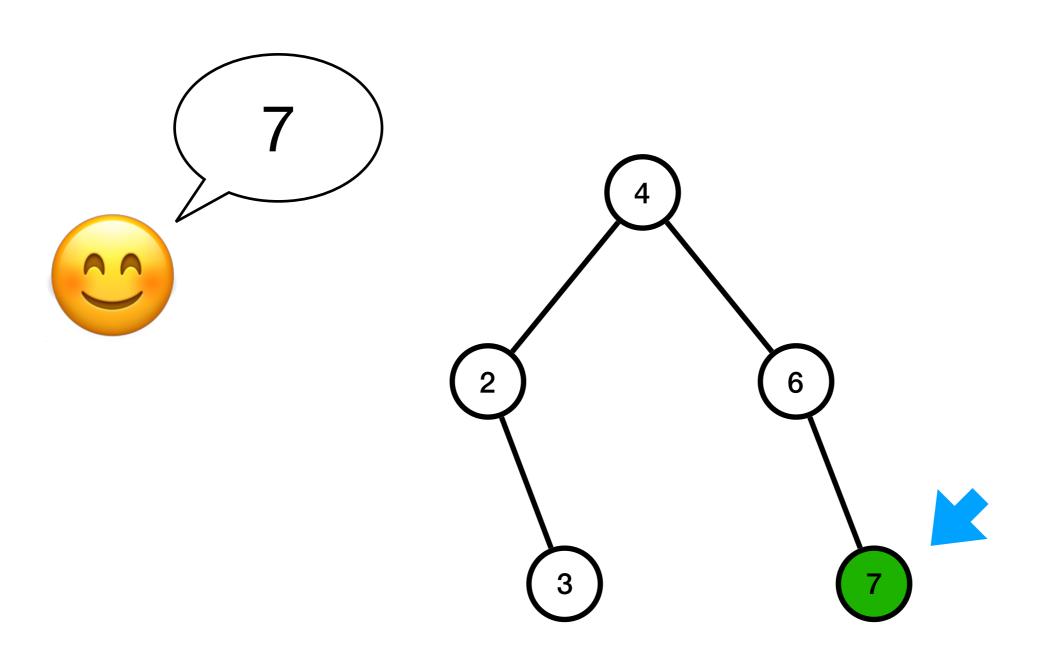


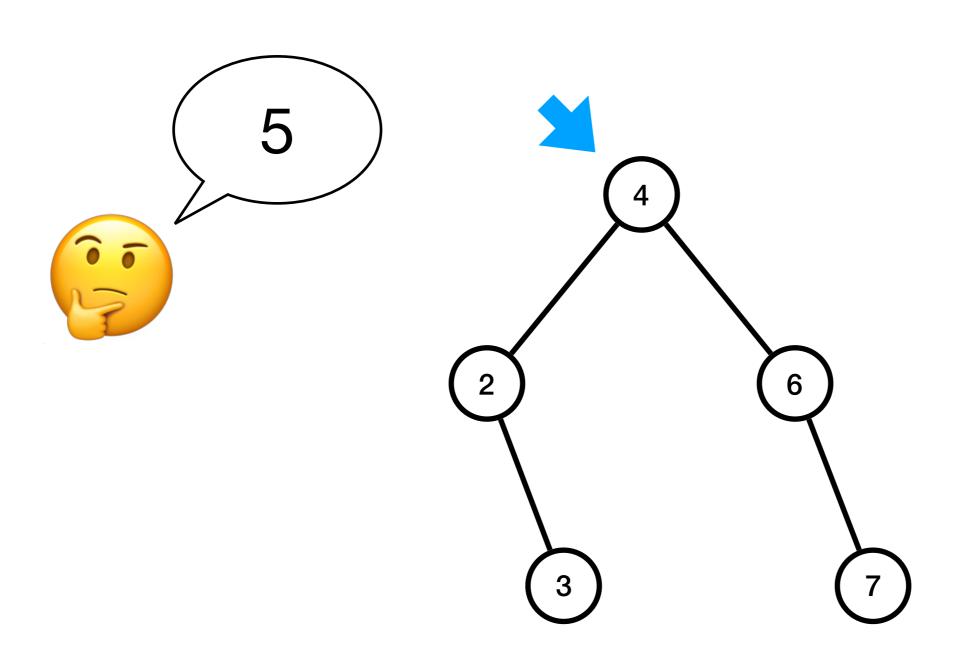


