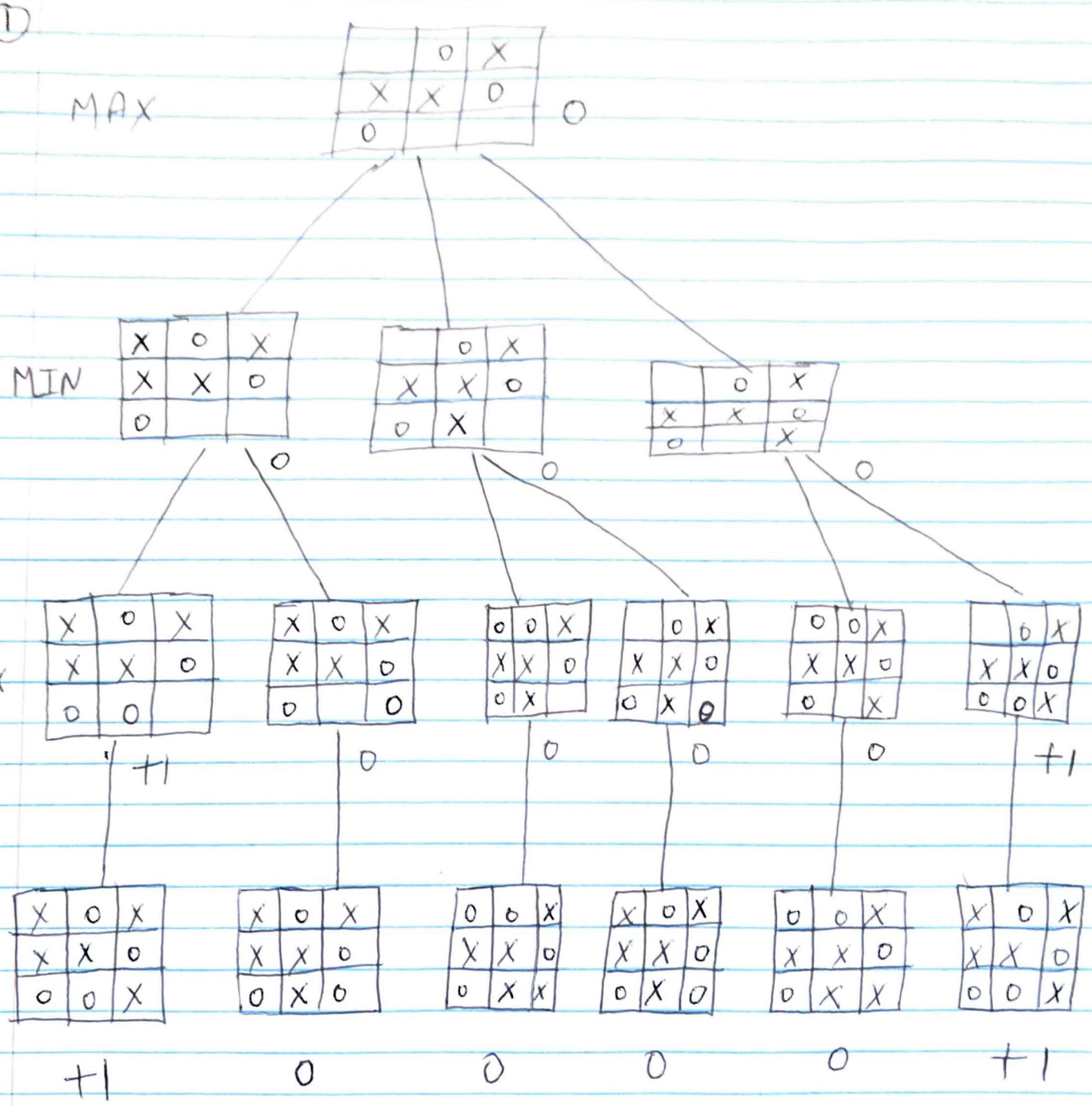
Name: Mahesh Koppala

ID: 1001764522

Assignment-2

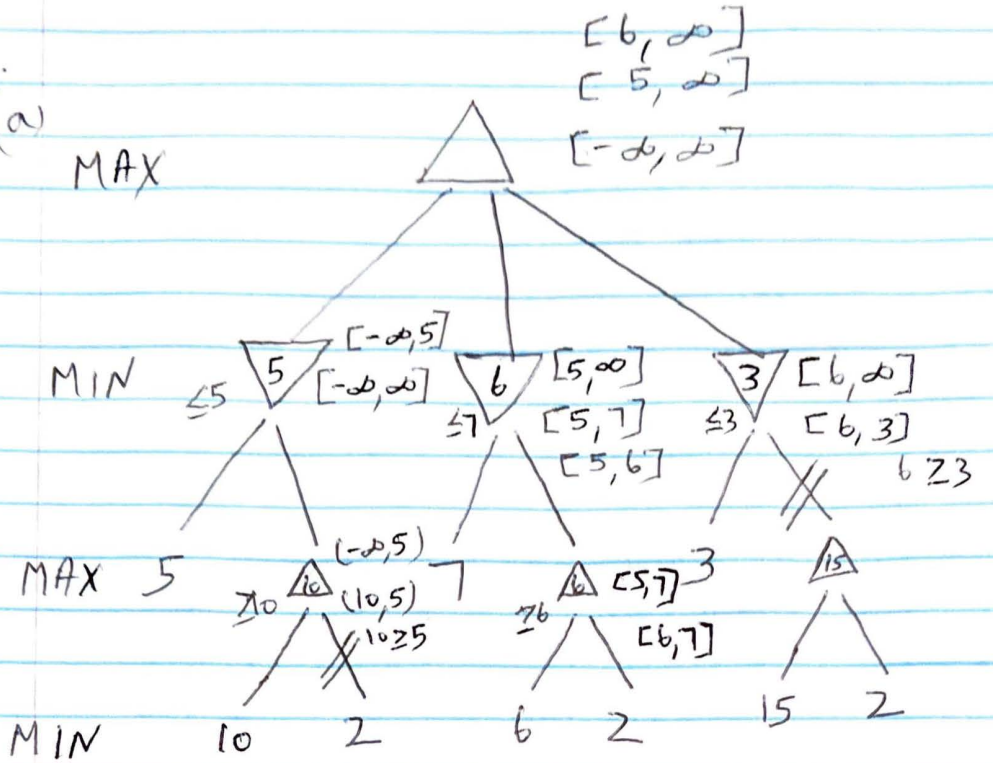
(1)



Max value for root node is zero and the min value for the subsequent level nodes is also zero. Therefore these may not be an optimal path for Max player. But still Move1 (or Move 3 can End up Winning

2.

(a)



α ≥ β → pruning occurs

α → Max

β → Min

Minmax algorithm will pick the second action to execute.

