Game Playing

(my favourite topic so far!)

Game Playing

It's your turn to play the next move!

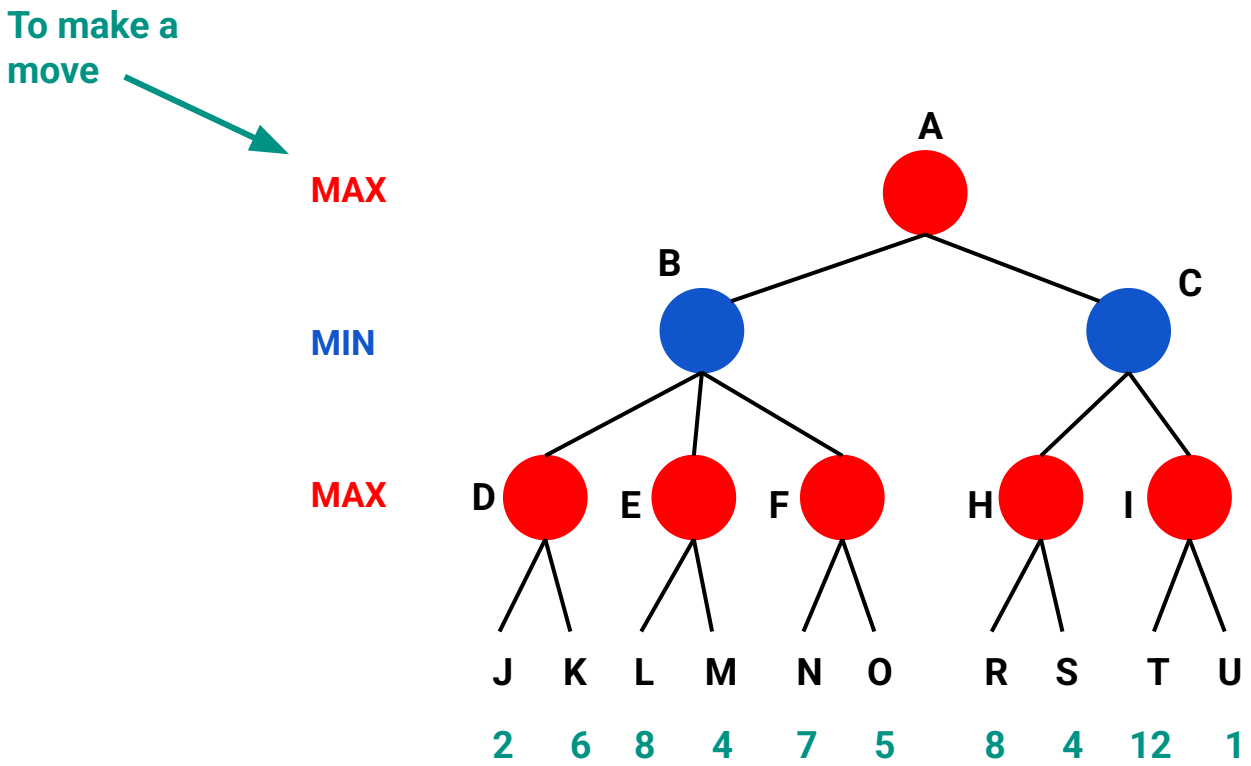
TODO:

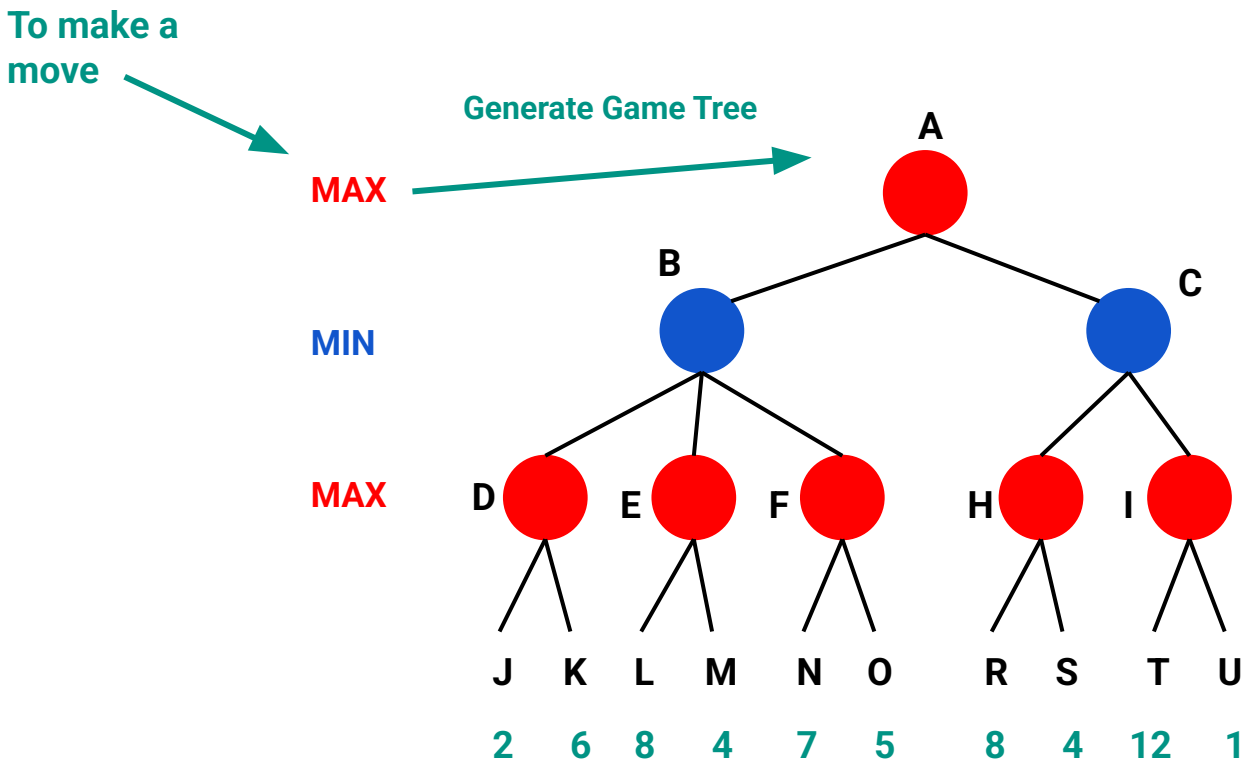
1. Generate the **game tree**
2. **Search** for the best move
3. WIN!

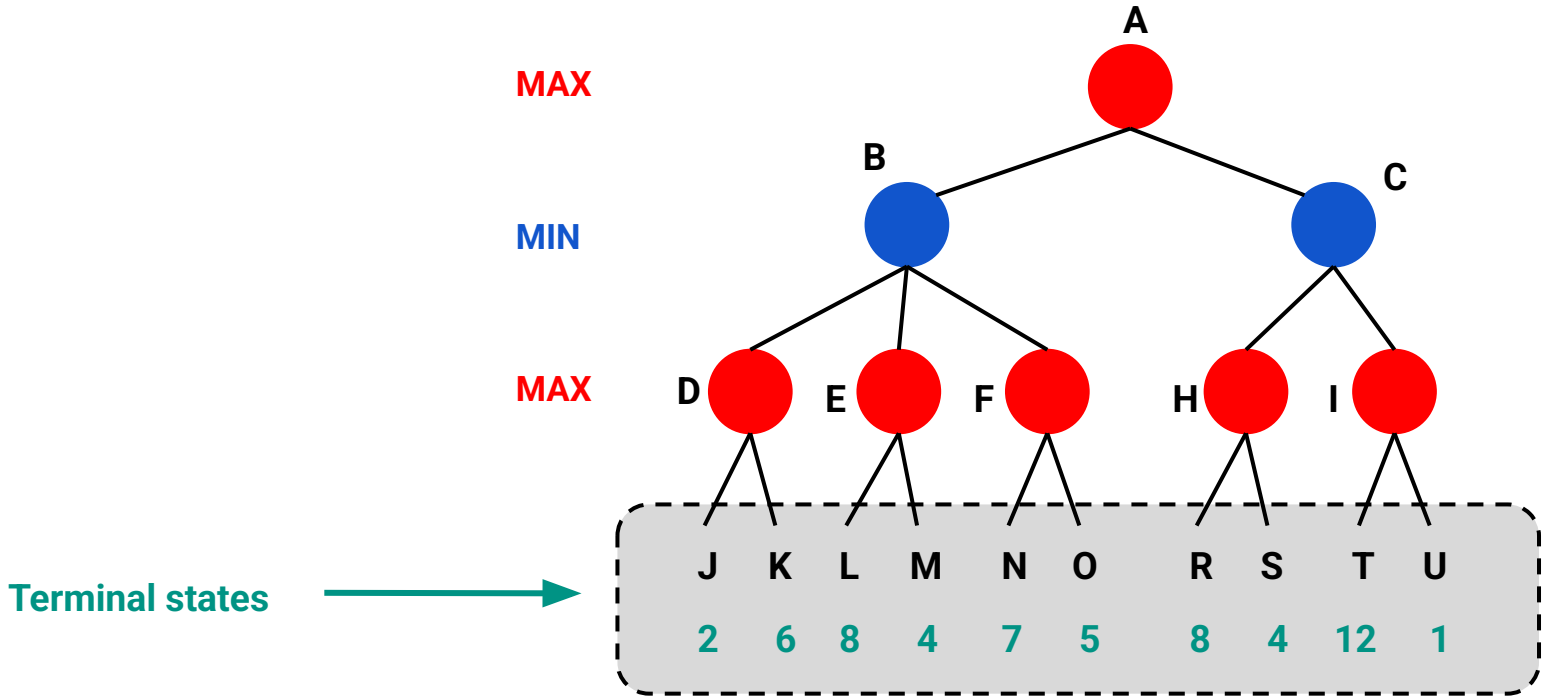
Assumption: Your opponent is smart, they will always pick the best move (i.e. optimal move).

