Artificial Intelligence
Practice Final Exam
May, 2022

We will go over this in May 9 final exam review, so best to do this in advance to prepare.

These are more than 100 points of questions (longer than the actual final will be). The point value serves to give an idea how much it might be worth.

Problem 1: 20 points

Suppose that you have a training set T with 10,000 instances. The classification attribute has 5 values A, B, C, D, and E, with frequencies 0.05, 0.3, 0.4, 0.2, 0.05. There is some fixed representation of the predictive attributes that we will ignore.

a. (1 pt) Ignoring frequencies, indicate how many bits a traditional fixed-width encoding would take for the classification attribute of this training set.

5 values requires 3 bits, so 3 * 10,000 = **30,000 bits**

b. (4 pts) Indicate a prefix-free code for these values that would require the minimum number of bits to encode the classification attribute of this training set.

Map the most frequent items to the shortest codes, so:
C -> 0, B -> 10, D -> 110, A -> 1110, E -> 1111 (you could swap A with E)

c. (5 pts) Record how many bits would be needed to encode the classification attribute of this training set using the code from b). Include a decoding table using a "." separator as in class examples.

Using this code and instances of: 500 A, 3000 B, 4000 C, 2000 D, 500 E
500*4 + 3000*2 + 4000*1 + 2000*3 + 500*4 = 20,000 bits

A decoding table using "." would be 1110.10.0.110.1111, requiring 2 bits per char gives 18*2=36

Total = 20,036 bits

You have trained a classifier for this data with 80% accuracy. The classifier uses 128 bits to represent it. The classifier never produces a false negative for A or B; that is, you will never need to correct to A or B, but sometimes a produced {A,B,C,D,E} will need to get corrected to {C,D,E}

d. (5 pts) Design a prefix-free code to combine with this classifier and encode the classification attribute.

Key: the 20% inaccuracy needs to be spread over mis-classified C, D, E which means .4/.65 C, .2/.65 D and .05/.65 E; this is just a prediction of course

Count cases and assign the most to the shortest codes – do not need codes for A, B:
0 -> classifier correct; since that covers 8000 of cases
10 -> incorrect, actually “C” ; 1231 cases (rounded up)
110 -> incorrect, actually “D” ; 615 cases (rounded down)
111 -> incorrect, actually “E” ; 154 cases (complement)

Note: would be ok if your rounding was say 1230 and/or 616 and 153-155

e. (5 pts) Record how many bits would be needed to encode the training set using the code from d). Include a decoding table using a “.” separator as in class examples, make sure to include empty for the elements that do not require a code.

Using the above code:
8000 * 1 + 1231 * 2 + 615 * 3 + 154 * 3 = 12,769 bits

A decoding table might be: 0...10.110.111
reflecting that there is no code to correct back to A or B
9+5=14*2=28 bits

Total including classifier: 12,769 + 128 + 28 = **12,925 bits**

Problem 2: 15 points

P ^ Q <=> C v D
D => ¬H
¬(P ^ Q)
¬(P ^ E)
B => D
C => ¬(P ^ ¬E)
C v G
D => ¬(¬E ^ G)

a. (5 pts) Convert the above sentences into CNF. Show your work.
b. (10 pts) Trace an execution of DPLL on your CNF clauses with the following modifications:
    - For hard cases always guess the lowest alphabetical atom, true first.
    - For easy cases, indicate all easy cases available, then apply the lowest alphabetical

A.
1. (P ^ Q => C v D) ^ (C v D => P ^ Q)

2.
[¬(P ^ Q) v C v D] ^ [¬(C v D) v P ^ Q]
¬D v ¬H
¬(P ^ Q)
¬(P ^ E)
¬B v D
¬C v ¬(P ^ ¬E)
C v G
¬D v ¬(¬E ^ G)

3.
[¬P v ¬Q v C v D] ^ [¬C ^ ¬D v P ^ Q]
¬D v ¬H
¬P v ¬Q
¬P v ¬E
¬B v D
¬C v ¬P v E
C v G
¬D v E v ¬G

4. Just second part of first clause
¬C v P ^ Q
¬D v P ^ Q
Then again for final set:
¬C v P
¬C v Q
¬D v P
¬D v Q
¬P v ¬Q v C v D
¬D v ¬H
¬P v ¬Q
¬P v ¬E
¬B v D
¬C v ¬P v E
C v G
¬D v E v ¬G

B.