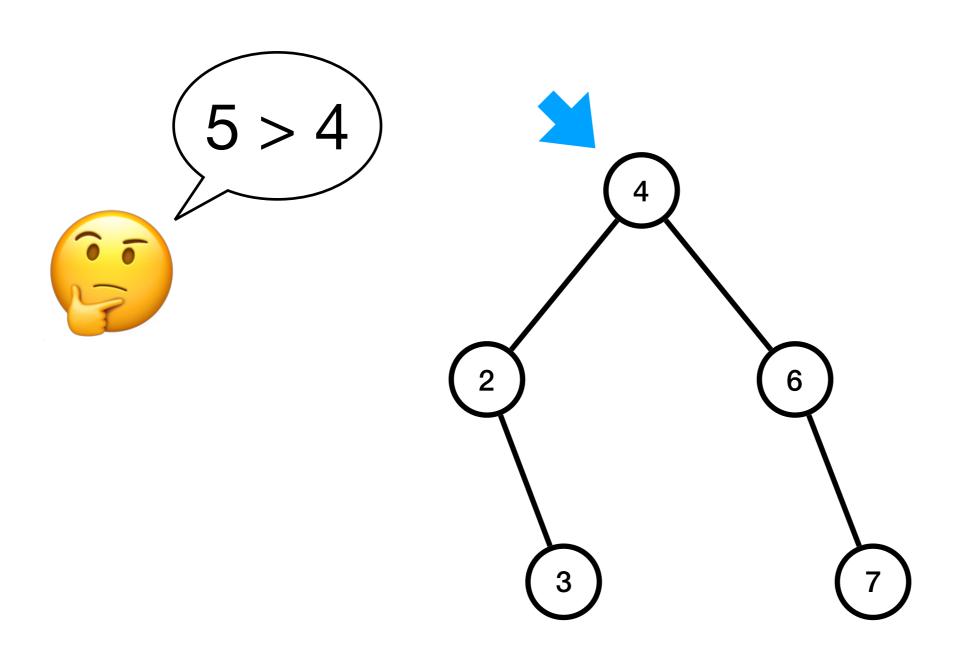


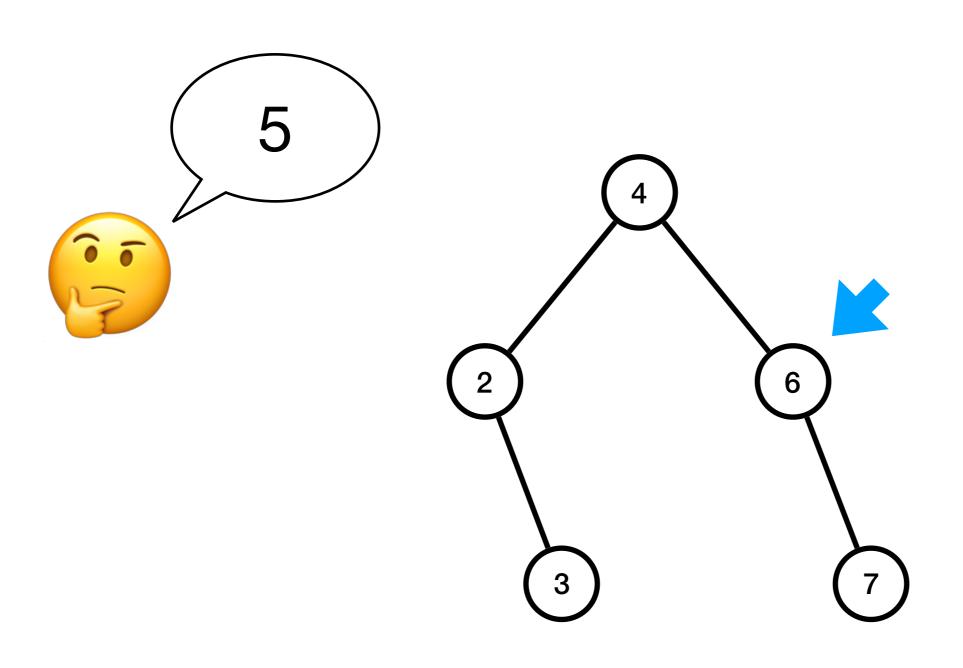