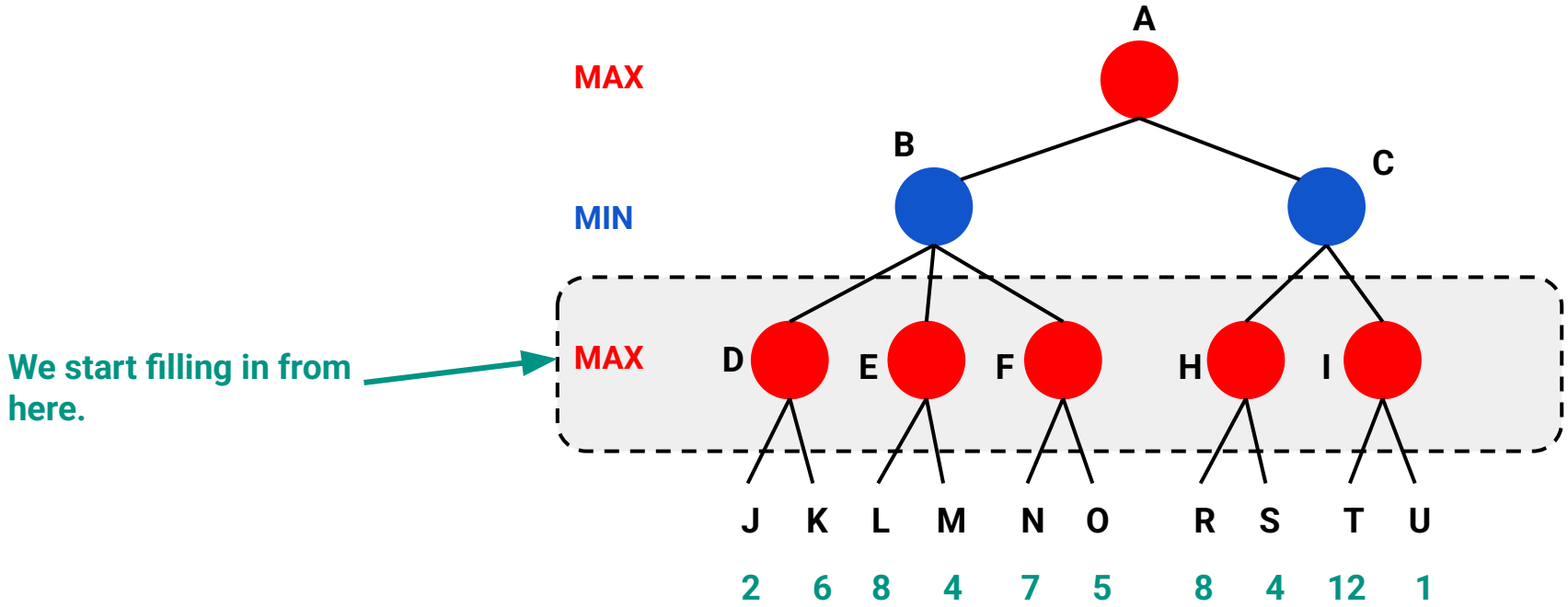
Exercise 2 a

Minimax Algorithm

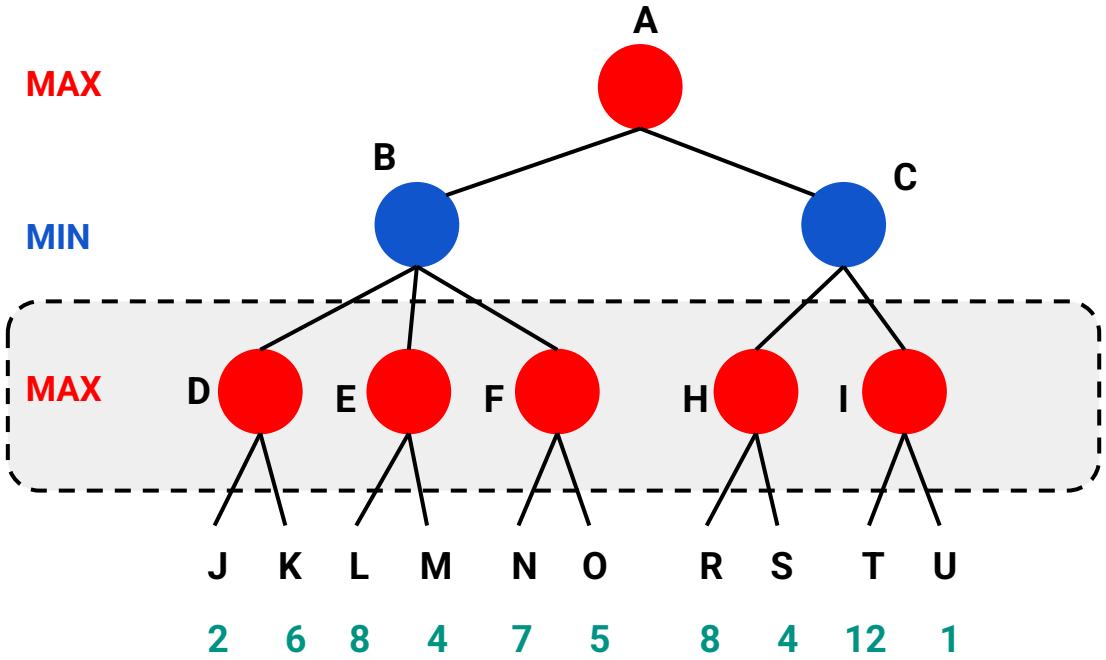

To make a move
MAX
MIN
MAX
A
B
C
D
E
F
H
I
J
K
L
M
N
O
R
S
T
U
2
6
8
4
7
5
8
4
12
1


To make a move
Generate Game Tree
MAX
MIN
MAX
A
B
C
D
E
F
H
I
J
K
L
M
N
O
R
S
T
U
2
6
8
4
7
5
8
4
12
1

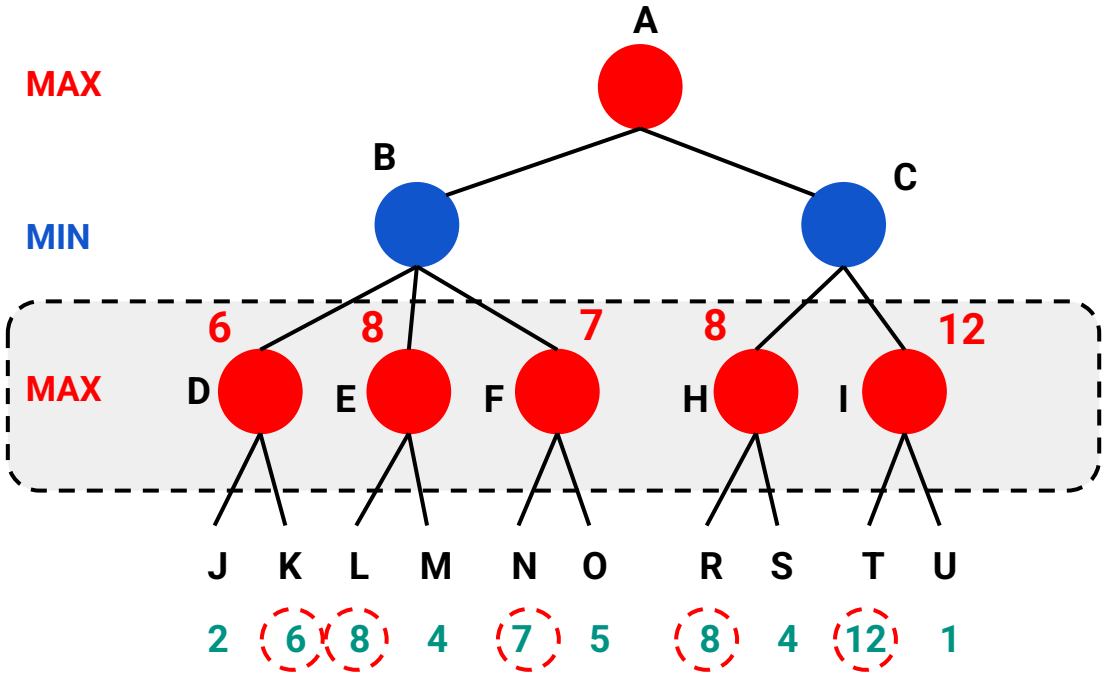

MAX
MIN
MAX
A
B
C
D
E
F
H
I
J K L M N O R S T U
2 6 8 4 7 5 8 4 12 1
Terminal states


MAX
MIN
MAX
We start filling in from here.
A
B
C
D
E
F
H
I
J K L M N O R S T U
2 6 8 4 7 5 8 4 12 1