Branches marked with (//) are pruned.

(b) Given Max utility value = 15
Min utility value = 2

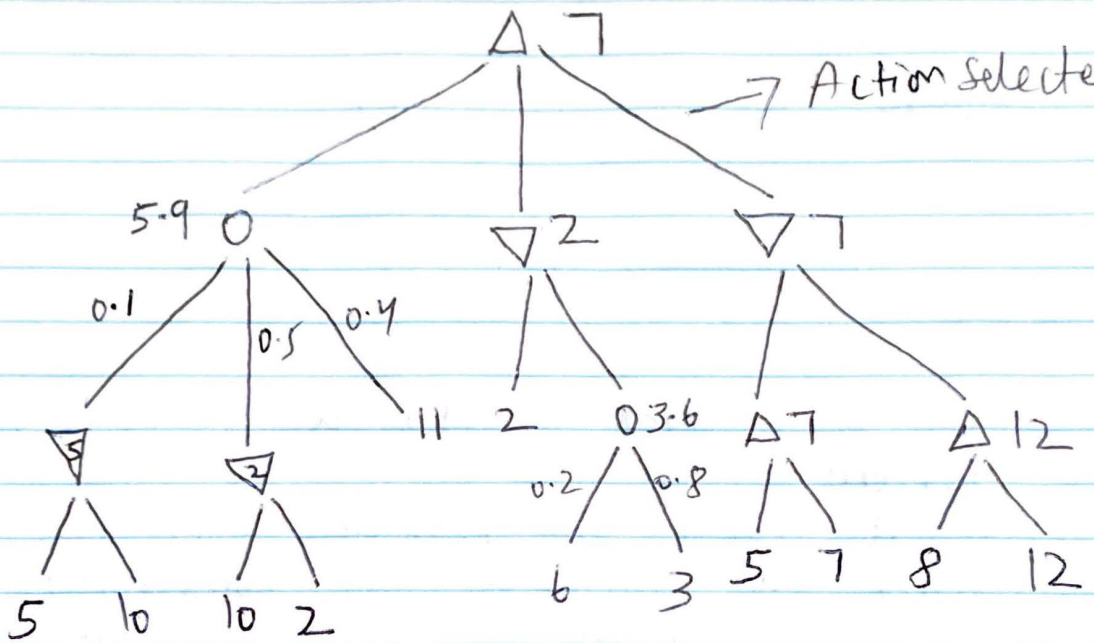
Since, there are no terminals which have values greater than 15 & less than 2. Therefore no pruning occurs. the solution is the same as the above one.

(c) Against optimal opponent

(i) Highest pay off would be 6

Against a random strategy opponent, Highest pay-off would be 7 and lowest pay off would be 6.

