C1 Artificial Intelligence (25 Points)
Search

a) (11 points) Uniform cost search (also called branch-and-bound search) is a modification of the breadth-first strategy where the node on the fringe with the lowest path cost is expanded. It is straightforward to show that uniform cost search finds the cheapest solution provided that no operators have negative costs. Here we explore operators with negative costs in the context of this search.

   1) Suppose that a negative lower bound `c` is placed on the cost of any given step -- that is, negative costs are allowed, but the cost of a step cannot be less than `c`. Does this allow uniform-cost search to avoid searching the whole tree to guarantee an optimal solution has been found?

   2) Suppose that there is a set of operators that form a loop, so that executing the set in some order results in no net change to the state. If all of these operators have negative cost, what does this imply about the optimal behavior for a rational agent in such an environment?

b) (6 points) Suppose we have a graph search problem where each of the operators have the same cost. Recall that in an informed search `h(n)` is an estimate of the cost of going from the node `n` to a goal node, and `g(n)` is the cost of going from the start node to the node `n`. Suppose that we run a greedy (also called best first) search algorithm with `h(n) = -g(n)`. What sort of search will the greedy search emulate? What sort of search will the greedy search emulate if `h(n) = g(n)`?

c) (4 points) Indicate whether the following statement is true or false and give justification for your answer (no points will be given without the justification being correct): The time and memory complexity of IDA* can be improved by using the A* algorithm to do search in individual iterations.

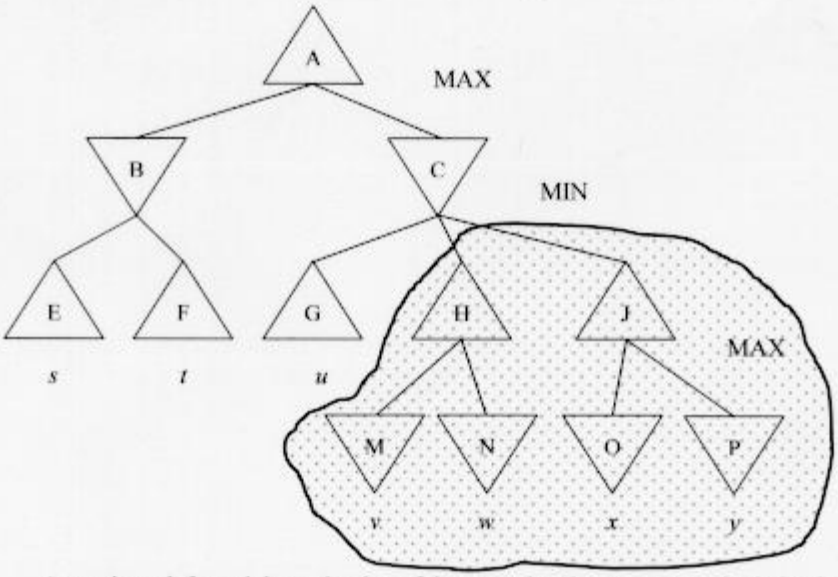
d) (4 points) Indicate whether the following statement is true or false and give justification for your answer (no points will be given without the justification being correct): Consider a uniform search tree where all operators have unit cost, and the optimal goal is at depth 12. If this tree is being searched with A* with a heuristic function, `h`, whose maximum value is 5, then every node at depth less than or equal to 6 is guaranteed to be expanded.

C2 Artificial Intelligence (25 Points)
Game Playing

e) (10 points) Consider a two-person game where there are some number of stacks of coins on the table (each stack possibly containing a different number of coins). A player moves by choosing a stack of coins and splitting it into two unequal sized stacks (e.g., if a player picks a stack of 7 coins, it can be split into two stacks of 5 and 2, or two stacks of 3 and 4, or two stacks of 6 and 1). When the game reaches a stage such that the player whose turn it is now cannot pick a stack from the table and split it this way, the player is considered to have **lost** the game (and the other player wins).

Suppose that **initially there is a stack of 5** coins on the table. Show, using the minimax strategy, that the player with the first move (MAX) will always win this game, no matter whom the opponent (MIN) is. Explain the representation you are using for the game-tree state, and generate the full minimax tree to do this.

f) (4 points) Consider the following game-tree, where *s, t, u, v, w, x,* and *y* are the evaluation values of the leaf nodes from MAX's point of view.