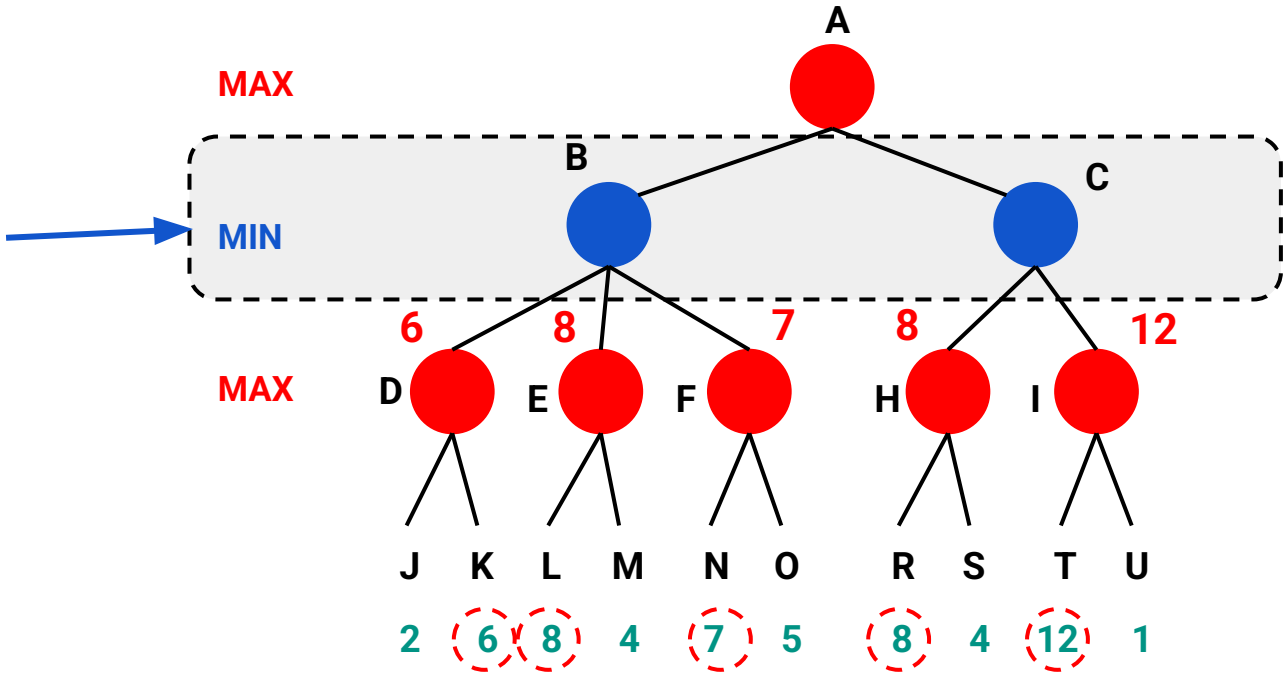
Take the max of the children's values



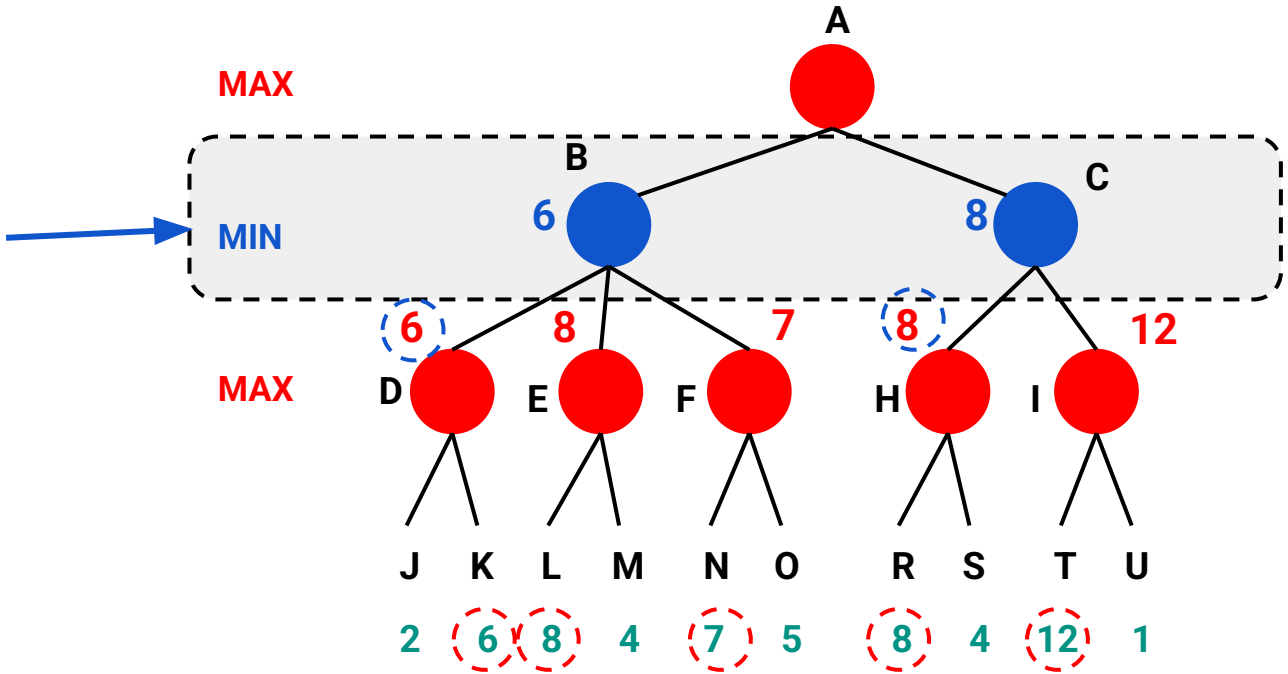
Take the max of the children's values



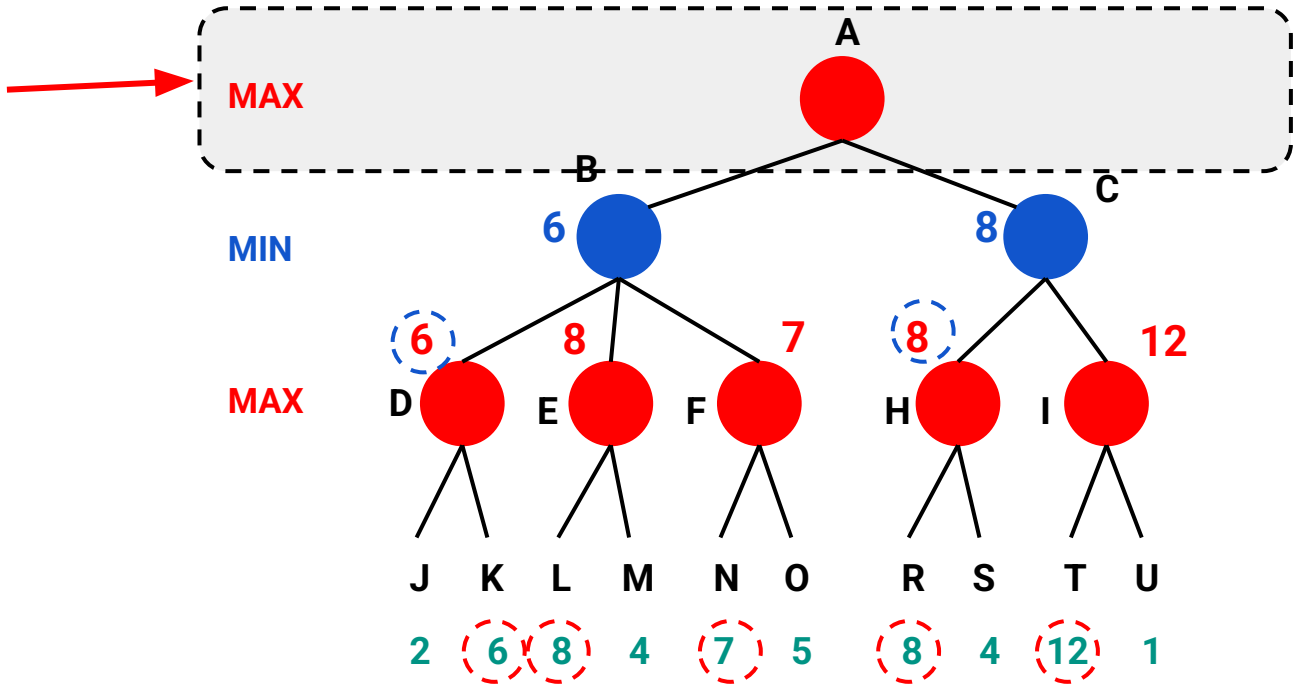
Take the min of the children's values



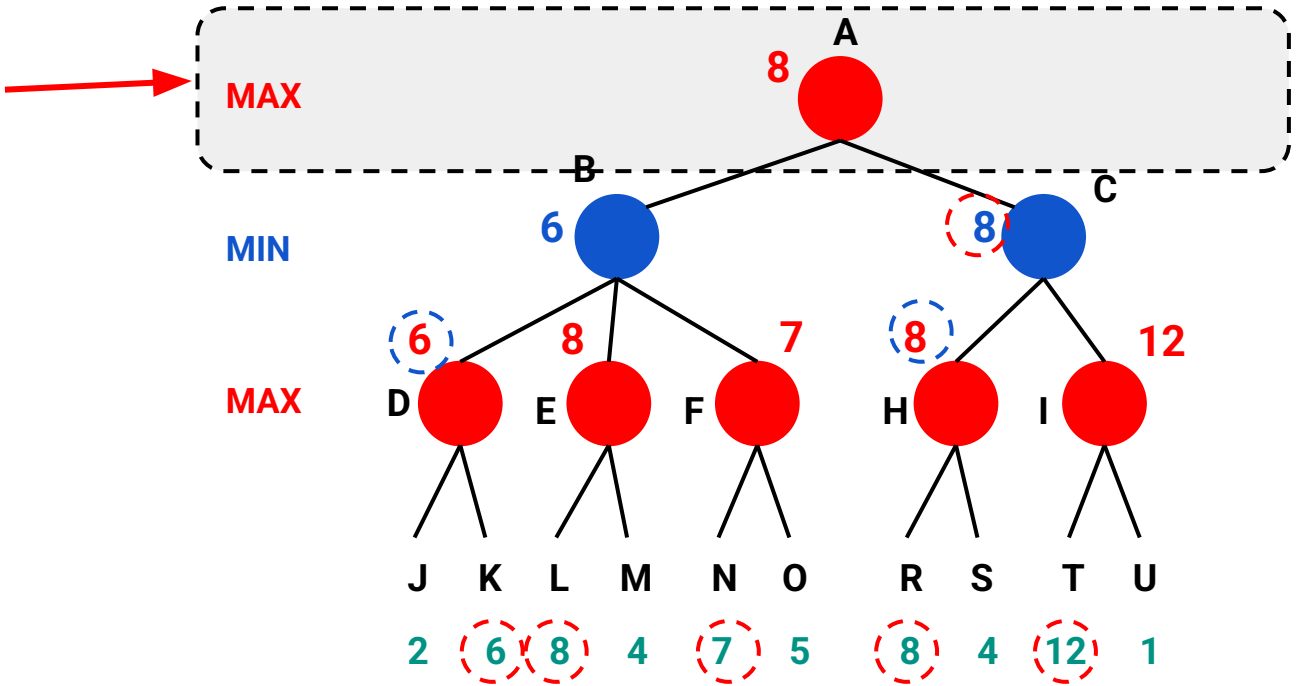
Take the min of the children's values



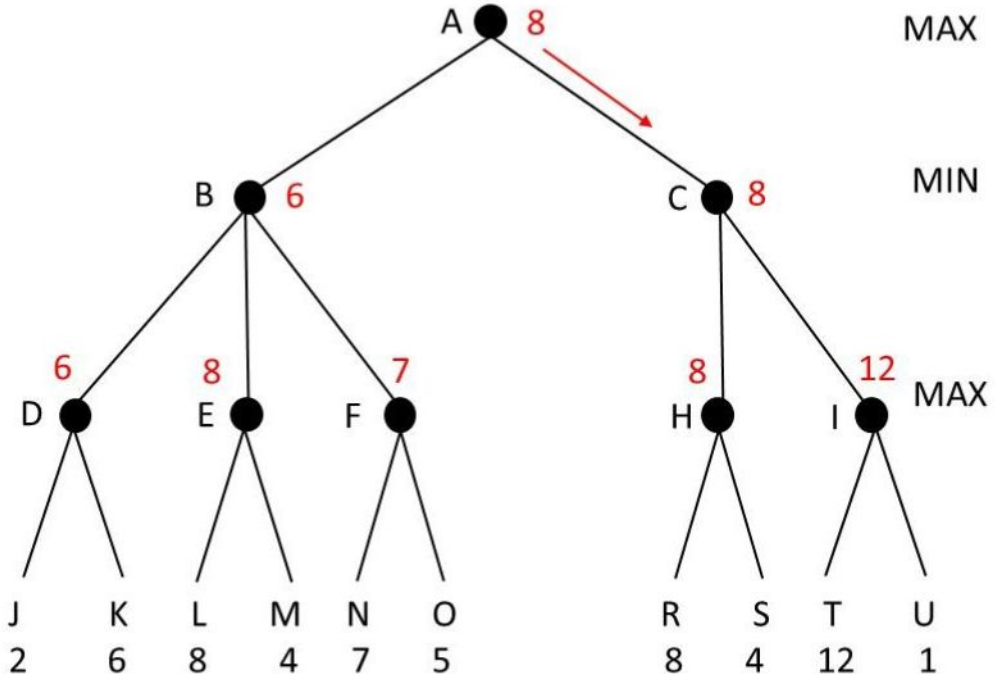
Take the max of the children's values



Take the max of the children's values



The backed-up values are shown in red. Max should choose C.



Exercise 2 b

Alpha-Beta Algorithm

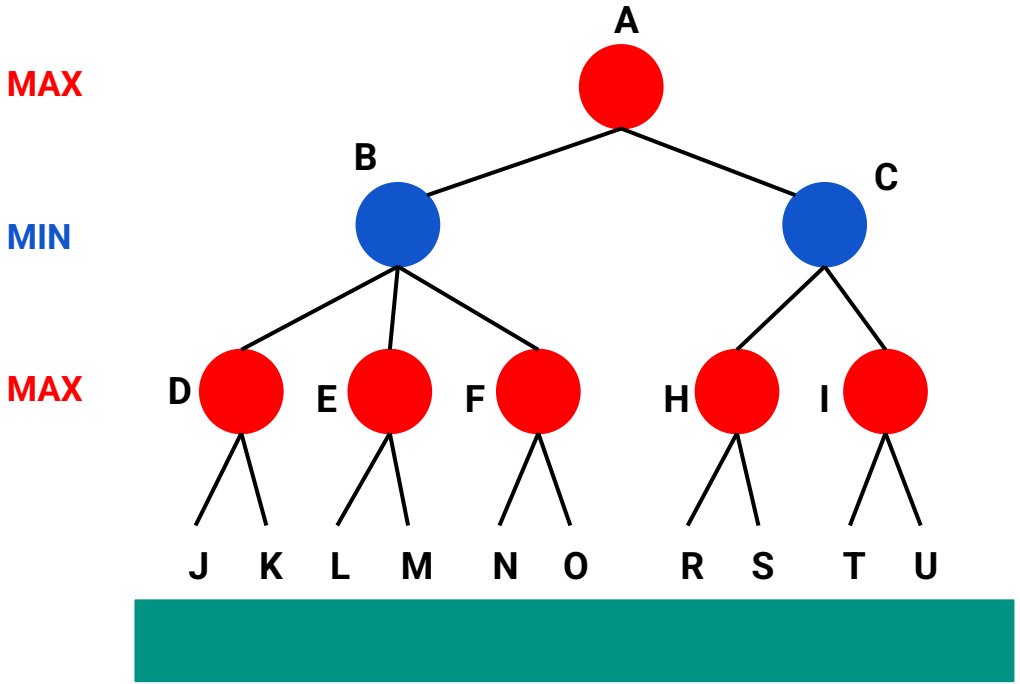
Alpha-beta pruning

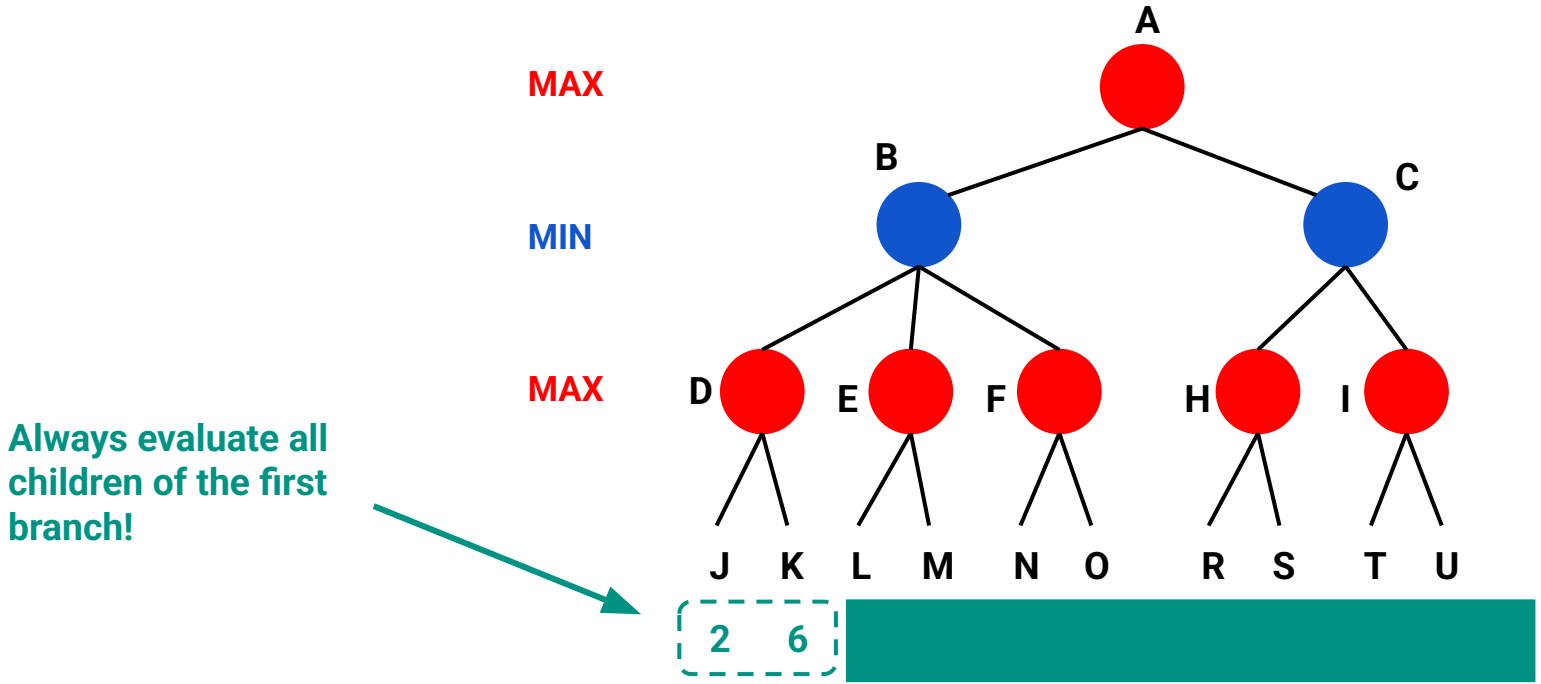
- **Produce the same result as Minimax**
- **Purpose:** Generate less nodes

