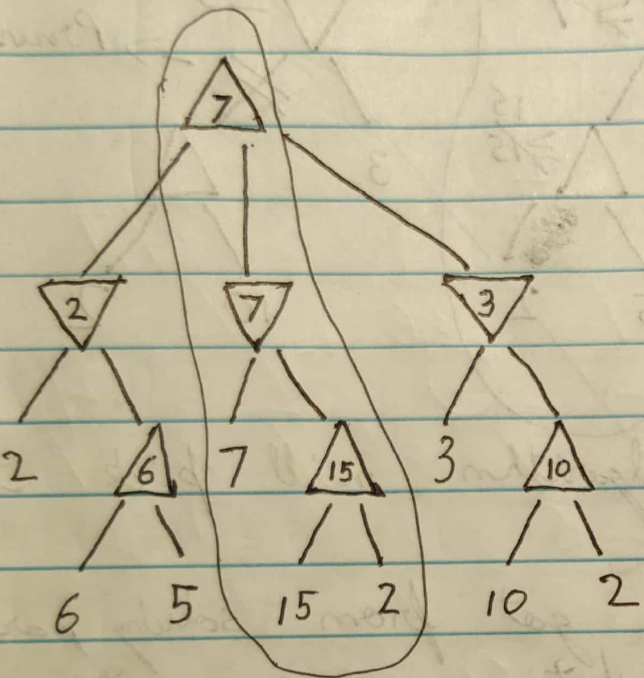
Assignment - 2

ANGAD MANJUNATHA
1001718335

Task - 1

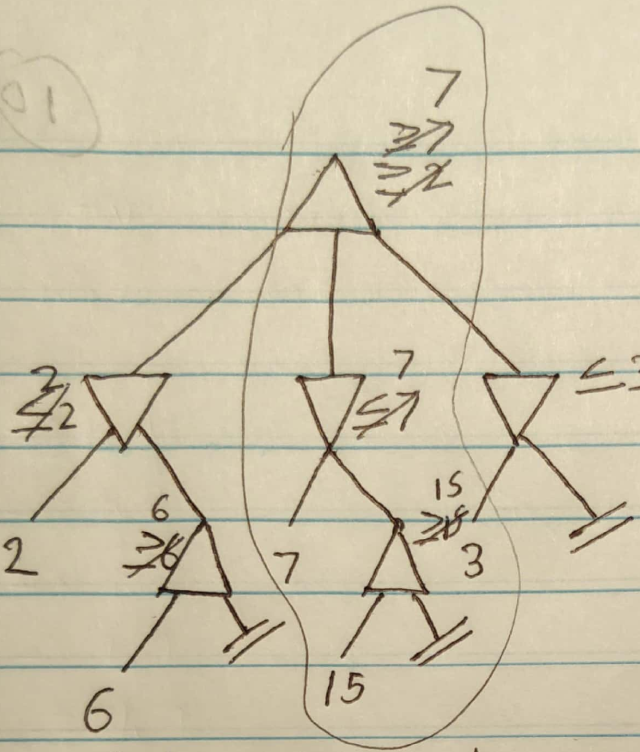
a.



The min-max algorithm will pick the middle action.

Max [6, 5] = 6
Min [2, 6] = 2
Max [15, 2] = 15
Min [7, 15] = 7
Max [10, 2] = 10
Min [3, 10] = 3
Max [2, 7, 3] = 7

b.



The alpha-beta algorithm will pick the middle action.
The answer we got from solving part A is similar to what we got in part B.

3

looking at

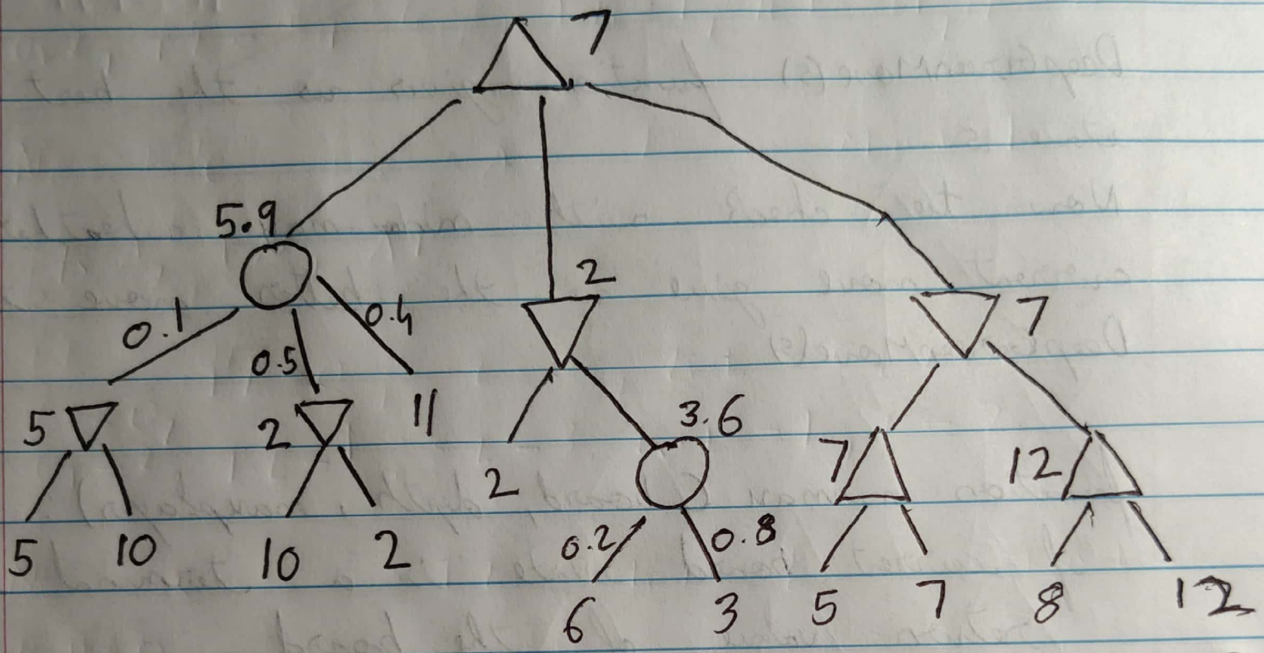
c. From ^ part B's solution, if we know the max utility value, then after traversing in the left to right order. and if we get the max utility value of 15, then we will not have to find the other max value on the right side of the tree. Therefore the other nodes in the right will be pruned

