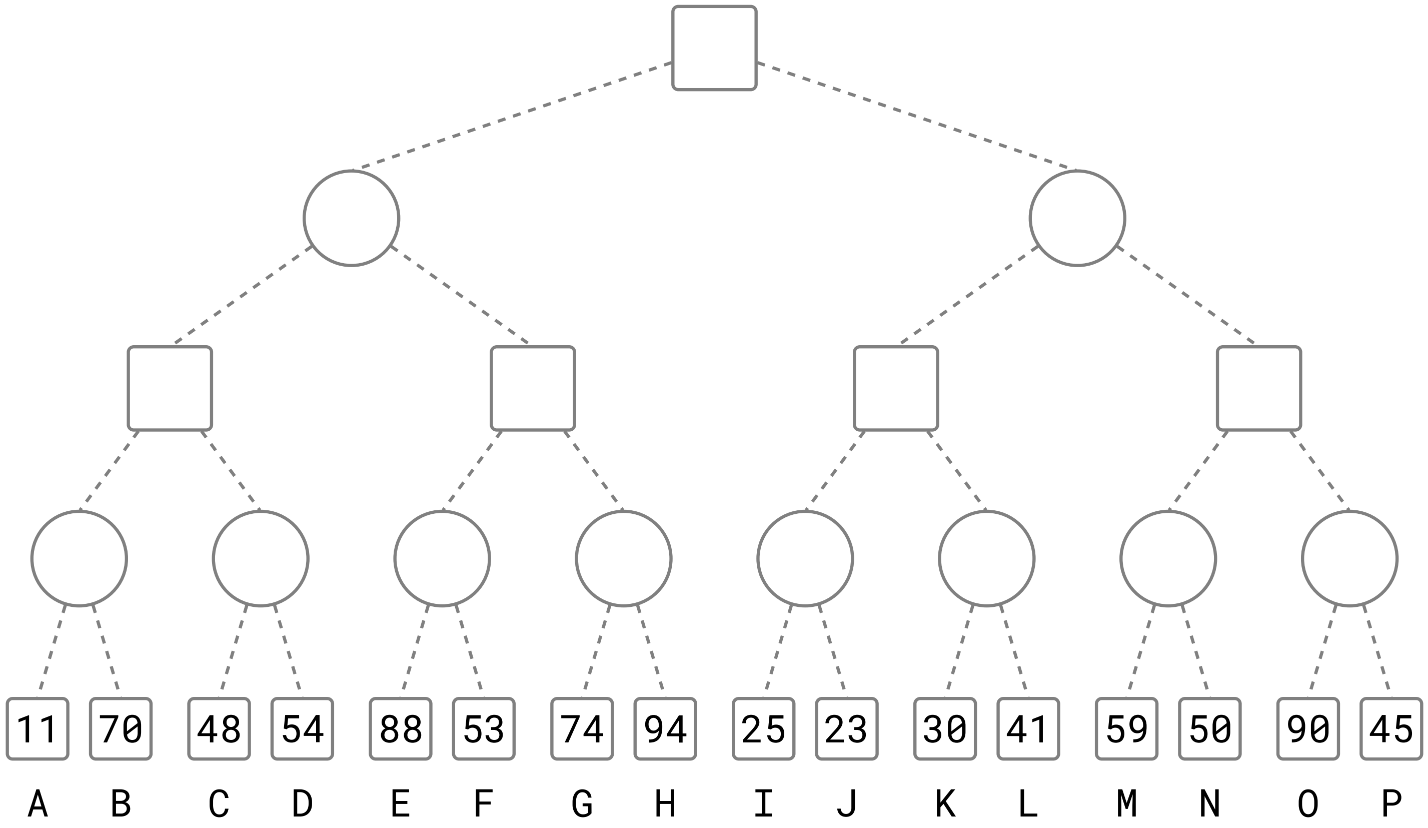