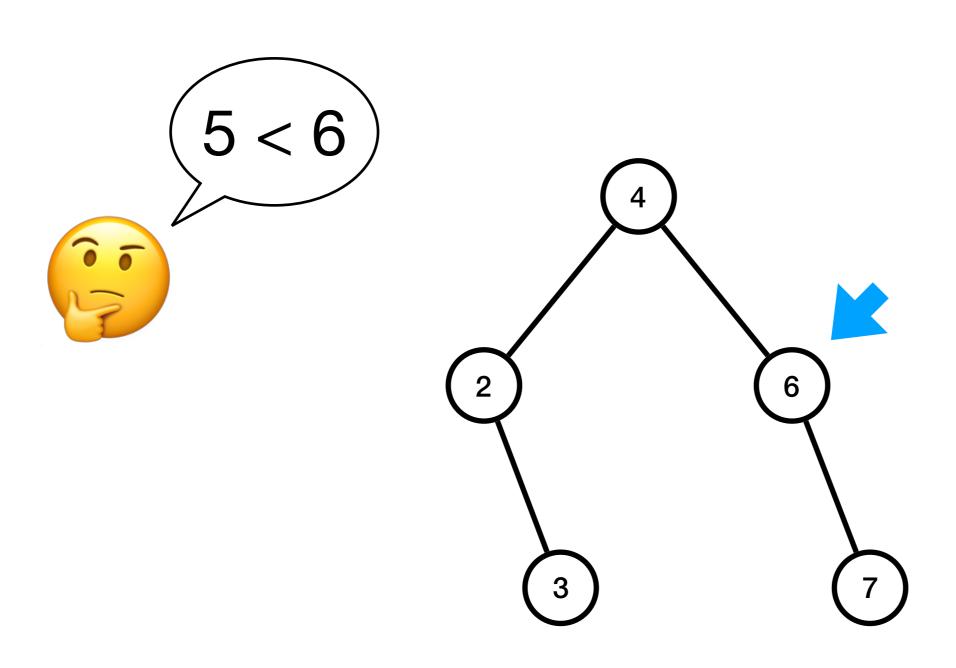


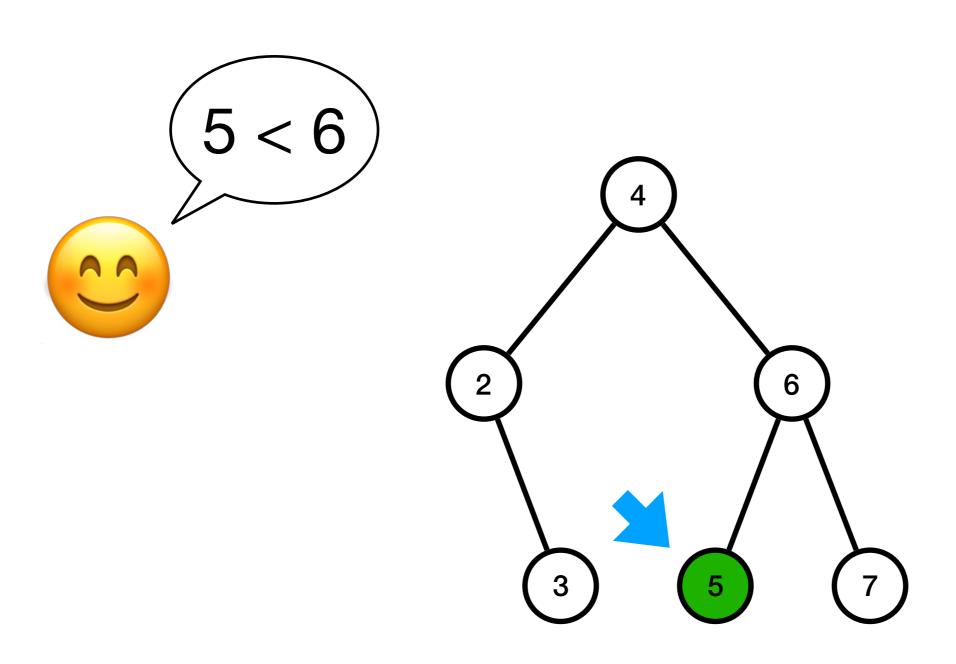