¬C v P ^ Q

¬D v P ^ Q

¬C v P

¬C v Q

¬D v P

¬D v Q

¬P v ¬Q v C v D

¬D v ¬H

¬P v ¬Q

¬P v ¬E

¬B v D

¬C v ¬P v E

C v G

¬D v E v ¬G

- easy case pure-literal H=false

¬C v P ^ Q

¬D v P ^ Q

¬C v P

¬C v Q

¬D v P

¬D v Q

¬P v ¬Q v C v D

¬P v ¬Q

¬P v ¬E

¬B v D

¬C v ¬P v E

C v G

¬D v E v ¬G

- easy case pure-literal B=false, leaves

¬C v P

¬C v Q

¬D v P

¬D v Q

¬P v ¬Q v C v D

¬P v ¬Q

¬P v ¬E

¬C v ¬P v E

C v G

¬D v E v ¬G
- guess C=true
P
Q
¬D v P
¬D v Q
¬P v ¬Q
¬P v ¬E
¬P v E
¬D v E v ¬G
- easy case pure D=false {also G, P,Q}
P
Q
¬P v ¬Q
¬P v ¬E
¬P v E
- easy case singleton P=true {also Q}
Q
¬Q
¬E
E
- easy case singleton E=true {also Q}
Contradiction with ¬E

-- backtrack to before C (but after first 2 easy cases)
¬C v P
¬C v Q
¬D v P
¬D v Q
¬P v ¬Q v C v D
¬P v ¬Q
¬P v ¬E
¬C v ¬P v E
C v G
¬D v E v ¬G
guess C=false
¬D v P
¬D v Q
¬P v ¬Q v D
¬P v ¬Q

¬P v ¬E
G
¬D v E v ¬G
- easy case singleton G=true, leaves

¬D v P
¬D v Q
¬P v ¬Q v D
¬P v ¬Q
¬P v ¬E
¬D v E
- guess D=true
P
Q
¬P v ¬Q
¬P v ¬E
E
- easy case singleton E=true {also P, Q}
P
Q
¬P v ¬Q
¬P
- easy case singleton P=true
- contradiction with singleton ¬P
-- backtrack to guess D=false
¬P v ¬Q
¬P v ¬Q
¬P v ¬E
- easy case pure-literal E=false {also P, Q}
¬P v ¬Q
¬P v ¬Q
- easy case pure-literal P=false {also Q}
Clauses satisfied: but Q unbound, set to default of false
B=false, C=false, D=false, E=false, G=true, H=false, P=false, Q=false

Problem 3: 15 points

	A	B	C	D	E	F	G	H	I	J	K	L
x	2959	903	2935	2118	1663	789	61	1341	2027	1801	1480	1645

y	734	300	2061	2313	984	1370	455	1480	1114	1859	1252	1160

a. Using Euclidean squared distance for $D(p, q) = (p_x - q_x)^2 + (p_y - q_y)^2$ Trace the execution of k-means on the table above using C1=<750,1500>, C2=<2200,500> C3=<1250,1250> as initial centroids, you do not need to show distance tables for each round. If a centroid has no clustered points, keep its coordinates unchanged into the next round.

Round 1:
C1 = {F,G}
C2 = {B,D,E,H,J,K,L}
C3 = {A,C,I}
Re-compute centroids as mean of clustered points
C1 = ([425 912.5])
C2 = ([1564.4 1335.4])
C3 = ([2640.3 1303])

Round 2:
C1 = {B,F,G}
C2 = {D,E,H,I,J,K,L}
C3 = {A,C}
Re-compute
C1 = ([584.3 708.3])
C2 = ([1725 1451.7])
C3 = ([2947 1397.5])

Round 3
Clustering unchanged. Terminate.

b. Using the same data, trace the execution of hierarchical clustering down to 3 clusters using complete-linkage and manhattan distance of $D(p, q) = \text{abs}(p_x - q_x) + \text{abs}(p_y - q_y)$

