Xie Zhou
912143385
ECS170
01/29/2017

Problem Set 3

Problem 1

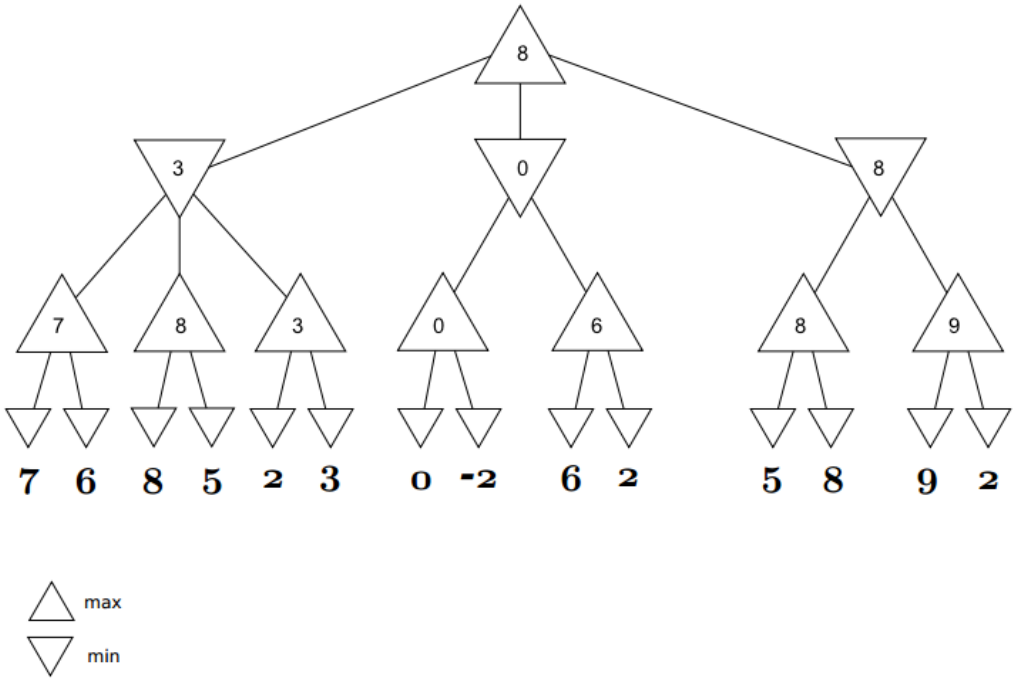
a) No. Minimax is done from bottom to top. Because MAX's moves are always rational and the whole tree(every node) is explored, that guarantees: whatever move MIN makes, MAX will always make the best (most beneficial to himself) move.
b) If MIN makes a suboptimal move, MAX will get an even higher score. This is because nash equilibrium (w/ respect to MAX) guarantees that MAX will get at least the amount of payoff Y if both player act perfectly. If not, MAX will get more than Y.
c) Yes we do. For example, on MIN's turn, suppose there are two states with values a and b where a < b. If being rational, MIN will choose a over b. Accordingly, the algorithm should be pruning the b branch. However, if MIN's move is suboptimal (he chooses b over a), the pruning algorithm needs to be rerun to add the b branch (and further expand the node).
d) If our prior knowledge isn't correct and MIN makes an unpredicted move, we will need to expand some other node. Following that, we would need to expand more nodes (more search space). And because of the corrupted information, we are unable to prune the tree correctly. There would be more search space and also the time constraint might fail.