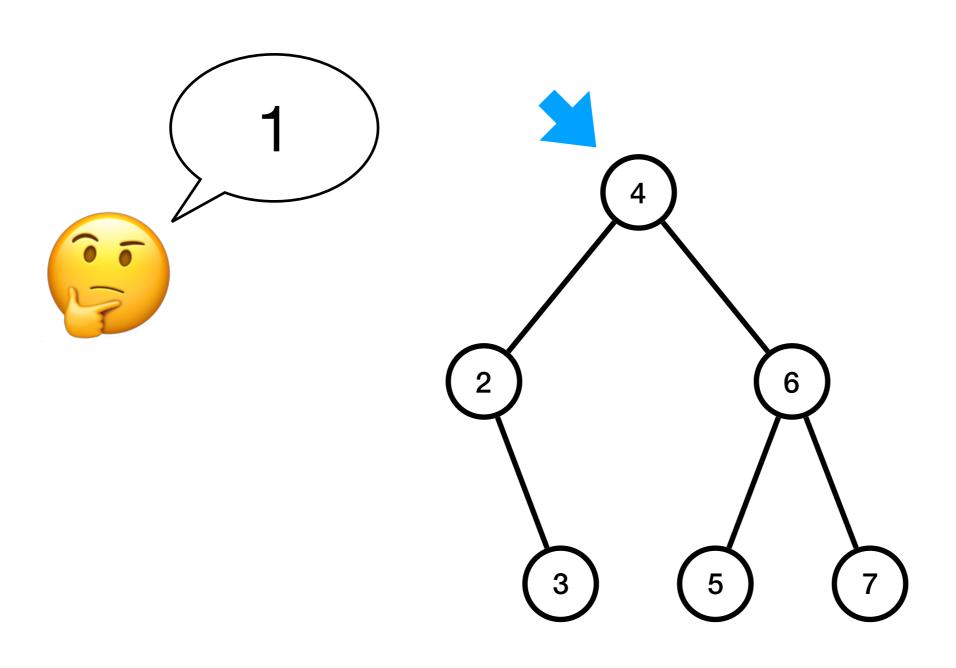


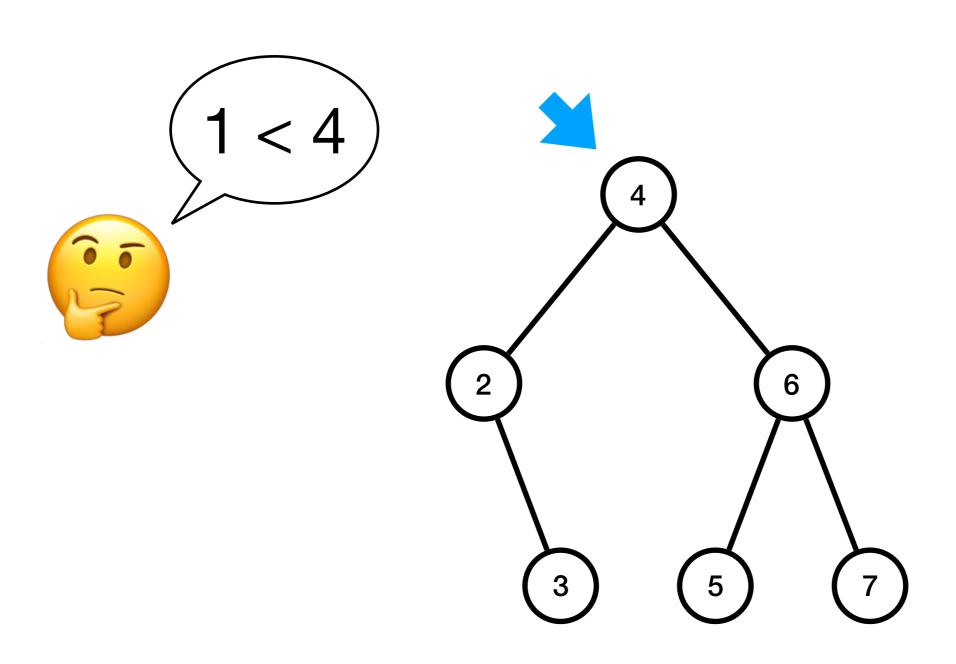