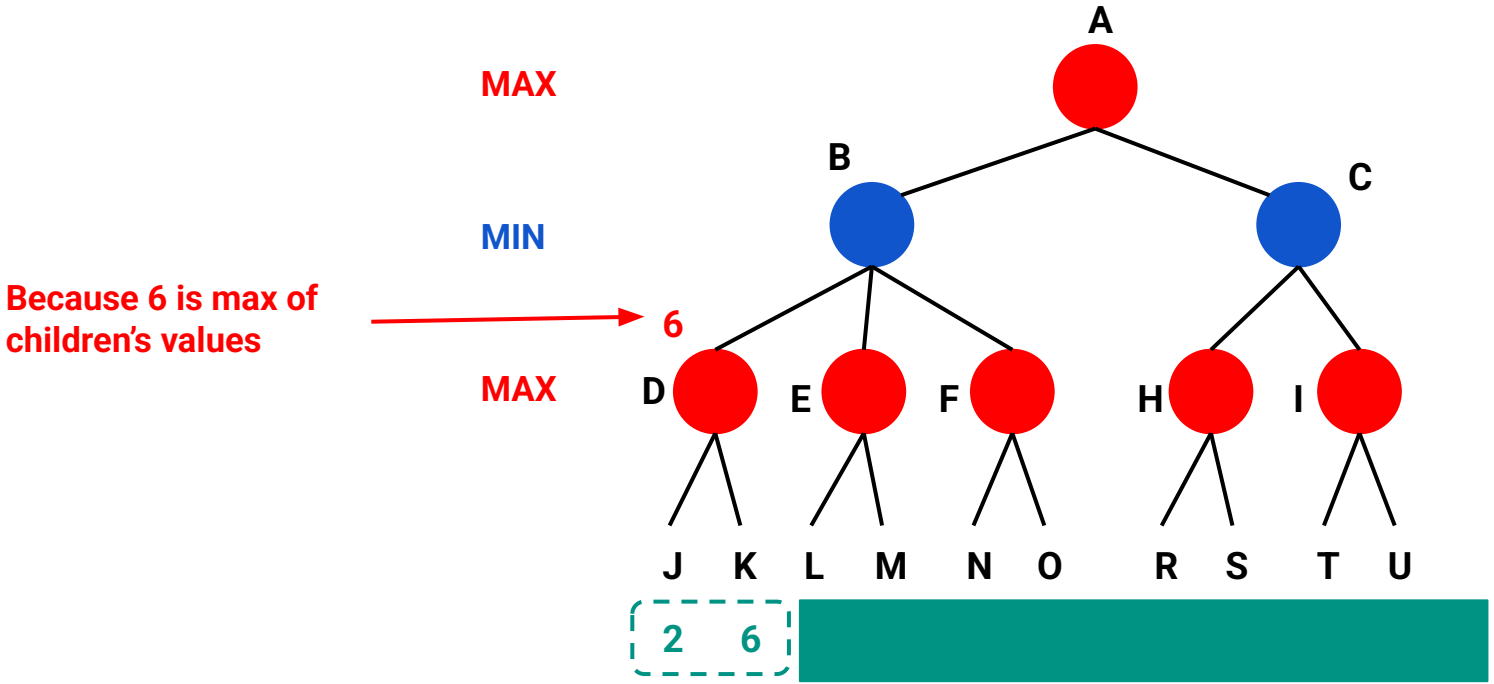
  => Save time, memory

Pretend that we don't know any value.

We'll reveal them one by one.

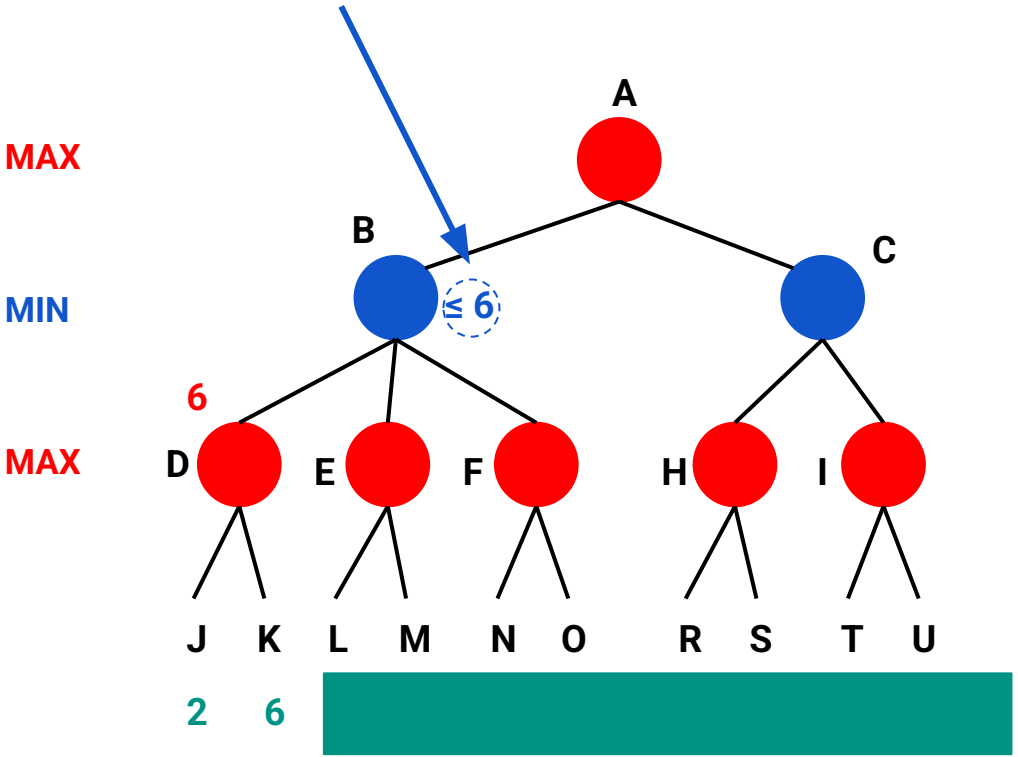


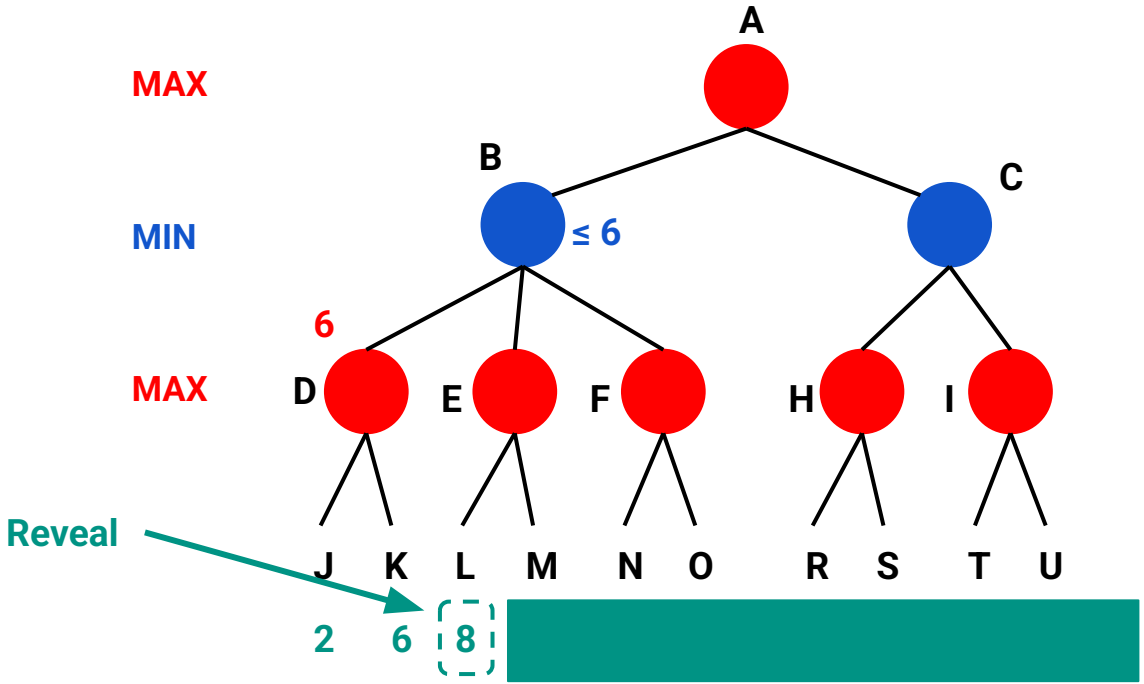

MAX
MIN
MAX
A
B
C
D
E
F
H
I
J
K
L
M
N
O
R
S
T
U
2
6
Always evaluate all children of the first branch!


MAX
MIN
MAX
Because 6 is max of children's values
6
A
B
C
D
E
F
H
I
J
K
L
M
N
O
R
S
T
U
2
6

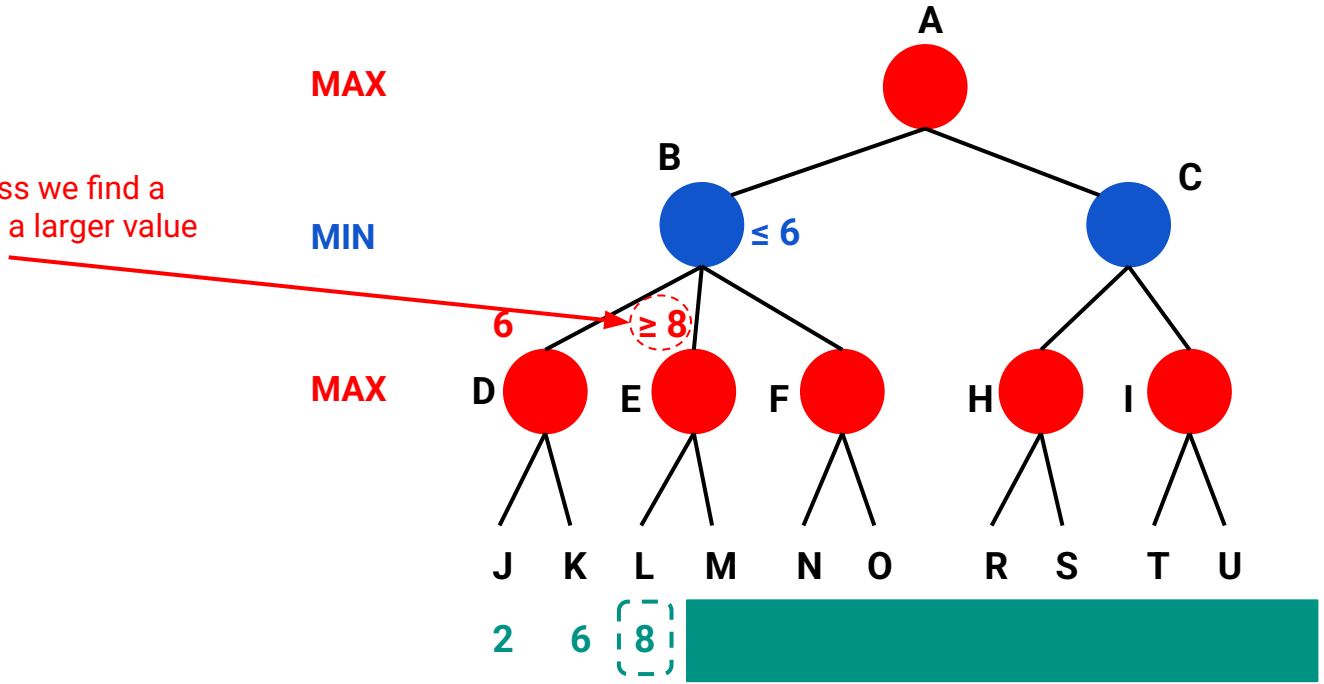
B is at most 6

I.e. B will be 6 unless we find a different child with a smaller value




A
MAX
B
C
MIN
≤ 6
6
MAX
D
E
F
H
I
Reveal
J
K
L
M
N
O
R
S
T
U
2
6
8

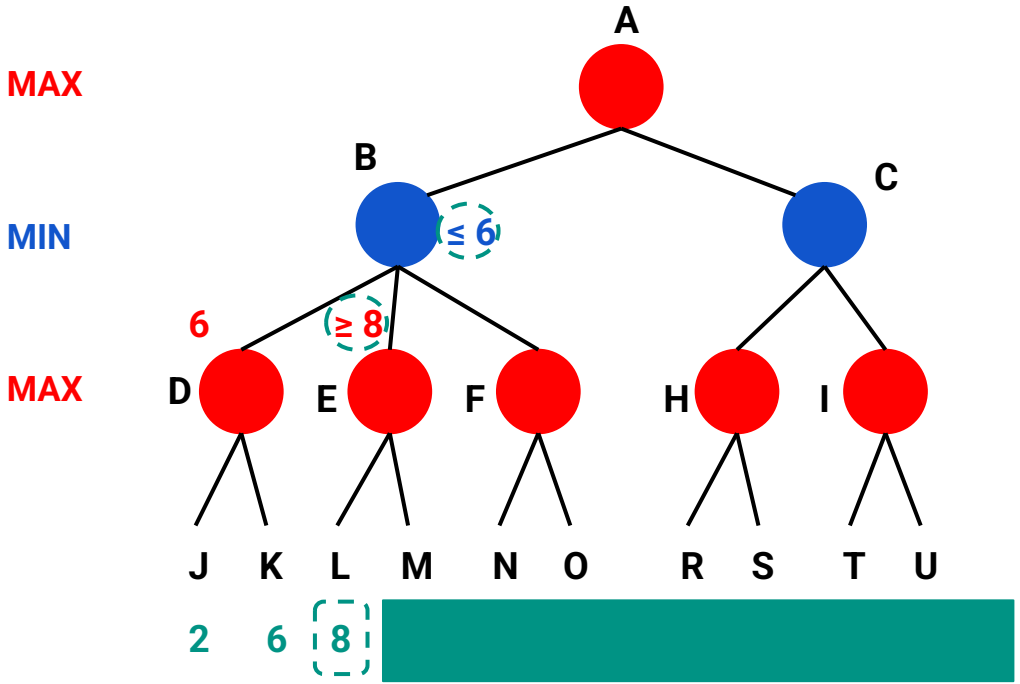
E is at least 8
I.e. E will be 8 unless we find a different child with a larger value

