

HW3
CS 171 Intro to AI
Winter 2017

Due at EEE class dropbox: Thursday Feb 9, 11:59PM

Chapter 5
Problems 1-4:

See following two pages for problems 1-4 on game tree search, which you solve just by “playing computer”.

Chapter 6

We may or may not grade these, depending on time. But they are good midterm study questions so you should try them.

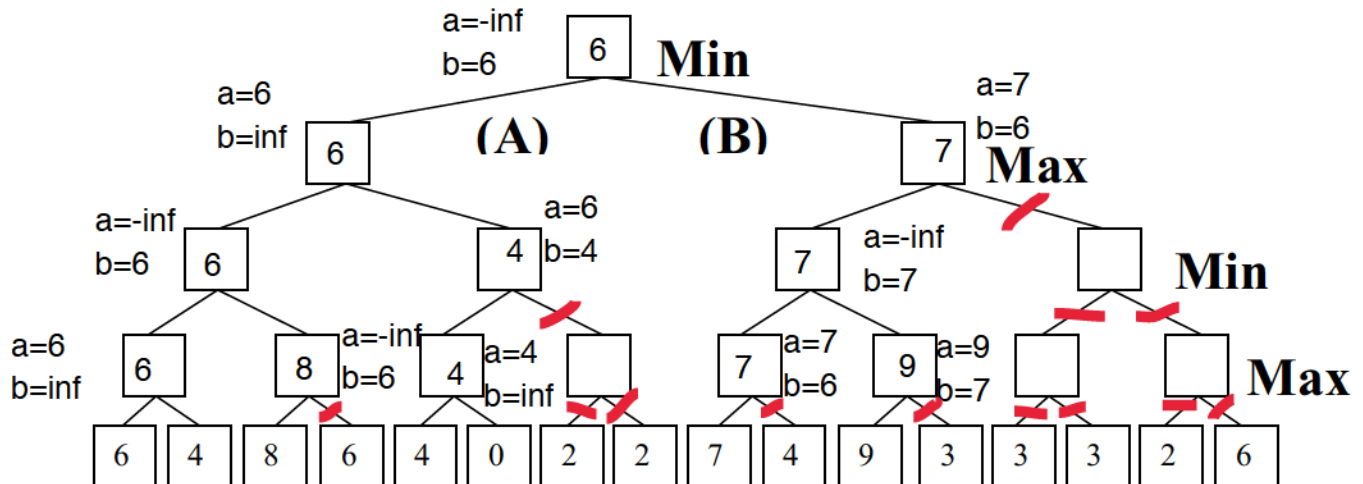
5. Problem 6.4, R&N 3rd ed. page 231.
6. Problem 6.5, R&N 3rd ed. page 231.
7. Problem 6.13, R&N 3rd ed. page 232.

Questions to:

Piazza site; Discussion section on Wednesday; TA (Junkyu Lee) - see syllabus.

(25 pts total, 5 pts off each wrong answer, but not negative) MINI-MAX SEARCH IN GAME TREES.

1.a. The game tree below illustrates a position reached in the game. It is MIN's turn to move. Inside each leaf node is the estimated score of that resulting position returned by the heuristic static evaluator. FILL IN EACH BLANK SQUARE WITH THE PROPER VALUE ACCORDING TO MINI-MAX SEARCH.



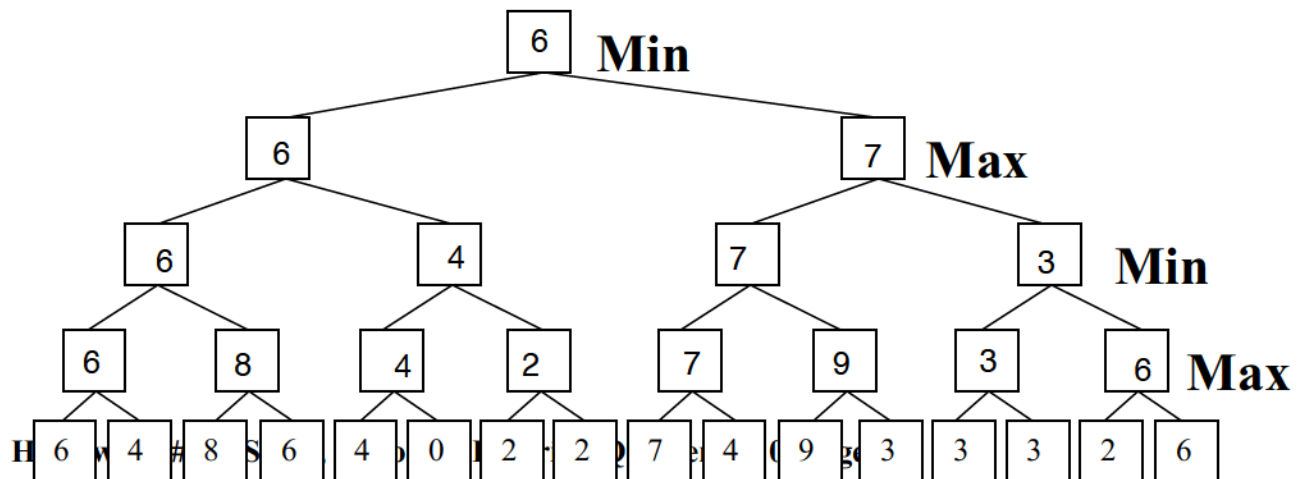
1.b. What is MIN's best move (write A or B) A