Problem 2

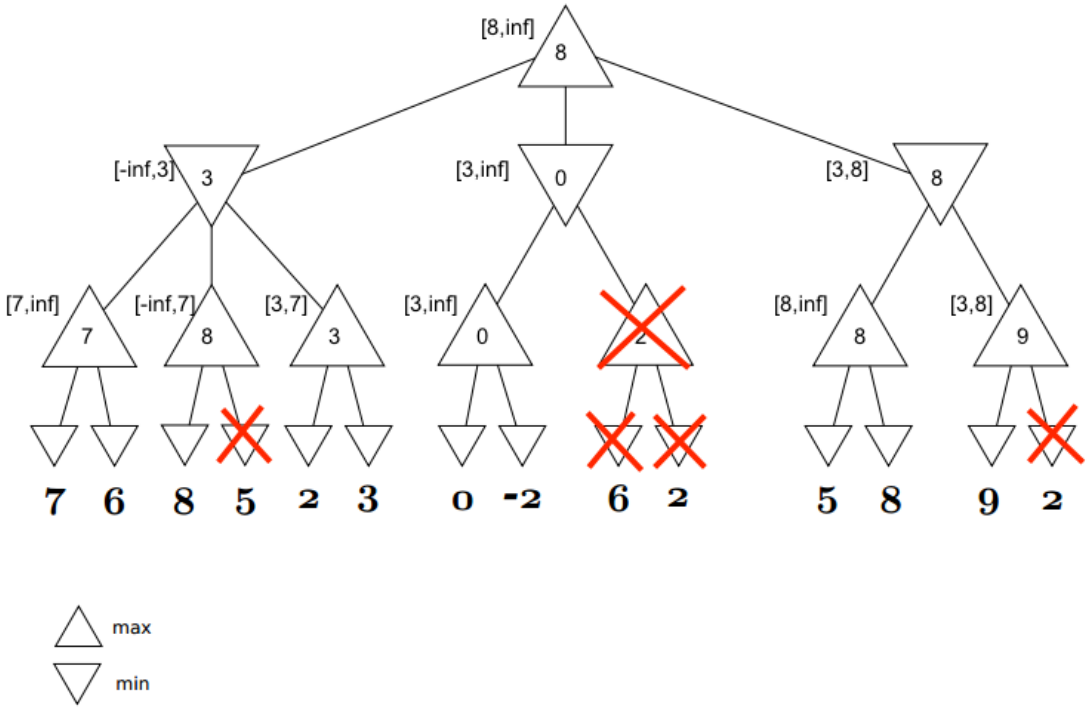
Suppose n is MAX, then alpha records its current best choice and determines if the MIN branches down the tree needs to be pruned or not. Beta is from its parent node, it determines if n itself still needs to be expanded or not. The situation is reversed when n is MIN.
In short,
if n is MAX, alpha is the lower bound on its children and beta is the upper bound on itself.
If n is MIN, alpha is the lower bound on itself and beta is the upper bound on its children.

Problem 3

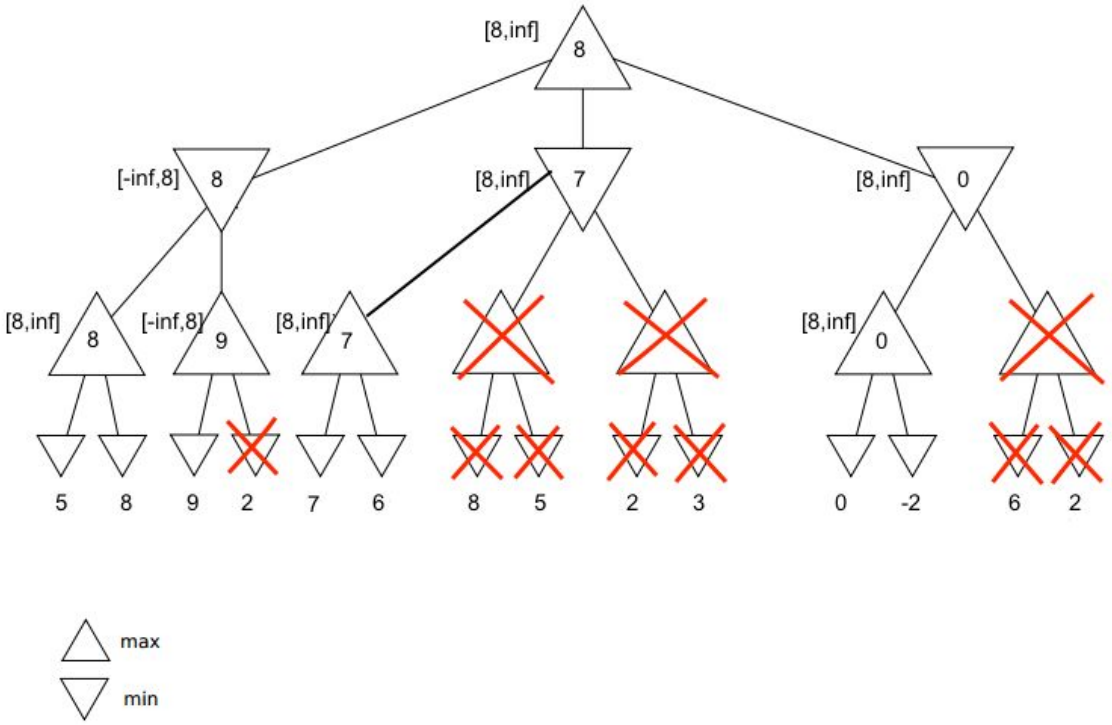
a)


