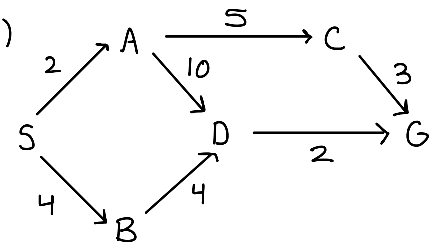
Problem 3:

a)



S: h(S) = 7

A: h(A) = 6

B: h(B) = 5

C: h(C) = 1

D: h(D) = 4

G: h(G) = 0

b) A^* algorithm

Start: S Target = G

1. Frontier: S (7)

   Explored:

2. Explored: S (7)

   Frontier: A (2+6 = 8) B (4+5 = 9)

   cost ↙ h(x)

3. Explored: S (7), A (8)

   Frontier: B (4+5 = 9)
   C (2+5+1 = 8)
   D (2+10+4 = 16)

4. Explored: S (7), A (8), C (8)

   Frontier: B (4+5 = 9)
   D (2+10+4 = 16)
   G (2+5+3+0 = 10)

5. Explored: S (7), A (8), C (8), B (9)

   Frontier: G (7+3+0 = 10)
   D (4+4+4 = 12)

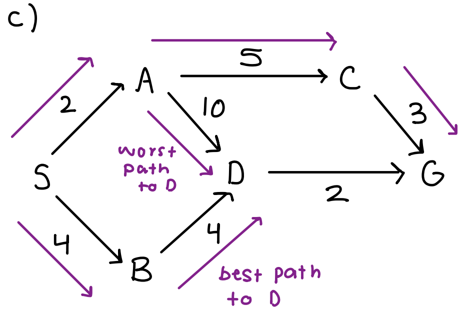
6. Explored: S (7), A (8), C (8), B (9), G (10)

   Frontier: D (4+4+4 = 12)

Path: S → A → C → G

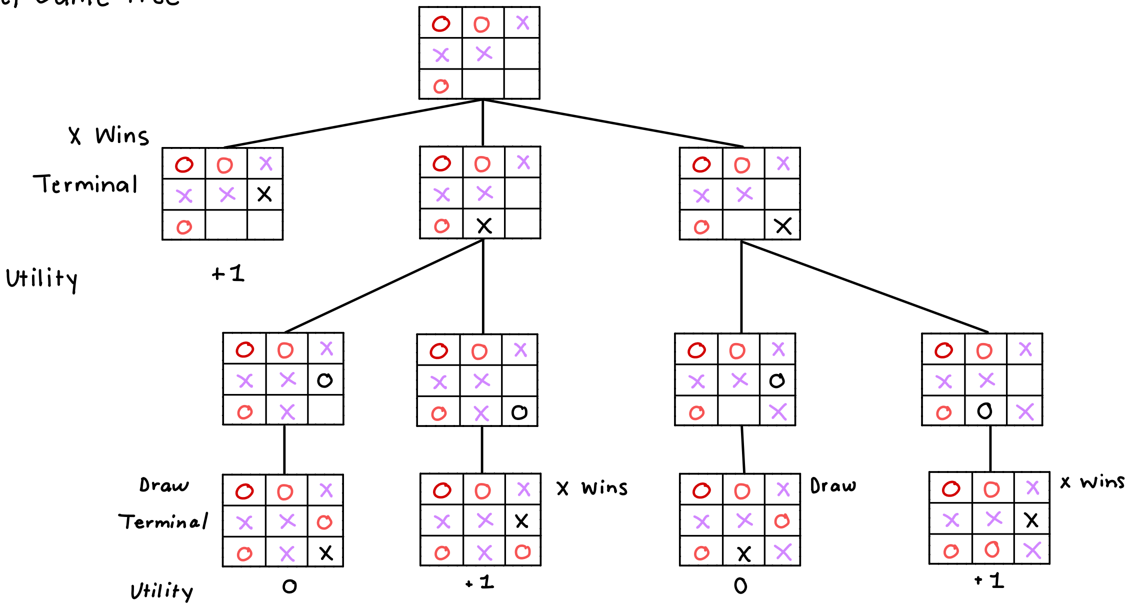
Total Cost: 2+5+3 = 10

c)



Problem 4: Tic-Tac-Toe Minimax

a) Game Tree



b) How many layers (space complexity) in your tree (including root)?

Root
Max(X) : after X moves
Min (O) : after O moves
Terminal : Final X move

There are 4 layers

c) How many terminal states / nodes?

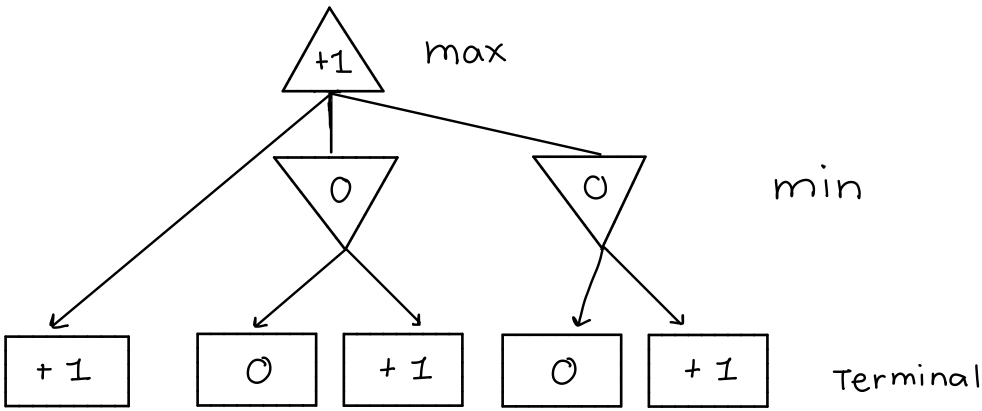
5 terminal nodes

Branch 1 = 1 - X Wins

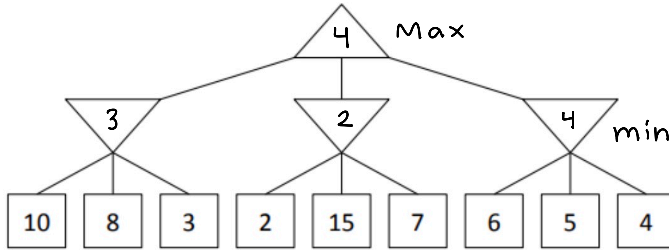
Branch 2 = 2 - X Wins + Draw

Branch 3 = 2 - X wins + Draw

d) Zero-Sum Game Tree w/ Minimax Values



Problem 5:



Min(10,8,3) Min(2,15,7) Min(6,5,4)

Root (Max) = 4

Which nodes can be pruned from game tree?

α = Max

β = Min

No pruning possible for first subtree

After the first subtree, we update α at root to 3.

In 2nd subtree, the first value at Min node is 2.
Since $2 \leq 3$, we can prune the other children (15 and 7) because the Min node will return a value less than or equal to 2, and as Max has 3, the branch won't be chosen by Max.

No pruning for 3rd subtree because the minimum value doesn't become less than or equal to 3.