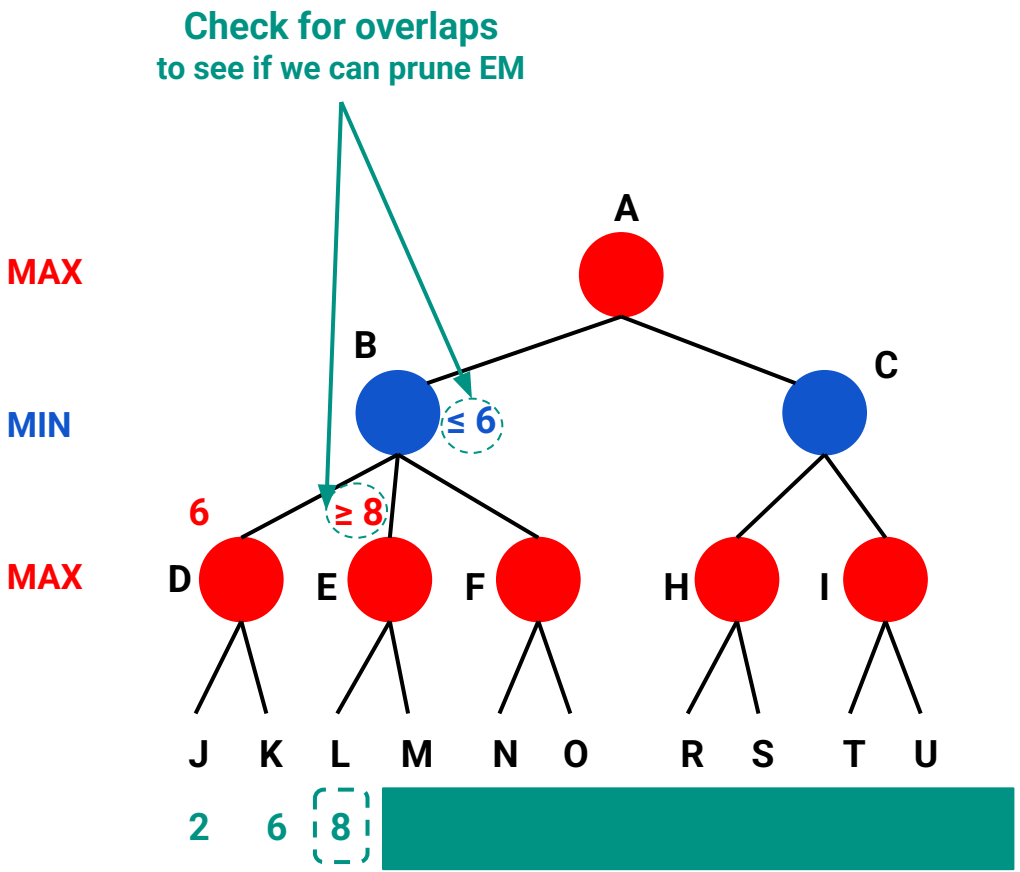


Tips

When writing down the intervals.

- **MIN: ≤**
- **MAX: ≥**

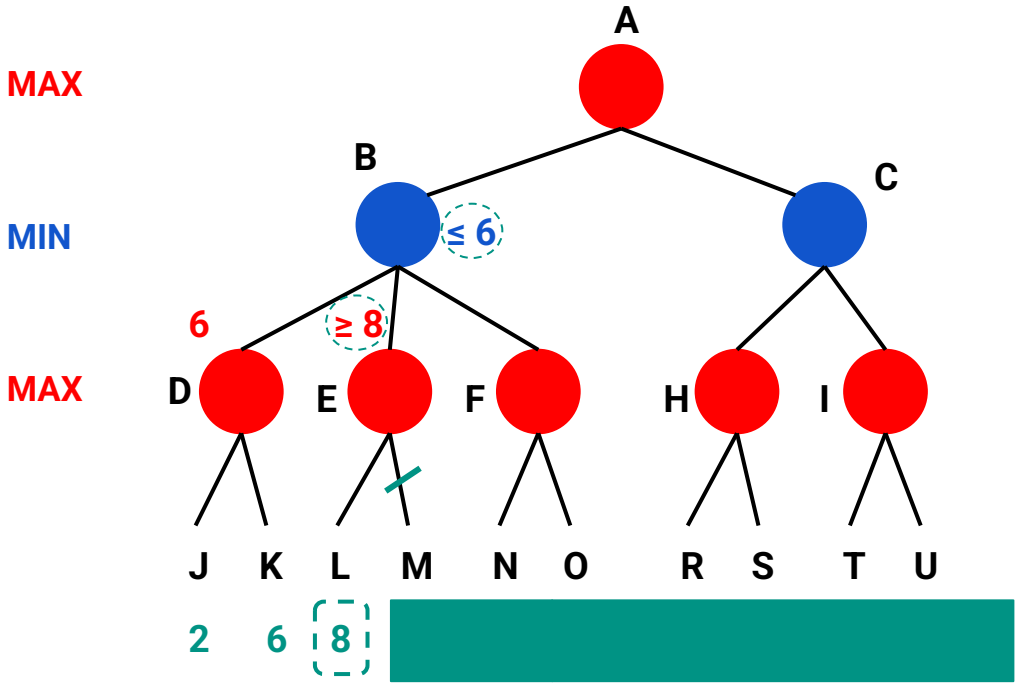


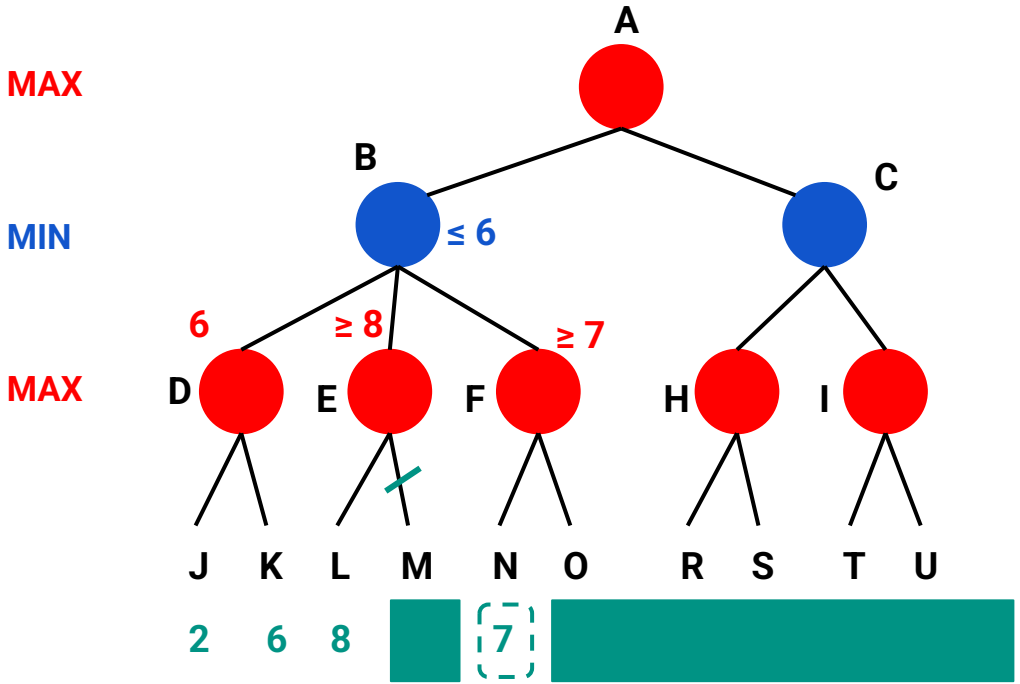

Check for overlaps
to see if we can prune EM
MAX
MIN
MAX
A
B
C
≤ 6
6
≥ 8
D
E
F
H
I
J
K
L
M
N
O
R
S
T
U
2
6
8

Prune EM

Reasoning (in Lecture):
The intervals ≤ 6 and ≥ 8 do not overlap, so **no need to evaluate the children**

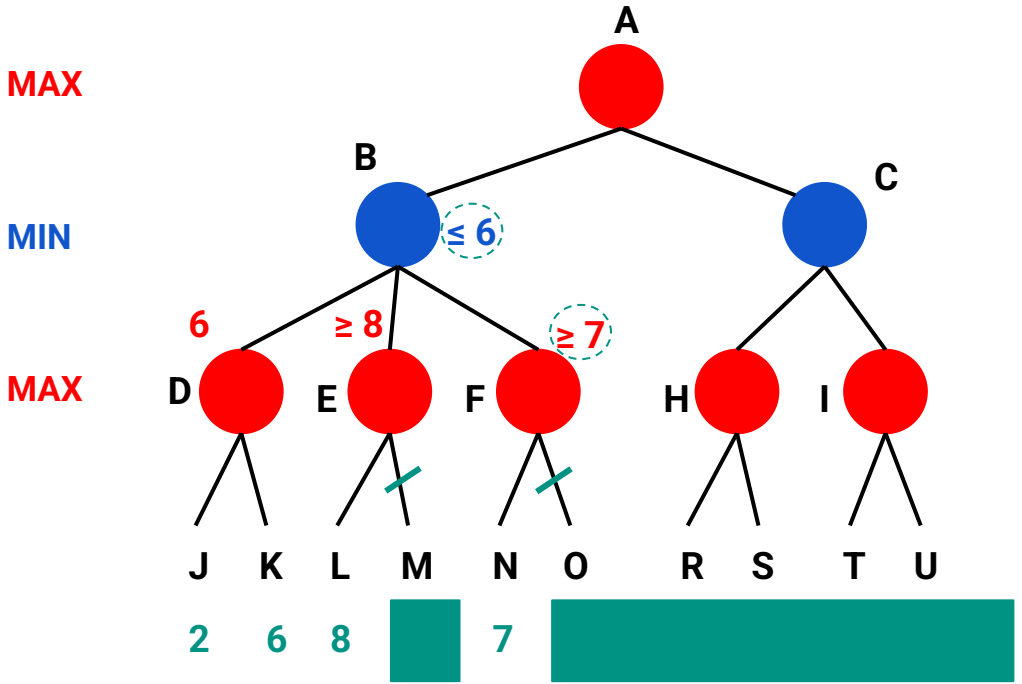
Intuitive reasoning:
E (MAX)'s value is at least 8, **B (MIN) already had a better option - D[6]**, so B doesn't care about E and its children anymore.