{A} {B} {C} {D} {E} {F} {G} {H} {I} {J} {K} {L}

E and L closest at 194
{A} {B} {C} {D} {F} {G} {H} {I} {J} {K} {E L}

H and K closest at 367
{A} {B} {C} {D} {F} {G} {H K} {I} {J} {E L}

E,L and I closest at 494
{A} {B} {C} {D} {F} {G} {H K} {J} {{E L} I}

D and J closest at 771
{A} {B} {C} {D J} {F} {G} {H K} {{E L}I}

F and H,K closest at 809
{A} {B} {C} {D J} {G} {F {H K}} {{E L}I}

B and G closest at 997
{A} {B G} {C} {D J} {F {H K}} {{E L}I}

C and D,J closest at 1336
{A} {B G} {C {D J}} {F {H K}} {{E L}I}

E,L,I and F,H,K closest at 1494
{A} {B G} {C {D J}} {{F {H K}} {E L} I}}

A and C,D,J closest at 2420
{B G} {A {C {D J}}} {{F {H K}} {E L} I}}

Terminate with 3 clusters

Extra Credit: (5 pts) Suppose that all points where x+y>3000 have the label ‘a’, and the other points are labeled ‘b’. Is the equation 3x+4y = 10000 a linear separator for this data? Is it a Support Vector Machine? Explain why or why not for both.

Yes, this is a linear separator as all points with the ‘a’ label (and no ‘b’) are on one side of this equation.

However no, this is not a support vector machine for these points as it does not maximize the margin between the two sets of data. Without doing too much math, a simple plot reveals the support vectors of (1645, 1160, ‘b’), (2027,1114, ‘a’), (1801,1859, ‘a’), and plotting this equation you can see that it is not that.

Problem 4: 20 points

You have a box with 2 coins:

- an “up” coin that comes heads 70% of the time, and
- a “down” coin that comes up tails 60% of the time.

a. (2 pts) You randomly select a coin and flip it. What is the probability that it comes up heads? Show your work.

P(H) = P(U)*P(H|U) + P(D)*P(H|D)
= .5*.7+.5*.4 = **.55**

b. (3 pts) You randomly select a coin and flip it: it comes up tails. What is the probability that it comes up heads when you flip the same coin again? Show your work.

$P(H|T) = P(H,U|T) + P(H,D|T)$ # count events
$= P(H|U,T)*P(U|T) + P(H|D,T)*P(D|T)$ # conjunctive rule
$= P(H|U)*P(U|T) + P(H|D)*P(D|T)$ # independent flips once coin known

$P(T) = .45$ # complement of P(H)
$P(U|T) = P(T|U) * P(U) / P(T)$ # Bayes Law
$= .3 * .5 / .45 = \frac{1}{3}$
$P(D|T)$ = complement 2/3

Plugging back in
$P(H|T) = .7 * 1/3 + .4 * 2/3 = \mathbf{.5}$

You are offered a game (where x represents your winning payout):

- You randomly select a coin and flip it, noting the outcome
- You now choose to keep the same coin, or swap with the other one to flip again
- If you stay: your opponent automatically bets, winning \$20 against losing ‘x’ to you that another flip of the same coin has the same outcome
- If you switch coins: your opponent automatically bets, winning \$20 against losing ‘x’ to you on the opposite outcome (that is tails if you flipped heads the first time, and vice-versa)

c. (5 pts) Draw a decision tree for this game still using ‘x’ for the payout if you win (\$20 if you lose). You may submit a separate pdf or image file of a hand drawing for this.
d. (10 pts) Solve for the smallest integral value of x that makes this game worth playing. Explain your answer.

-> Solve for the smallest x where E(game)>0

$P(T|T,Stay) = .5$ # complement of P(H|T)
$P(U|H,Stay) = P(H|U) * P(U) / P(H)$ # Bayes Law
$= .7 * .5 / .55 = 7/11\ (.63\underline{63})$
$P(H|H,Stay) = P(H|U)*P(U|H) + P(H|D)*P(D|H)$ # same derivation as above
$= .7 * 7/11 + .4 * 4/11 = 13/22\ (.5\underline{90})$
$P(T|H,Stay) = 9/22$ # complement of P(H|H)

The above assumes stay. If switch, then flip P of which coin you have
$P(H|H,Switch) = P(H|U)*P(D|H) + P(H|D)*P(U|H)$
$= .7 * 4/11 + .4 * 7/11 = 28/55\ (.5\underline{09})$
$P(T|H,Switch) = 27/55$ # complement
$P(H|T,Switch) = P(H|U)*P(D|T) + P(H|D)*P(U|T)$
$= .7 * 2/3 + .4 * 1/3 = 3/5\ (.6)$