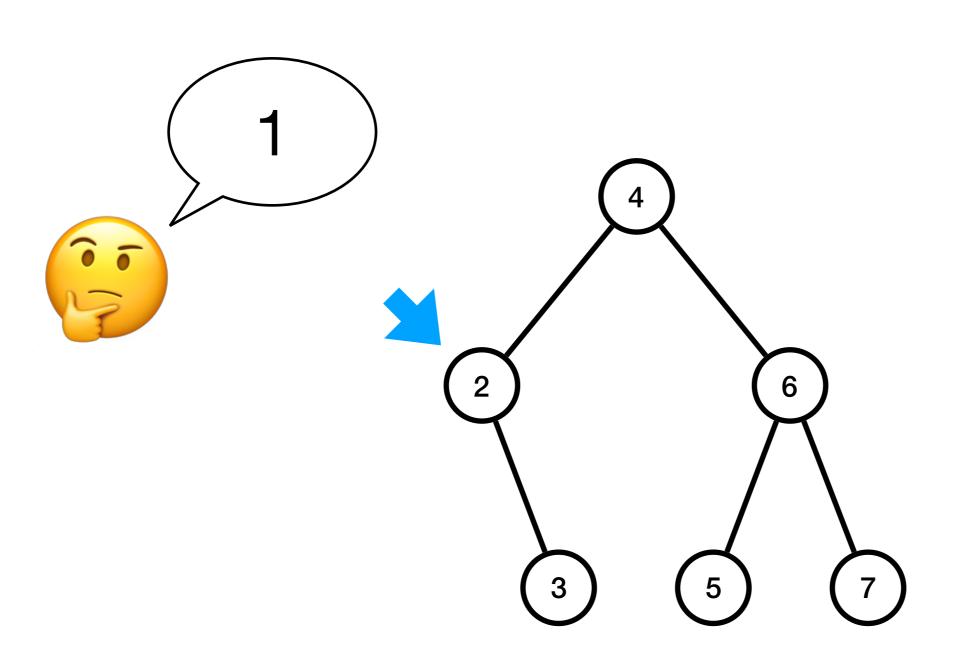


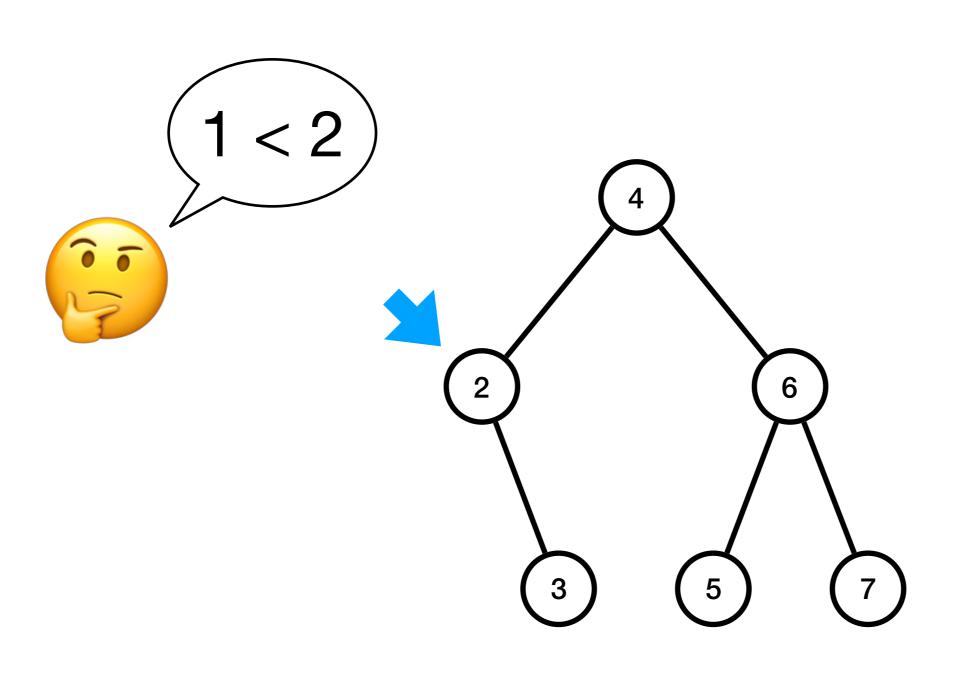