MAX
MIN
MAX
A
B
C
≤ 6
6
≥ 8
≥ 7
D
E
F
H
I
J
K
L
M
N
O
R
S
T
U
2
6
8
7

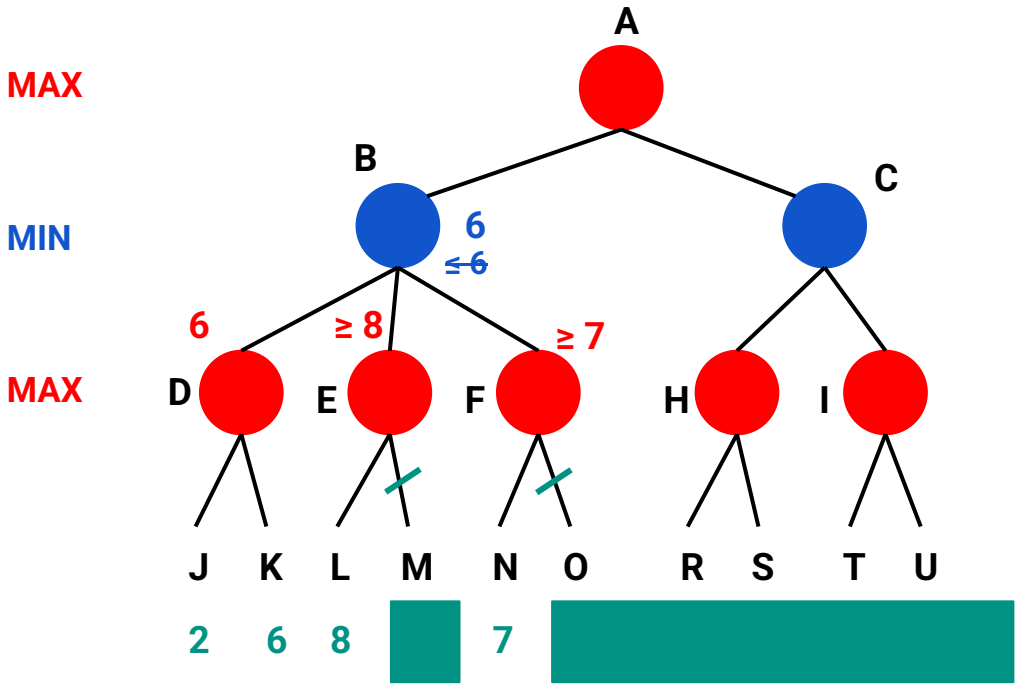
Prune FO

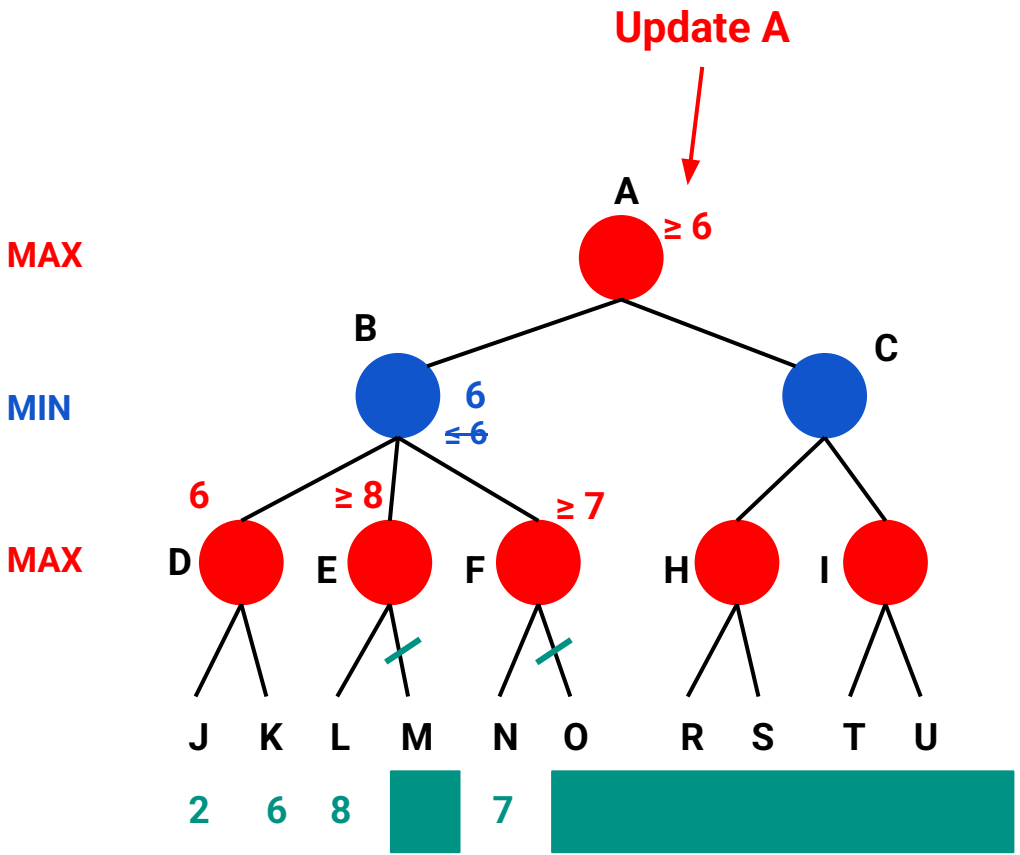
Reasoning:
The intervals ≤ 6 and ≥ 7 do not overlap, so **no need to evaluate the children**



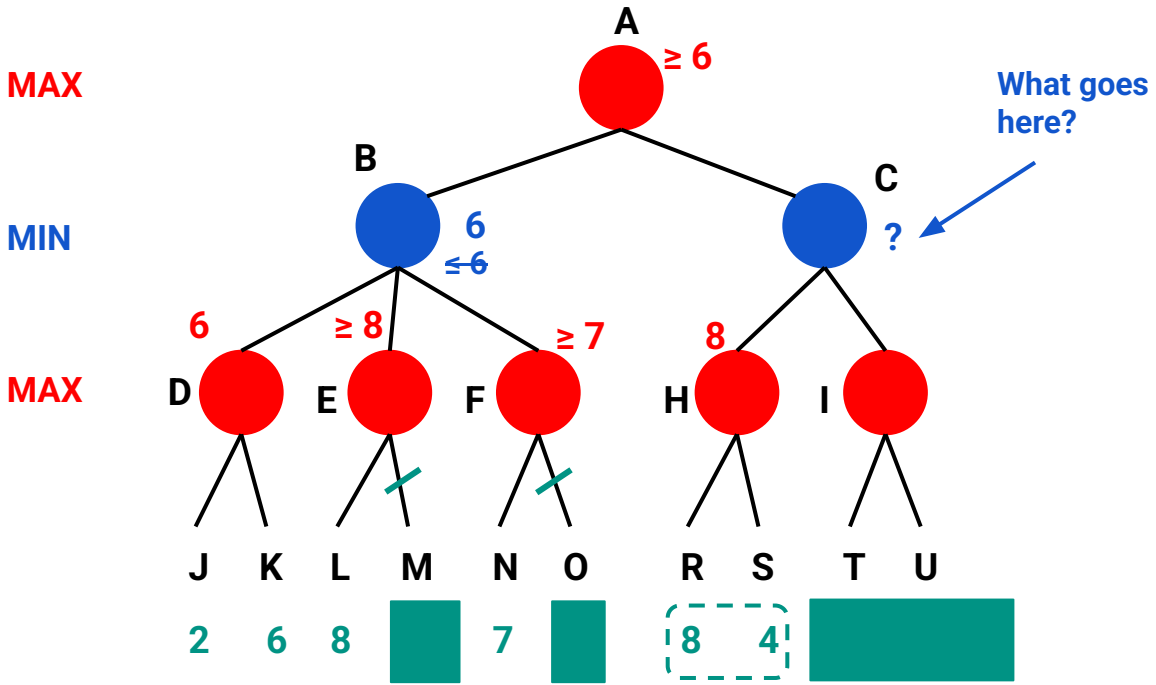
B's value is now sure to be 6

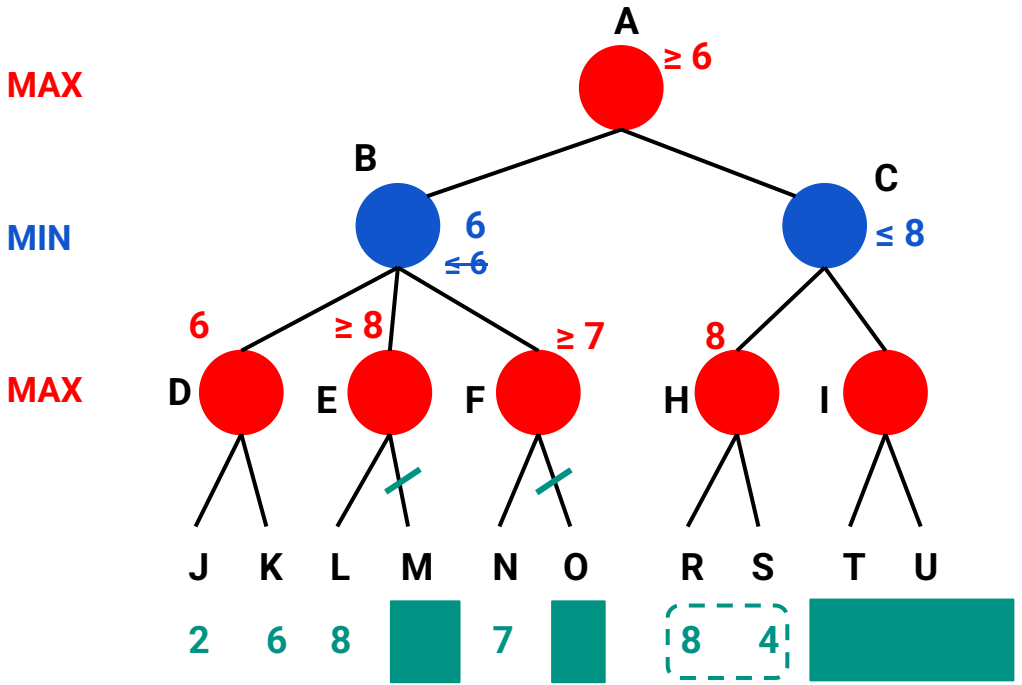
Reminder: Assuming that everyone pick the optimal move