P(T|T,Switch) = 2/5 (.4) # complement

E(Stay|H) = x * P(T|H) - 20 * P(H|H) # Stay after heads means a win on tails
= x * 9/22 - 20 * 13/22
E(Stay|T) = x * P(H|T) - 20 * P(T|T) # Stay after tails means a win on heads
= x * .5 - 20 * .5
E(Switch|H) = x * P(H|H,Sw) - 20 P(T|H,Sw) # Switch after heads means a win on heads
= x * 28/55 - 20 * 27/55
E(Switch|T) = x * P(T|T,Sw) - 20 P(H|T,Sw) # Switch after tails means a win on tails
= x * 2/5 - 20 * ⅗
E(H) = max(E(Stay|H), E(Switch|H))
E(T) = max(E(Stay|T), E(Switch|T))
E(Game) = P(H)*E(H) + P(T)*E(T)

Now solve this for x or plug into spreadsheet or lab 3 solver and iterate.
Answer: x=$20 for E(x=20)=$0.198 using policy switch on H, stay on T
since E(x=19) is negative

Extra Credit (5 pts): Suppose that in a higher stakes version of the same game, you win $105 and lose $100, and after the first free play you can pay $2 to replay the game an infinite number of times.
What is the expected value of such a game using discounted future rewards of .9 and what policy should you use. Use a tolerance of 0.001 and steps of 150 in any solver.

Policy: Switch on H, Stay on T, Win or Lose Replay the game.
E(Play)=3.051

Problem 5: 15 points

Let *D* be the domain of people and languages. Let *L* be the first-order language over *D* with the following primitives:

- S(x, a) - person x can speak language a
- C(x, y) - person x can communicate with y
- T(u, v, w) - person u can act as interpreter between persons v, w
- Spanish, Japanese: languages
- Fred, Mary: people

A. (5 points) Represent the following sentences in *L*:

i. For any two languages, there is someone who speaks both.
ii. If someone can communicate with two people, then that person can act as an interpreter between them.
iii. If two people both speak the same language, they can communicate

iv. Fred speaks Spanish
v. Mary speaks Japanese
vi. There is someone who can interpret between Fred and Mary

i. $\forall$x,y $\exists$a S(a, x) ^ S(a, y)
ii. $\forall$x,y,z C(x, z) ^ C(y, z) => T(z, x, y)
iii. $\forall$x,y,a S(x, a) ^ S(y, a) => C(x, y)
iv. S(Fred, Spanish)
v. S(Mary, Japanese)
vi. $\exists$a T(a, Fred, Mary)

B. (10 points) Using resolution theorem proving, show that vi. is a consequence of i.-v. You need to show the intermediate steps of conversion, and show your initial CNF clauses with Skolemization, and then each step of the resolution proof.

i.a. S(Sk0(x, y), x)
i.b. S(Sk0(x, y), y)
ii. ¬C(x, z) v ¬C(y, z) v T(z, x, y)
iii. ¬S(x, a) v ¬S(y, a) v C(x, y)
iv. S(Fred, Spanish)
v. S(Mary, Japanese)
¬vi. ¬T(a, Fred, Mary)

Resolve vi with ii {x -> Fred, y -> Mary}
1) ¬C(Fred, z) v ¬C(Mary, z)
Resolve 1 with iii. {x -> Fred, y -> z}
2) ¬S(Fred, a) v ¬S(z, a) v ¬C(Mary, z)
Resolve 2 with iv. {a -> Spanish}
3) ¬S(z, Spanish) v ¬C(Mary, z)
Resolve 3 with iii. {x->Mary, y ->z}
4) ¬S(z, Spanish) v ¬S(Mary, a) v ¬S(z, a)
Resolve 4 with v. {a->Japanese}
5) ¬S(z, Spanish) v ¬S(z, Japanese)
Resolve 5 with i.a {z->Sk0(x, y), x->Spanish}
6) ¬S(Sk0(Spanish, y), Japanese)
Resolve 6 with i.b {x->Spanish, y->Japanese}
Null clause.

Problem 6: 15 points

Consider the following scheduling problem with N tasks and K processors. The time it takes to complete each task T depends on the processor it is assigned to where actual time = T.length / P.speed.