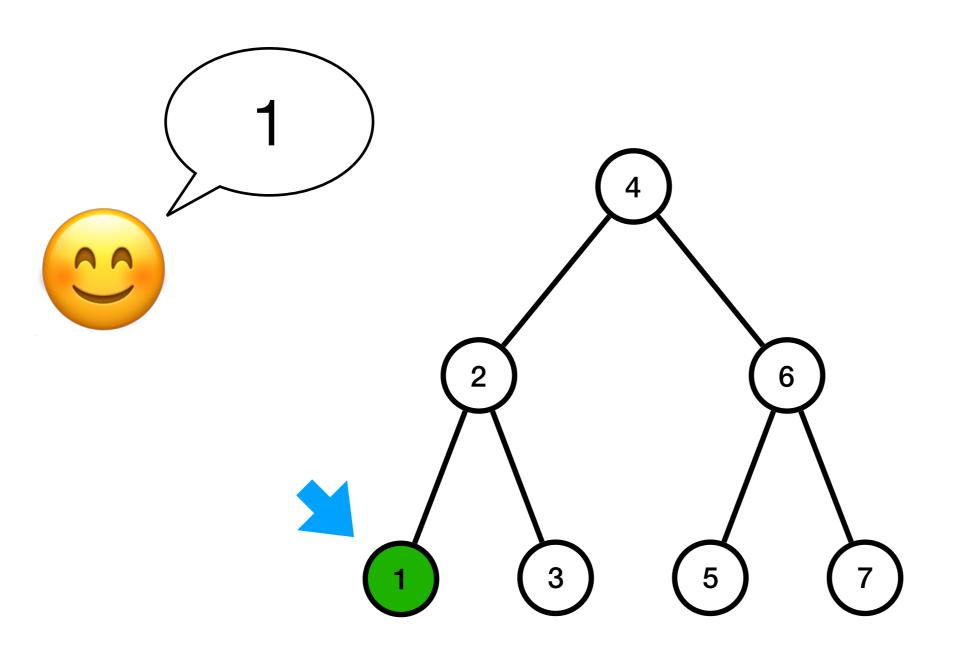


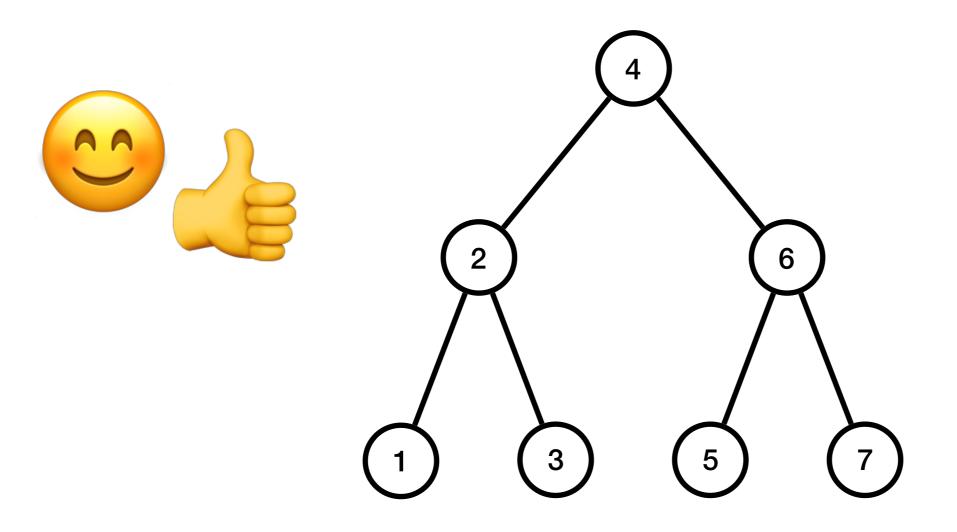












nil