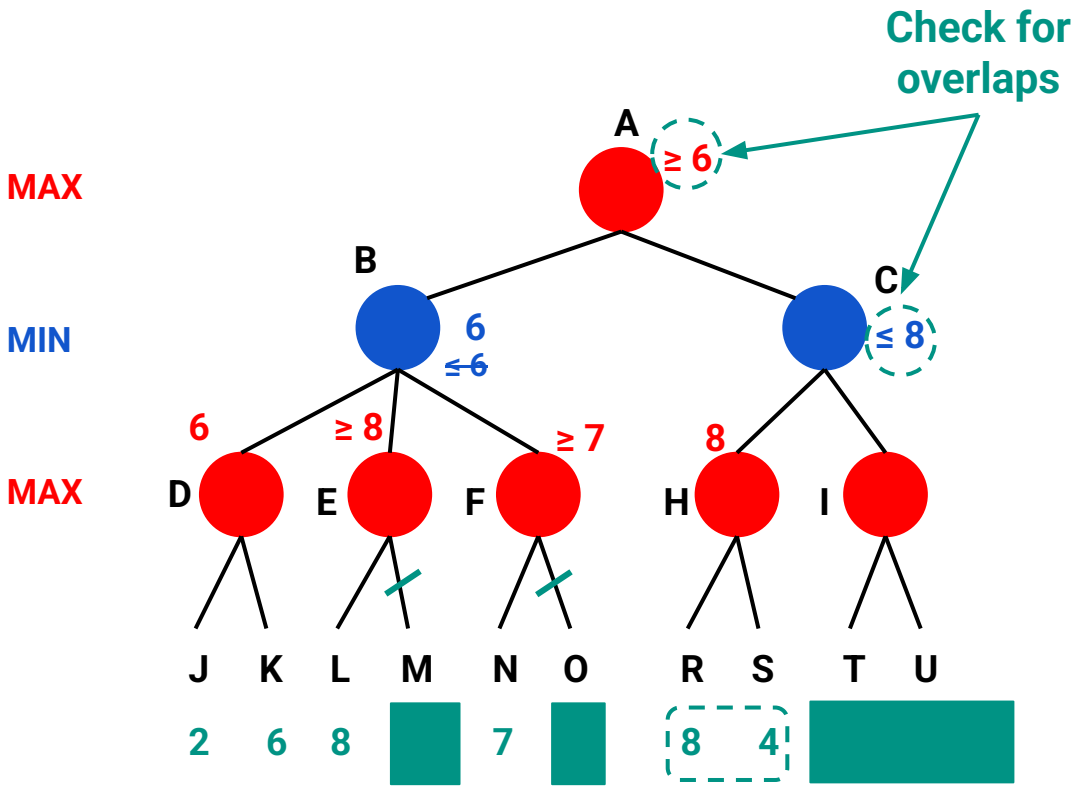


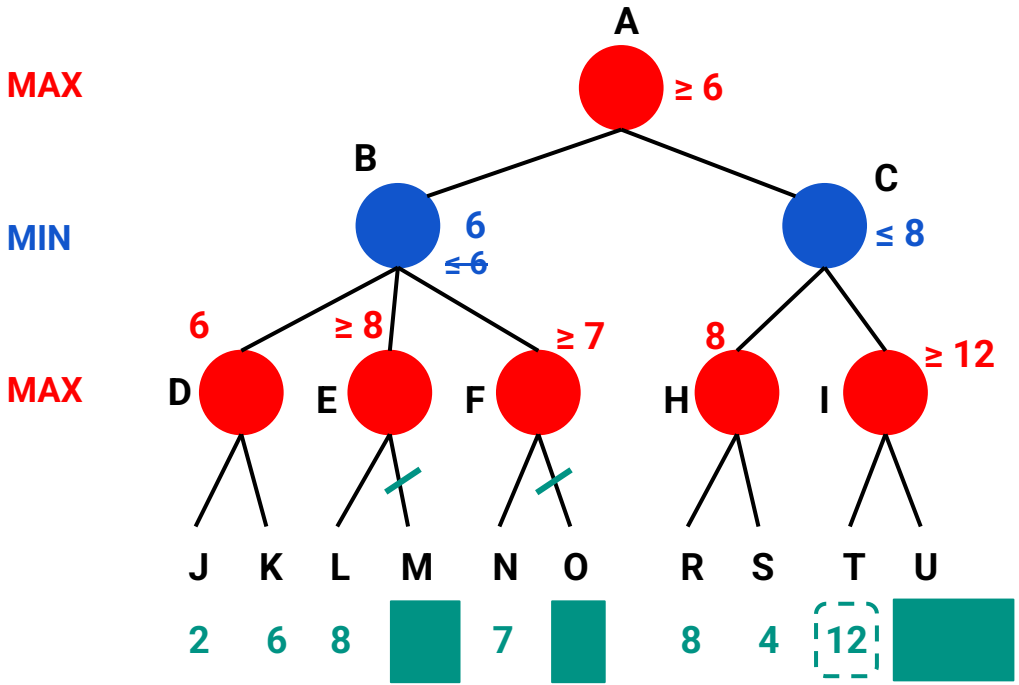

Update A
A
≥ 6
MAX
B
C
MIN
6
≤ 6
6
≥ 8
≥ 7
MAX
D
E
F
H
I
J
K
L
M
N
O
R
S
T
U
2
6
8
7

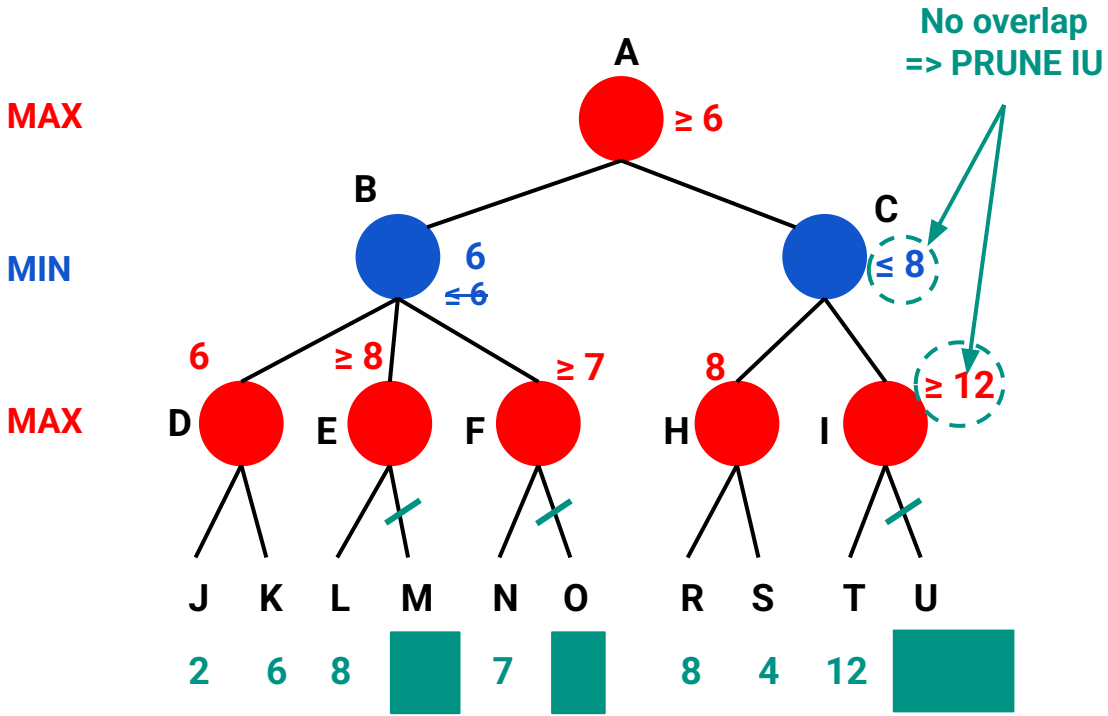
Always evaluate all children of the first branch!

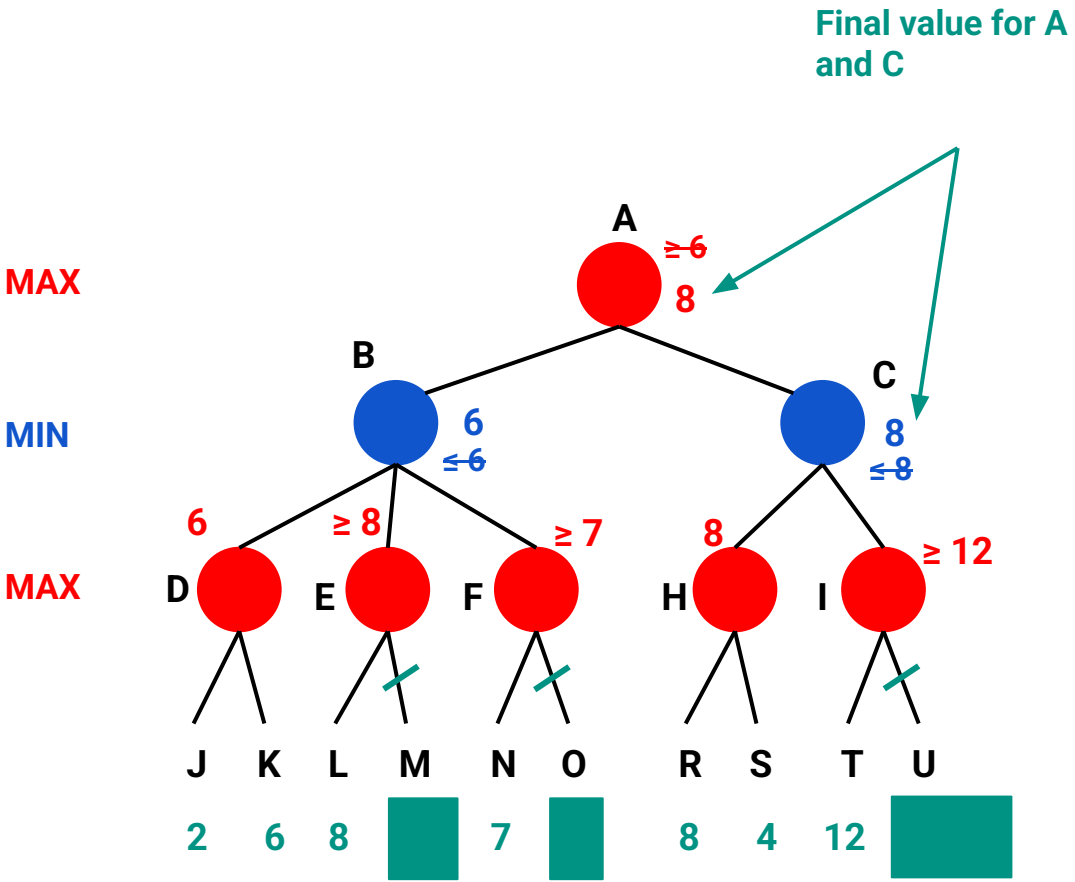



MAX
MIN
MAX
A
≥ 6
B
6
≤ 6
C
≤ 8
6
≥ 8
≥ 7
8
D
E
F
H
I
J
K
L
M
N
O
R
S
T
U
2
6
8
7
8
4

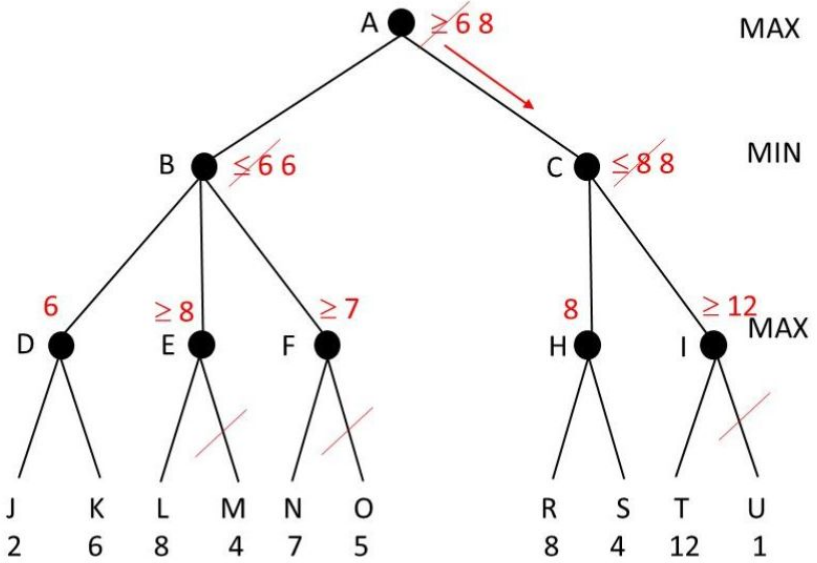

Check for overlaps
MAX
MIN
MAX
A
≥ 6
B
6
≤ 6
C
≤ 8
6
≥ 8
≥ 7
8
D
E
F
H
I
J
K
L
M
N
O
R
S
T
U
2
6
8
7
8
4


MAX
MIN
MAX
A
≥ 6
B
C
6
≤ 6
≤ 8
6
≥ 8
≥ 7
8
≥ 12
D
E
F
H
I
J
K
L
M
N
O
R
S
T
U
2
6
8
7
8
4
12


No overlap
=> PRUNE IU
MAX
MIN
MAX
A
≥ 6
B
6
≤ 6
C
≤ 8
6
≥ 8
≥ 7
8
≥ 12
D
E
F
H
I
J
K
L
M
N
O
R
S
T
U
2
6
8
7
8
4
12


Final value for A and C
MAX
MIN
MAX
A
≥ 6
8
B
6
≤ 6
C
8
≤ 8
6
≥ 8
≥ 7
8
≥ 12
D
E
F
H
I
J
K
L
M
N
O
R
S
T
U
2
6
8
7
8
4
12

Three branches are pruned as shown. Max should choose C – the same move as minimax.



Exercise 1

Adversarial Search

Exercise 1 (**NOT** homework)

Why do search in game playing programs always proceed forward from the current position rather than backward from the goal?

Exercise 1 (**NOT** homework)

Why do search in game playing programs always proceed forward from the current position rather than backward from the goal?

Answer:

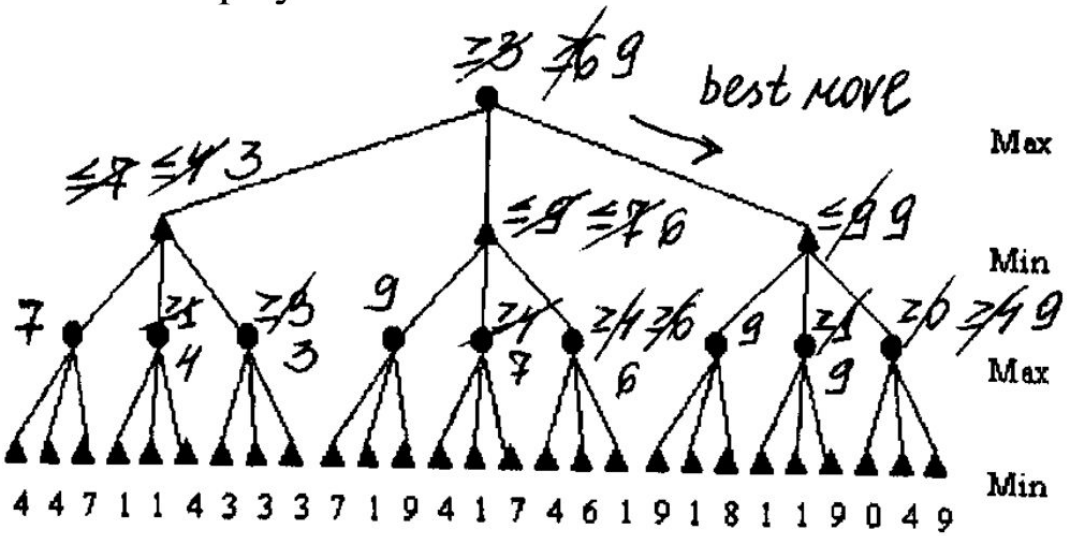
1) **Typically, there are many goal states => we don't know which goal state to start going backward from**
2) **If the goal state is too far from the current state for the search to terminate, the backward search will not give any useful information, whereas a forward incomplete search will give us a good approximation**

Exercise 3

Alpha-Beta Algorithm

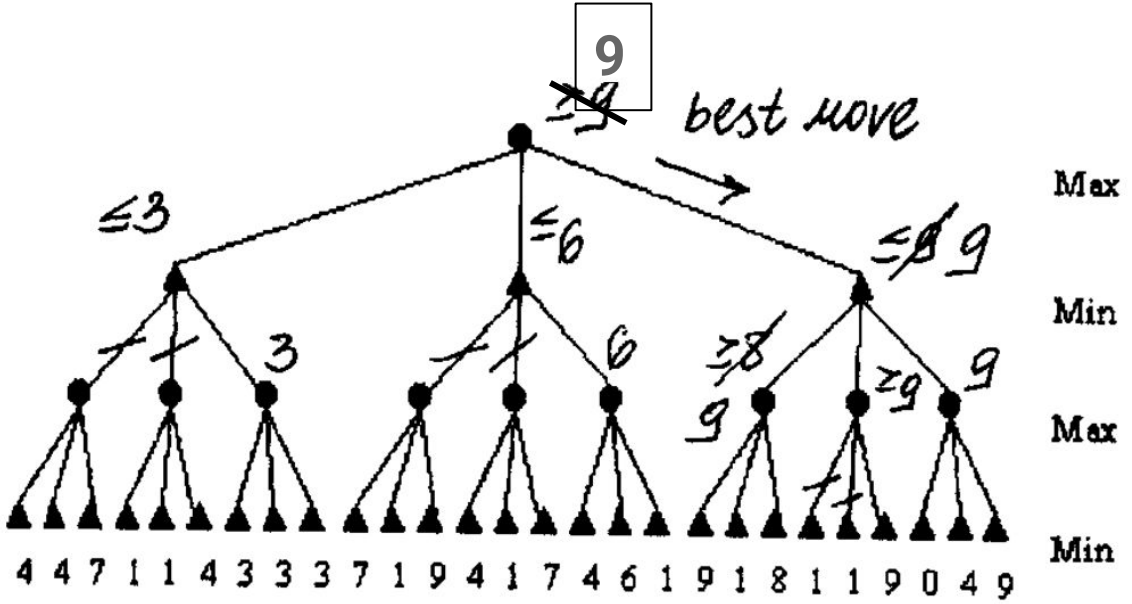
Exercise 3a

No pruning is possible. The first player should take the third move.