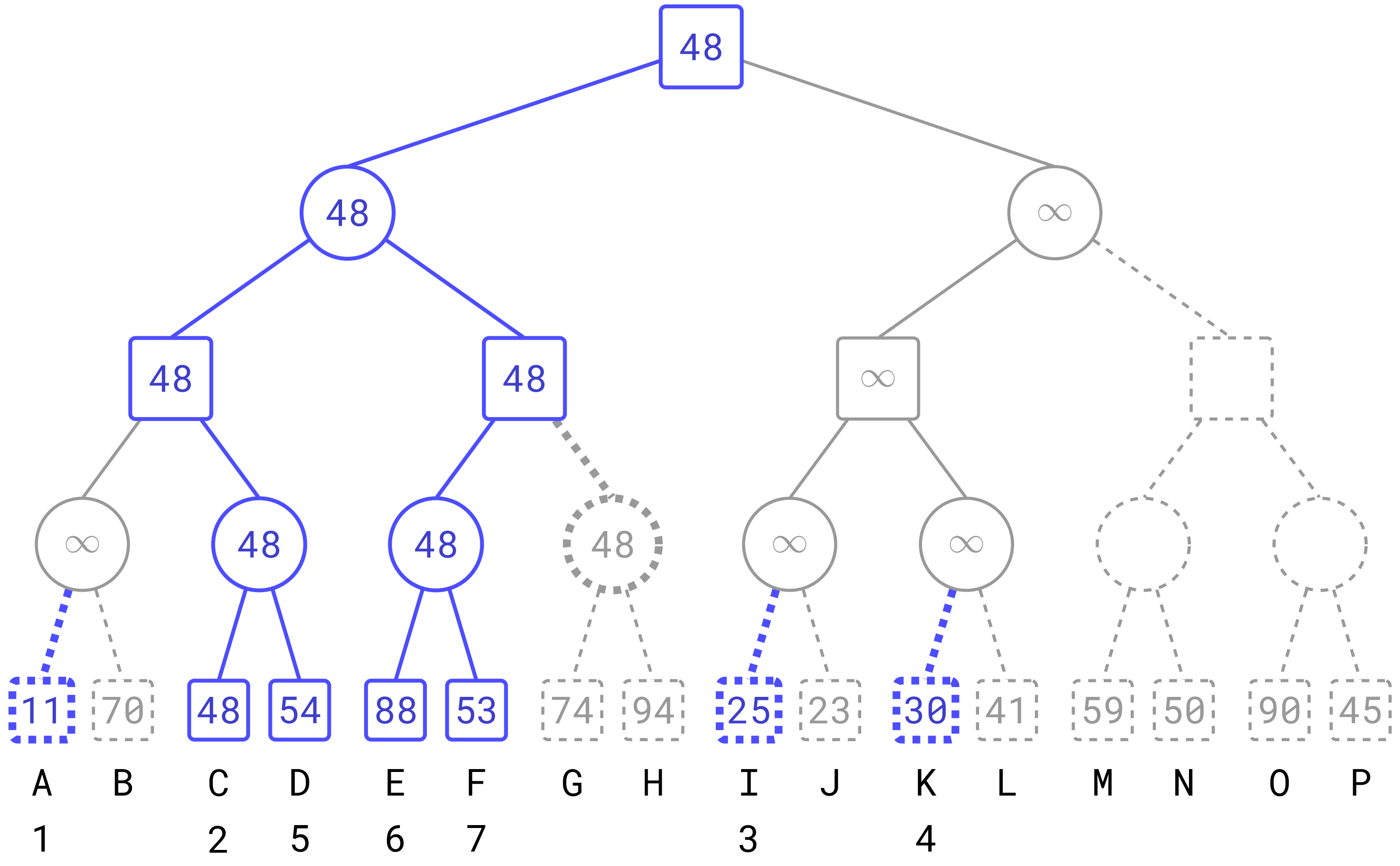
Practice Assignment: Game Tree Example