1. ~~2.~~

2. (25 pts max, -5 for each error, but not negative) ALPHA-BETA PRUNING.

This is the same tree and conditions as above. CROSS OUT EACH LEAF NODE THAT WILL NOT BE EXAMINED BECAUSE IT IS PRUNED BY ALPHA-BETA PRUNING.

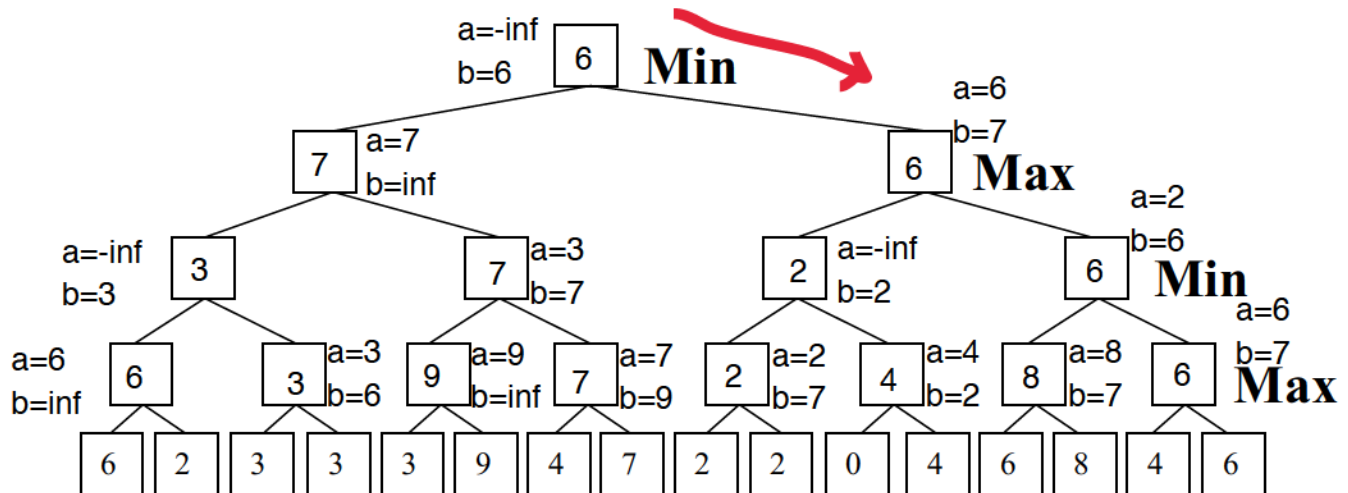
You do not need to indicate the branch node values again.



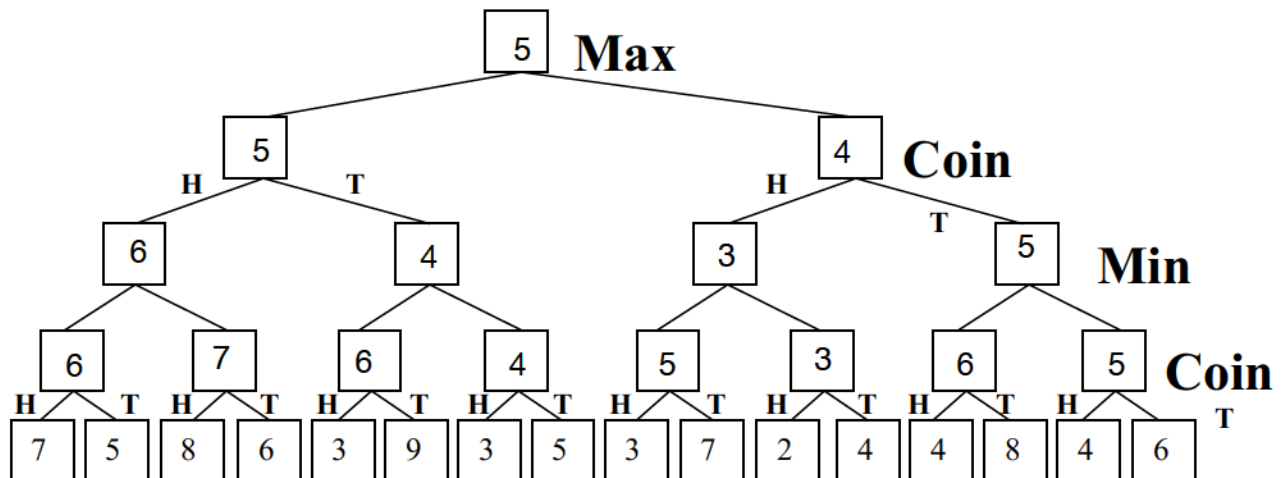
3. (25 pts max, -5 for each error, but not negative) ALPHA-BETA PRUNING.