8
3
0
8
7
8
3
0
6
8
9
7 6 8 5 2 3 0 -2 6 2 5 8 9 2
max
min

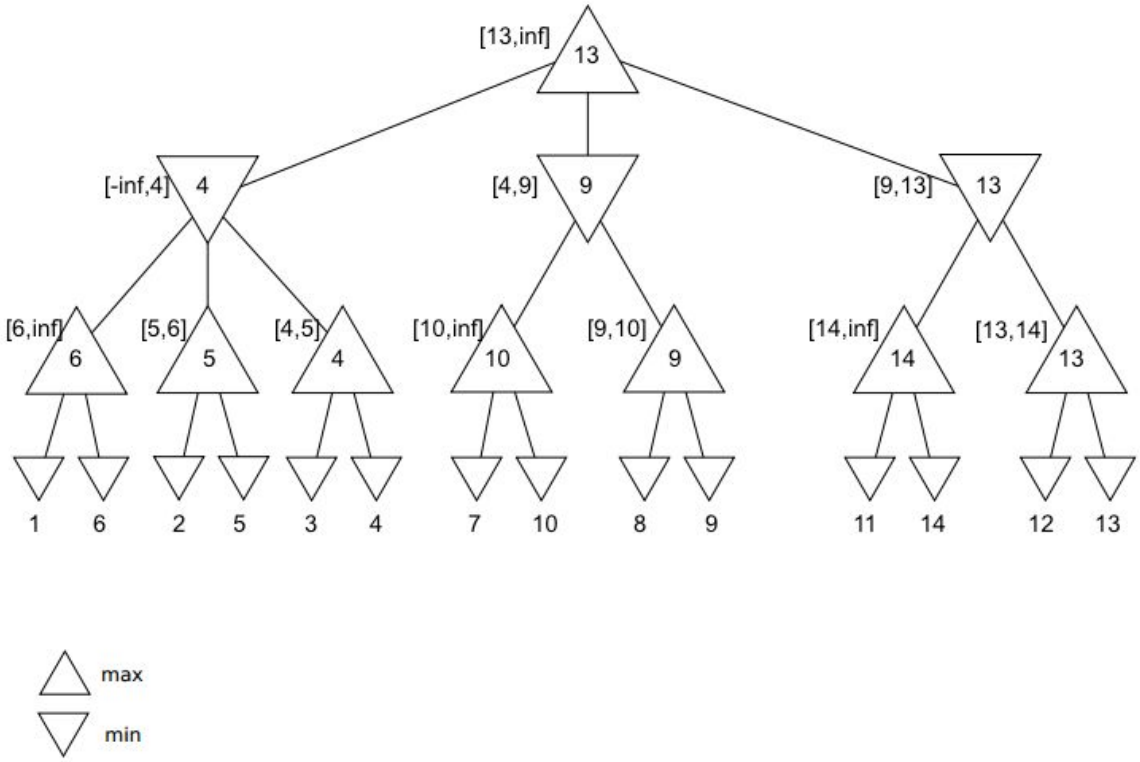
b)


[8,inf]
8
[-inf,3]
3
[3,inf]
0
[3,8]
8
[7,inf]
7
[-inf,7]
8
[3,7]
3
[3,inf]
0
2
[8,inf]
8
[3,8]
9
7 6 8 5 2 3 0 -2 6 2 5 8 9 2
max
min

c)


[8,inf]
8
[-inf,8]
8
[8,inf]
7
[8,inf]
0
[8,inf]
8
[-inf,8]
9
[8,inf]
7
[8,inf]
0
5
8
9
2
7
6
8
5
2
3
0
-2
6
2
max
min

Problem 4


[13,inf]
13
[-inf,4]
4
[4,9]
9
[9,13]
13
[6,inf]
6
[5,6]
5
[4,5]
4
[10,inf]
10
[9,10]
9
[14,inf]
14
[13,14]
13
1
6
2
5
3
4
7
10
8
9
11
14
12
13
max
min