(3)



The value at root is expected payoff against an optimal opponent.

if the MIN plays optimal strategy

maximum actual pay off = 7

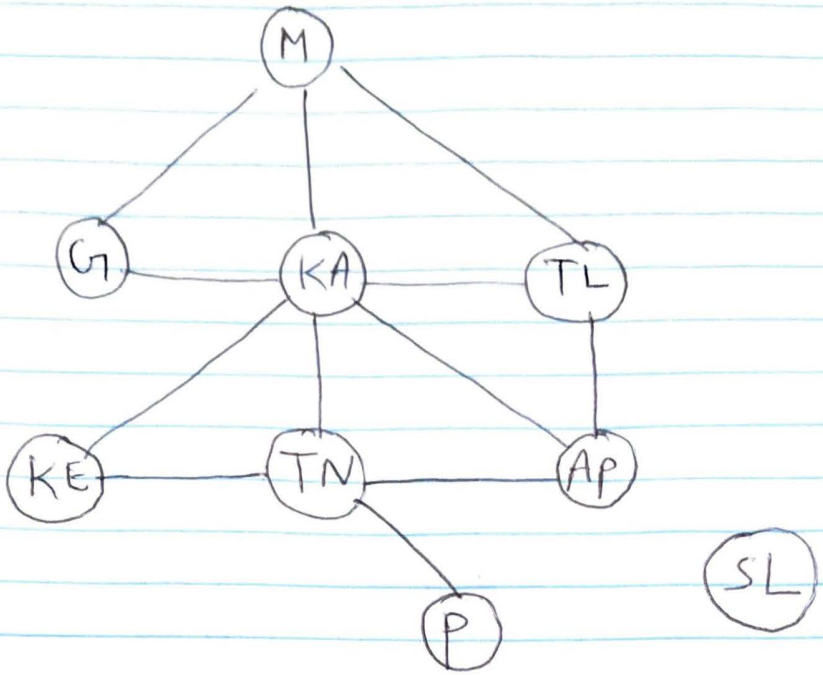
Minimum actual pay off = 5

if the MIN plays random strategy

maximum pay off = 12

minimum pay off = 5

(4) (a)



SL can be solved as it's an independent sub problem.

Yes, it makes the problem easier as we can identify the variables and constraints between them and will turn it into a tree structured CSP which can be solved faster.

(b) Step 1 :- (Level 1)

> We choose Variable with minimum MRV & highest degree

Unassigned Variables : { M, G, KA, TL, KE, TN, Ap, P, SL }

		MRV	Degree
	M	3	3
	G	3	2
	KA	3	5
	TL	3	3
choose	KE	3	2
KA	TN	3	4
	Ap	3	3
	P	3	1
	SL	3	0

Level 2

unassigned variables: { M, G, TL, KE, TN, Ap, P, SL }

		MRV	Degree
	M —	2	3
	G —	2	2
	TL —	2	3
choose	KE —	2	2
TN	TN —	2	4
	Ap —	2	3
	P —	3	1
	SL —	3	0

Level 3

unassigned variables: { M, G, TL, KE, Ap, P, SL }

		MRV	Degree
	M —	2	3
	G —	2	2
	TL —	2	3
choose	KE —	1	2
Ap	Ap —	1	3
	P —	2	1
	SL —	3	0

Level 4

unassigned variables: { M, G, TL, KE, P, SL }

		MRV	Degree
	M —	2	3
	G —	2	2
choose	TL —	1	3
TL	KE —	1	2
	P —	2	1
	SL —	3	0

Level 5

unassigned variables: { M, G, KE, P, SL }

		MRV	Degree
	M –	1	3
	G –	2	2
choose	KE –	1	2
M	P –	2	1
	SL –	3	0

Level 6

unassigned variables : { G, KE, P, SL }

		MRV	Degree
	G –	1	2
choose	KE –	1	2
KE	P –	2	1
	SL –	3	0

Level 7

unassigned variables : { G, P, SL }

		MRV	Degree
choose	G –	1	2
G	P –	2	1
	SL –	3	0

Level 8 unassigned variables : { P, SL }

		MRV	Degree
choose	P –	2	1
P	SL –	3	0

Level 9 unassigned variable : { SL }

		MRV	Degree
choose SL	SL –	3	0

(c)

M	G	KA	TL	Ap	KE	TN	P	SL
~~RGB~~	~~RGB~~	R	~~RGB~~	~~RGB~~	~~RGB~~	~~RGB~~	RGB	RGB

check KA – G
KA – M
KA – TN
KA – KE
KA – Ap
KA – TL

M	G	KA	TL	Ap	KE	TN	P	SL
GB	GB	R	GB	GB	GB	GB	RGB	RGB

check TL – M
TL – Ap
TL – KA
Ap – KA
Ap – TL
Ap – TN
M – G
M – KA
M – TL
TN – KA
TN – KE
TN – Ap
TN – P
KE – TN
KE – KA

No changes in these arcs.
so after propagation

M	G	KA	TL	Ap	KE	TN	P	SL
GB	GB	R	GB	GB	GB	GB	RGB	RGB

(4) (a)

M	KA	—	R
	TL	—	G
	G	—	G
	Ap	—	B
	M	—	B
	KE	—	B
	TN	—	G
	P	—	R
	SL	—	G

→ one of the possible solution

(5) If we know exactly what move MIN player will make, we can use that instead of evaluating all possible moves.

```
function MIN-VALUE(state)
  if Terminal-Test(state) then return UTILITY(state)
  else
      return MAX-VALUE(DEEPGREENMOVE(state))
```

The solution this returns will be the exact strategy that MINMAX would return [if DeepGreen was an optimal player]. If DeepGreen is sub-optimal then this method will take advantage of it.

Since every MIN node will only have one child - the time complexity will only be $O(b^{m/2})$.