This is the same conditions as above, except the tree is now the mirror image. CROSS OUT EACH LEAF NODE THAT WILL NOT BE EXAMINED BECAUSE IT IS PRUNED BY ALPHA-BETA PRUNING.

You are not obliged to indicate the branch node values, though you will probably find it helpful to do so.



4. (25 pts max, -5 for each error, but not negative) The following problem asks about MINIMAX search in game trees with chance (also called EXPECTI-MAX search). This is just like MINI-MAX, but there is a coin flip ("COIN") between each player's move. If the coin turns up heads, the path labeled "H" is taken; if it turns up tails, the path labeled "T" is taken. The value passed upward from the COIN level is the probabilistic weighted average (the expected value) of the two paths "H" and "T". Assume a fair coin, i.e., $P(H)=P(T)=0.5$. Please think carefully about how to pass values upwards from the COIN level before working this problem. The game tree below illustrates one position reached in the game. It is MAX's turn to move. Below the leaf nodes are the estimated score of each resulting position returned by the heuristic static evaluator. FILL IN ALL BRANCH NODES WITH THE VALUES PASSED UPWARDS FROM THE LEAF NODES.



Ch. 6

5.

a) For each rect R_i w/ width, w , and height, h :

Variables = $\{x, y, w, h\}$

Constraints:

$$R_{ix} \geq 0$$

$$R_{ix} + R_{iw} \leq w$$

$$R_{iy} \geq 0$$

$$R_{iy} + R_{ih} \leq h$$

} Confines small rects to the large rect.

There are also constraints between rects $R_i \neq R_j$:

$$R_{ix} + R_{iw} \leq R_{jx} \text{ or } R_{ix} \geq R_{jx} + R_{jw}$$

$$R_{iy} + R_{ih} \leq R_{jy} \text{ or } R_{iy} \geq R_{jy} + R_{jh}$$

b) Variables = $\{\text{Teachers, Subjects, Classrooms, Time slots}\}$

Constraints:

Given: T_{ij} = teacher in classroom i at time j .

S_{ij} = set of subjects

$$\rightarrow T_{ij} \neq T_{kj}, \quad k \neq i$$

let $D(t)$ = subjects that teacher, t , can teach

then $C(t) = t$ is assigned to $T_{ij} \rightarrow S_{ij}$ is assigned value from $D(t)$

6. Variables = $\{T, W, O, F, U, R\}$

Constraints:

$$O + O = R + 10(x_1)$$

$$X_1 + W + W = U + 10(x_2)$$

$$X_2 + T + T = O + 10(x_3)$$

$$X_3 = F$$

All diff (F, O, U, R, T, W)

Domains:

$$- D_F, D_{x_3} = \{1\}$$

$$- D_{x_2}, D_{x_1} = \{0, 1\}$$

$$- D_T = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$$

$$- D_W, D_O, D_U, D_R = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$$

$$D_{x_1}, D_{x_5} = \{1, 2\}$$

$$D_{x_2}, D_{x_4} = \{0, 1, 3\}$$

$$D_{x_3} = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$$

$$D_w = D_r = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$$

$$F = 1$$

$$(1, 6, 7, 2, 8, 3, 0, 0, 1)$$

$$X_3 = 1$$

$$X_2 = 0$$

$$D_u = \{3, 5, 7, 9\}$$

$$X_1 = 0$$

$$D_w = \{0, 3, 4\}$$

$$O = 2$$

$$W = 3$$

$$X_1 = 1$$

$$836$$

$$O = 6$$

$$+ 836$$

$$W = 3$$

$$1672$$

7. With constraint variable, c , with domain as a set of all possible 3-tuples satisfying $A+B=C$, we can define a new set of constraints:

$$c[A] \in A$$

$$c[B] \in B$$

$$c[C] \in C$$

which enforces a binary constraint which is satisfied only if tuple, t , in the domain is c is compatible with value, x , in domain $A/B/C$ such that $t[A/B/C] = x$.