Exercise 3b

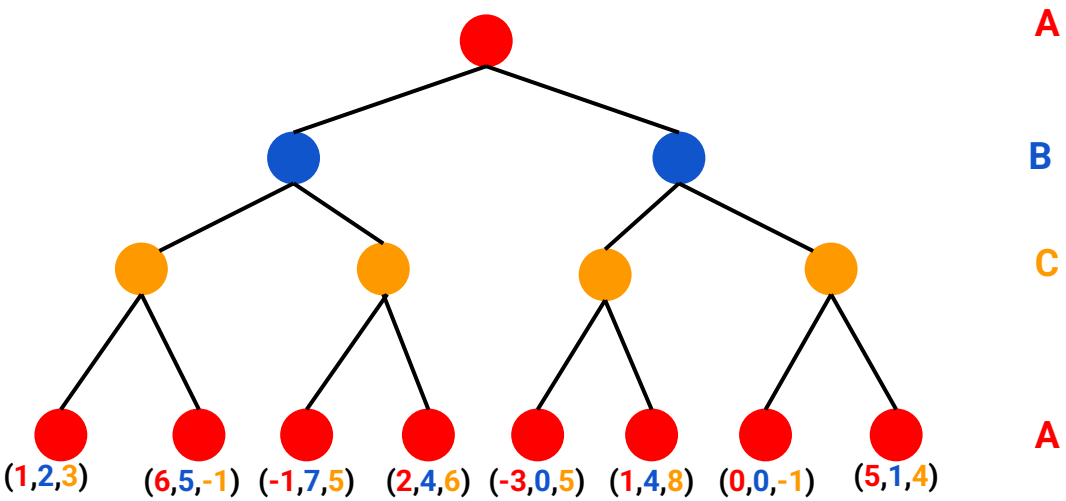
b) Pruning is possible. Again, the third move is the best one.



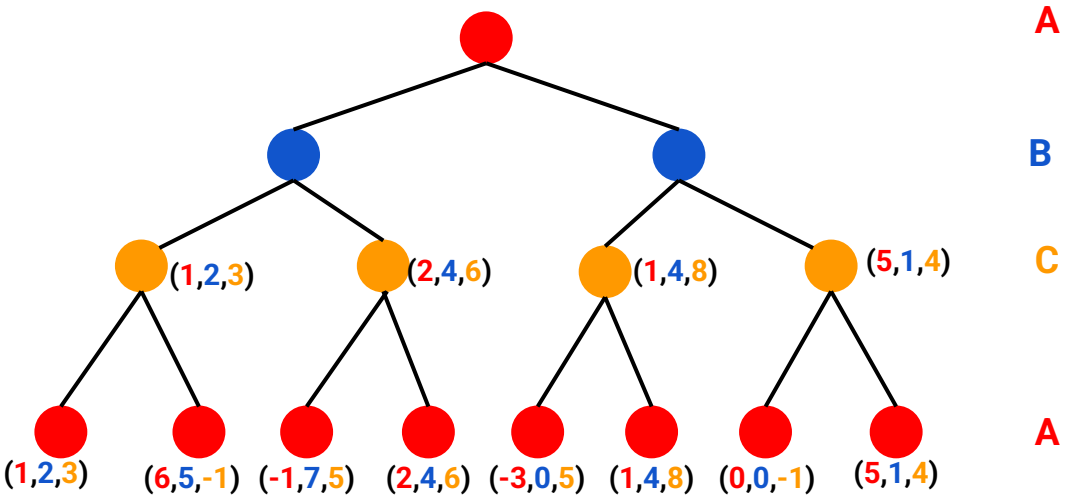
Exercise 4

3-player Minimax

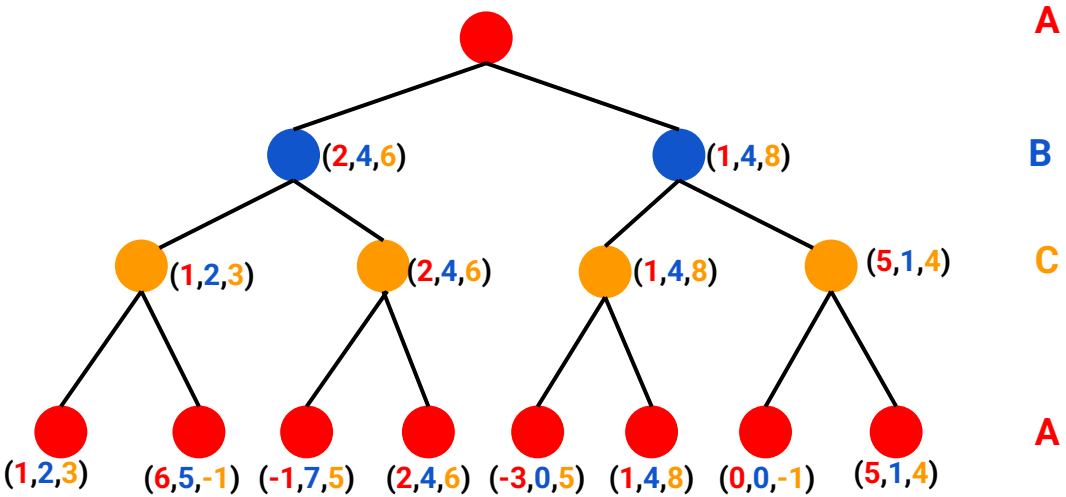
Exercise 4



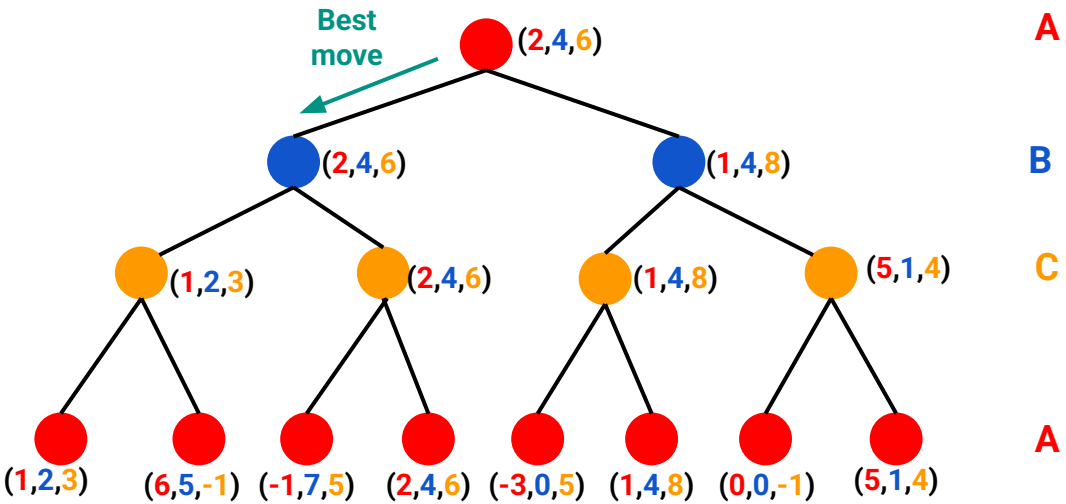
Exercise 4



Exercise 4

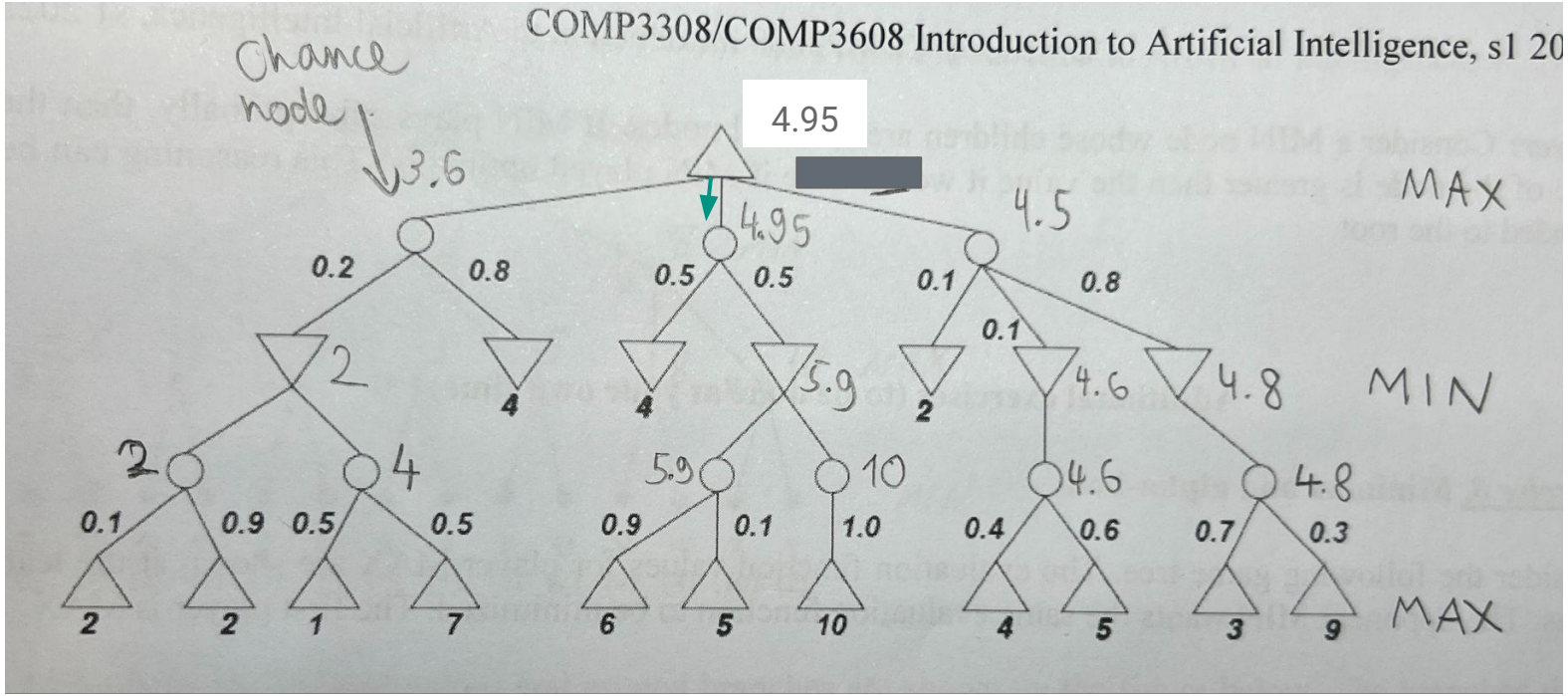


Exercise 4


Best move
(2,4,6)
A
(2,4,6)
(1,4,8)
B
(1,2,3)
(2,4,6)
(1,4,8)
(5,1,4)
C
(1,2,3)
(6,5,-1)
(-1,7,5)
(2,4,6)
(-3,0,5)
(1,4,8)
(0,0,-1)
(5,1,4)
A

Exercise 5

Exercise 5


COMP3308/COMP3608 Introduction to Artificial Intelligence, s1 20
Chance node
3.6
4.95
4.5
MAX
0.2
0.8
0.5
0.5
0.1
0.8
0.1
2
4
4
5.9
2
4.6
4.8
MIN
2
4
5.9
10
4.6
4.8
0.1
0.9
0.5
0.5
0.9
0.1
1.0
0.4
0.6
0.7
0.3
2
2
1
7
6
5
10
4
5
3
9
MAX