An example is given below: you have a set of tasks T1-T4 each with a length of time needed to complete. You also have a pair of processors P1 and P2 that offer different speeds. Although both processors can work in parallel, a single processor can only work on one task at a time, so if T1 and T2 are assigned to P1, that takes (24+84)/4 = 27.

The problem is to find an assignment of tasks to processors such that the total time taken is less than some D.

task	T1	T2	T3	T4
length	24	84	96	108

processor	P1	P2
speed	4	6

a. (5 pts) Represent this problem for a blind search. That is: what are the states, operators, as well as start, terminal and goal states? What is the branching factor for your representation?
b. (5 pts) Using your representation from a), Trace the execution of a depth first search on the above data for D=33. Use an ordering from lowest to highest number task and processor.
c. (5 pts) Also for D=33, trace the execution of a simple hill-climbing run (no-sideways motion) assuming the following:
    i. A state is an assignment of all tasks to a processor
    ii. The cost of a state is the max of computing the time needed on each processor - D. E.g. P1={T1,T3,T4} P2={T2} = max(24+108+96/4, 24/6) = max(57,4)=57 - 33 = 24
    iii. There are two actions available:
        1. Move a task to another processor
        2. Swap two tasks
    iv. Start state is all tasks assigned to P1 and thus an initial cost of 45
    v. The search may terminate if cost < 0

a.
States: A list for each processor, plus a list of “unassigned” tasks waiting for assignment
Operator: take an arbitrary task from unassigned and assign to a processor
Start: all tasks in unassigned
Terminal: empty unassigned ; important for blind search to have a termination condition
Goal: terminal state and max(Cost(processor)) < D

b.
Assign T1 to P1 {T1} {}
Assign T2 to P1 {T1,T2} {}
Assign T3 to P1 {T1,T2,T3} {}

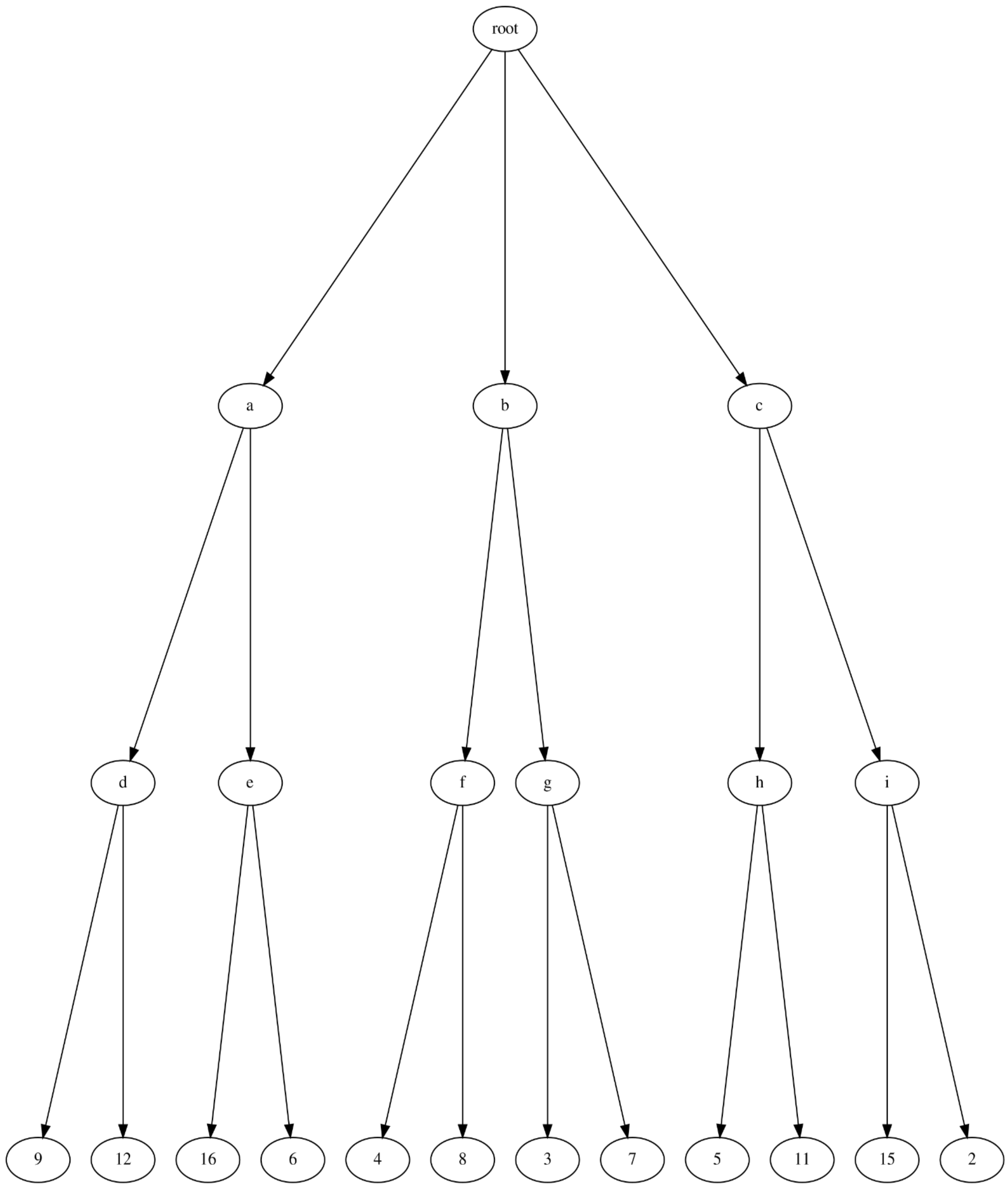
Assign T4 to P1 {T1,T2,T3,T4} {}
Terminal: cost = 78 not goal
Assign T4 to P2 {T1,T2,T3} {T4}
Terminal: cost = 51 not goal
Assign T3 to P2 {T1,T2} {T3}
Assign T4 to P1 {T1,T2,T4} {T3}
Terminal: cost = 54 not goal
Assign T4 to P2 {T1,T2} {T3,T4}
Terminal: cost = 34 not goal
Assign T2 to P2 {T1} {T2}
Assign T3 to P1 {T1,T3} {T2}
Assign T4 to P1 {T1,T3,T4} {T2}
Terminal: cost = 57 not goal
Assign T4 to P2 {T1,T3} {T2,T4}
Terminal: cost = 32 reached goal, terminate.