Prepared by S. Baskaran

Game Tree – PA



SSS* Search Tree



The numbers below the horizon node labels are sequence numbers that show the order in which the horizon nodes were SOLVED.

SSS* Solution

The game tree has 5 levels (a,b,c,d,e,f), the nodes in each level are numbered from left-to-right: the root is a1 followed by b1, b2; then c1,…,c4; d1,…,d8; and e1,…,e16.

To reduce clutter, the forward steps that take us from (sub-tree) root to horizon nodes are not shown here, those steps can be easily emulated by hand.

The backward steps that propagate evals from horizon nodes to the root are shown here.

Tie Breaker: when several nodes have the same h-value then select the deepest leftmost node.

Node Info.: `(NODE,PLAYER,STATUS,H)`
Leaf Node Info.: `(NODE,PLAYER,STATUS,H,LEAF-EVAL)`

```
QUEUE      (e3,MAX,SOLVED,48,48):
           (e11,MAX,SOLVED,30,30):
           (e9,MAX,SOLVED,25,25):
           (e1,MAX,SOLVED,11,11):[]

1.
LEAF       (e3,MAX,SOLVED,48,48)
ADD LIVE   (e4,MAX,LIVE,48,54)

QUEUE      (e4,MAX,LIVE,48,54):
           (e11,MAX,SOLVED,30,30):
           (e9,MAX,SOLVED,25,25):
           (e1,MAX,SOLVED,11,11):[]

2.
LEAF       (e4,MAX,LIVE,48,54)
ADD SOLVED (e4,MAX,SOLVED,48,54)

QUEUE      (e4,MAX,SOLVED,48,54):
           (e11,MAX,SOLVED,30,30):
           (e9,MAX,SOLVED,25,25):
           (e1,MAX,SOLVED,11,11):[]

3.
LEAF       (e4,MAX,SOLVED,48,54)
ADD SOLVED (d2,MIN,SOLVED,48)

QUEUE      (d2,MIN,SOLVED,48):
           (e11,MAX,SOLVED,30,30):
           (e9,MAX,SOLVED,25,25):
           (e1,MAX,SOLVED,11,11):[]

4.
NODE       (d2,MIN,SOLVED,48)
ADD SOLVED (c1,MAX,SOLVED,48)
PRUNE      (e1,MAX,SOLVED,11,11)

QUEUE      (c1,MAX,SOLVED,48):
           (e11,MAX,SOLVED,30,30):
           (e9,MAX,SOLVED,25,25):[]

5.
NODE       (c1,MAX,SOLVED,48)
ADD LIVE   (c2,MAX,LIVE,48)

QUEUE      (c2,MAX,LIVE,48):
           (e11,MAX,SOLVED,30,30):
           (e9,MAX,SOLVED,25,25):[]

6.
NODE       (c2,MAX,LIVE,48)
ADD LIVE   (d3,MIN,LIVE,48)
ADD LIVE   (d4,MIN,LIVE,48)

QUEUE      (d3,MIN,LIVE,48):
           (d4,MIN,LIVE,48):
           (e11,MAX,SOLVED,30,30):
           (e9,MAX,SOLVED,25,25):[]

7.
NODE       (d3,MIN,LIVE,48)
ADD LIVE   (e5,MAX,LIVE,48,88)

QUEUE      (e5,MAX,LIVE,48,88):
           (d4,MIN,LIVE,48):
           (e11,MAX,SOLVED,30,30):
           (e9,MAX,SOLVED,25,25):[]

8.
LEAF       (e5,MAX,LIVE,48,88)
ADD SOLVED (e5,MAX,SOLVED,48,88)

QUEUE      (e5,MAX,SOLVED,48,88):
           (d4,MIN,LIVE,48):
           (e11,MAX,SOLVED,30,30):
           (e9,MAX,SOLVED,25,25):[]

9.
LEAF       (e5,MAX,SOLVED,48,88)
ADD LIVE   (e6,MAX,LIVE,48,53)

QUEUE      (e6,MAX,LIVE,48,53):
           (d4,MIN,LIVE,48):
           (e11,MAX,SOLVED,30,30):
           (e9,MAX,SOLVED,25,25):[]

10.
LEAF       (e6,MAX,LIVE,48,53)
ADD SOLVED (e6,MAX,SOLVED,48,53)

QUEUE      (e6,MAX,SOLVED,48,53):
           (d4,MIN,LIVE,48):
           (e11,MAX,SOLVED,30,30):
           (e9,MAX,SOLVED,25,25):[]

11.
LEAF       (e6,MAX,SOLVED,48,53)
ADD SOLVED (d3,MIN,SOLVED,48)

QUEUE      (d3,MIN,SOLVED,48):
           (d4,MIN,LIVE,48):
           (e11,MAX,SOLVED,30,30):
           (e9,MAX,SOLVED,25,25):[]

12.
NODE       (d3,MIN,SOLVED,48)
ADD SOLVED (c2,MAX,SOLVED,48)
PRUNE      (d4,MIN,LIVE,48)

QUEUE      (c2,MAX,SOLVED,48):
           (e11,MAX,SOLVED,30,30):
           (e9,MAX,SOLVED,25,25):[]

13.
NODE       (c2,MAX,SOLVED,48)
ADD SOLVED (b1,MIN,SOLVED,48)

QUEUE      (b1,MIN,SOLVED,48):
           (e11,MAX,SOLVED,30,30):
           (e9,MAX,SOLVED,25,25):[]

14.
NODE       (b1,MIN,SOLVED,48)
ADD SOLVED (a1,MAX,SOLVED,48)
PRUNE      (e11,MAX,SOLVED,30,30)
PRUNE      (e9,MAX,SOLVED,25,25)

QUEUE      (a1,MAX,SOLVED,48):[]

15.
NODE       (a1,MAX,SOLVED,48)
GOAL       (a1,MAX,SOLVED,48)

QUEUE      []
```