Task - 2

DeepGreenMove(S) function gives us the best move from state s.
Now to check with min-max whether the current move give us the better move than DeepGreenMove(S).

```
function minmax ( board, depth, maxplayer):
  if current board state is a terminal state:
     return value of the board
  if maxplayer:   best value = - infinity
   for each move in board:
       value = minmax ( board, depth +1, false)
       best value = max ( bestvalue, value)
   return bestvalue
else: best value = + infinity
~~for each~~
     for each move in board:
         value = minmax ( board, depth +1, True)
         bestvalue = min ( best value, value)
   return bestvalue.
```

Task - 3.



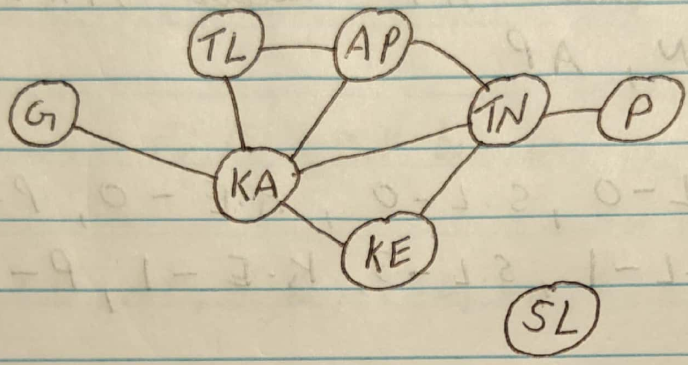
$5(0.1) + 2(0.5) + 11(0.4) =$
$0.5 + 1 + 4.4 = 5.9$

$6(0.2) + 3(0.8) =$
$1.2 + 2.4 = 3.6$

The algorithm will perform the 3rd action.

Task-4

a.



b. In the begining the degree heuristic of each state are as follows:-

G - 1 , TL - 2 , TN - 4 , SL - 0
KA - 5 , AP - 3 , KE - 2 , P - 1
and MRV of each is 3.
The first variable selected is KA because it has the higest D.H value

Selected KA

D.H G - 0 , TL - 1 , TN - 3 , SL - 0 , AP - 2 , KE - 1 , P - 1
Next we go for TN because its DH is 3
and all variables connected to KA now have MRV as 2.

Selected KA, TN

(7)

D.H G-0, TL-1, SL-0, A.P-1, K.E-0, P-0
MRV G-2, TL-2, SL-3, A.P-1, K.E-1, P-2

Now we select A.P because M.R.V-1 and D.H-1

Selected K.A, TN, AP

D.H. G-0, TL-0, S.L-0, K.E-0, P-0
M.R.V G-2, T.L-1, S.L-3, K.E-1, P-2

Now we select T.L or K.E because M.R.V-1 and D.H-0

Selected K.A, TN, AP, K.E

D.H G-0, TL-0, S.L-0, P-0
M.R.V G-2, T.L-1, S.L-3, P-2

Now we select T.L because M.R.V-1 and D.H=0

Selected KA, TN, A.P, K.E, TL,

D.H G-0, SL-0, P-0
MR.V G-2, S.L-3, P-2

Now we select either G or P because M.R.V=2 and D.H-0

Selected KA, TN, A.P, K.E, TL, G

D.H P-0, S.L-0

M.R.V P-2, S.L-3

Now we pick P because M.R.V-2 and D.H-0

Selected KA, TN, AP, K.E, T.L, G., P

Remaining is SL.

Therefore selected variable are
KA, TN, AP, K.E, T.L, G, P, S.L

c. KA TN AP K.E TL G P S.L

R RG.B RG B RG B RGB RG.B RGB RGB

~~TN → KA~~

~~AP → KA~~

~~KE → KA~~

~~TL → KA~~

~~G → KA~~

~~AP → TN~~

~~P → TN~~

~~KE → TN~~

~~TL → AP~~

~~TN → AP~~

~~TN → K.E~~

~~P → K.E~~

~~AP → TL~~

d. Yes we can. use structure of problem to make it more efficient. As SL in the map is seperated, we can treat SL as a seperate sub graph. Thus reducing time complexity.

e. KA → R

TL → B

AP → G

TN → B

K.E → G

P → R

G → B

S.L → R.