c.
P1={T1,T2,T3,T4} P2={} Cost = 45
Only move actions:
P1={T2,T3,T4} P2={T1} Cost = 39
P1={T1,T3,T4} P2={T2} Cost = 24
P1={T1,T2,T4} P2={T3} Cost = 21
P1={T1,T2,T3} P2={T4} Cost = 18

Choose T4 to P2: – now 4 moves, 3 swaps
P1={T1,T2,T3,T4} P2={} Cost = 45
P1={T2,T3,T4} P2={T1} Cost = 39
P1={T1,T3,T4} P2={T2} Cost = 24
P1={T1,T2,T4} P2={T3} Cost = 21
P1={T2,T3} P2={T1,T4} Cost = 12
P1={T1,T3} P2={T2,T4} Cost = -1
P1={T1,T2} P2={T3,T4} Cost = 1

Choose move T2 to P2: terminate
P1={T1,T3} P2={T2,T4}

Problem 7: 15 points


root
a
b
c
d
e
f
g
h
i
9
12
16
6
4
8
3
7
5
11
15
2

A. (10 points) Using the above game tree with payoffs for max. Write out all the equations for minimax (as in class and homework 2): which branch would a root max player choose for what value?
B. (5 points) If run again with root playing as max but also with alpha-beta pruning, which nodes would be pruned?

A.
max(root)=max(a,b,c) = max(12, 7, 11) = 12
a = min(d, e) = min(12, 16) = 12
b = min(f, g) = min(8, 7) = 7
c = min(h, i) = min(11, 15) = 11
d = max(9, 12) = 12
e = max(16, 6) = 16
f = max(4, 8) = 8
g = max(3, 7) = 7
h = max(5, 11) = 11
i = max(15, 2) = 15

So max takes branch a for 12.

B.
Entering e with a beta=12, e will be pruned because max can choose 16.
Entering b with alpha of 12, as soon as f returns 8, b (and g) are pruned; entering c with alpha of 12, as soon as h returns 11 to c, c (and i) are pruned.