Assuming a left-to-right evaluation of the tree, give the most general relation between *s, t, u v, w, x,* and *y* that must hold such that the tree in the dotted part would be pruned (i.e., would never be considered) by alpha-beta pruning.

g) (4 points) Using the same tree, give the most general relation that must hold so that node N will be pruned by alpha-beta pruning (assuming left-to-right evaluation again). NOTE: in this case the tree in the dotted part is *not* pruned – just the node N.

h) (7 points) Describe a search technique that might be called *iterative deepening α-β search.*

C3 Artificial Intelligence (25 points)

Planning

The robot exam domain consists of 3 operators:

```
study(subject)
taketest(subject)
forget(subject)
```

Unless a subject has been studied and is known by the robot, the robot will fail when taking a test in that subject. Due to the recent increase in memory prices, the robot only has enough memory to remember TWO subjects that have been studied. Studying will fail if the robot tries to study a third subject without forgetting some previous subject.

1. (9 points) Describe these three operators in a POP/STRIPS-style planning language. You will have to develop and use consistently a set of appropriate fluent predicates.
2. (6 points) These operators, together with a description of an initial state and a goal, constitute a planning problem. Assume that there are three subjects, "theory", "software", and "hardware". The goal is to take and pass a test in each of these subjects. Describe the START and FINISH operators correspond to these initial conditions and goals.
3. (5 points) Draw a complete plan for this problem.
4. (5 points) Let us now assume that if you study for a test, that you still may only pass it 90% of the time. BRIEFLY describe why this is a problem for POP-style planners, and how it can be solved.

C4 Artificial Intelligence (25 points)
Decision Trees
Consider the following data for housing loan application risk assessment:

	output catagory (risk)	credit history	debt	income
1	high	bad	high	low
2	high	unknown	high	middle
3	moderate	unknown	low	middle
4	high	unknown	low	low
5	low	unknown	low	high
6	low	unknown	low	high
7	high	bad	low	low
8	moderate	bad	low	high
9	low	good	low	high
10	low	good	high	high
11	high	good	high	low
12	moderate	good	high	middle
13	low	good	high	high
14	high	bad	high	middle

1. (15 points) Calculate the information gain for each attribute, and indicate which attribute would be chosen as the root of a decision tree.

2. (10 points) *Bagging* is a decision tree learning technique where, instead of simply using the initial training set as-is, we produce several, partially replicated training sets by sampling with replacement from the the initial set of training instances. For example, using the initial training set above, we might generate 5 new training sets of 10 elements each, by sampling with replacement from the initial set of 14 elements:

   (7 2 11 8 12 3 6 2 1 7) (12 7 6 14 2 13 4 2 14 5) (2 2 9 3 4 8 4 12 13 11) (11 5 12 5 13 8 8 2 9 8) (8 6 12 11 3 12 7 7 11 7)

   The result will be several decision trees, and to classify an instance we ask each tree to vote on one classification, and then take the majority classification as the output classification of the system. What problems in the standard decision tree learning algorithm might be ameliorated by this technique?

The following tables may be useful.

x	*decimal* x	$\log_2(x)$	$-x\log_2(x)$
1/14	0.07	-3.81	0.27
2/14	0.14	-2.81	0.4
3/14	0.21	-2.22	0.48
4/14	0.29	-1.81	0.52
5/14	0.36	-1.49	0.53
6/14	0.43	-1.22	0.52
7/14	0.5	-1.0	0.5
8/14	0.57	-0.81	0.46
9/14	0.64	-0.64	0.41
10/14	0.71	-0.49	0.35
11/14	0.79	-0.35	0.27
12/14	0.86	-0.22	0.19
13/14	0.93	-0.11	0.1
14/14	1.0	0.0	0.0

x	*decimal* x	$\log_2(x)$	$-x\log_2(x)$
1/7	0.14	-2.81	0.4
2/7	0.29	-1.81	0.52
3/7	0.43	-1.22	0.52
4/7	0.57	-0.81	0.46
5/7	0.71	-0.49	0.35
6/7	0.86	-0.22	0.19
7/7	1.0	0.0	0.0

x	*decimal* x	$\log_2(x)$	$-x\log_2(x)$
1/6	0.17	-2.58	0.43
2/6	0.33	-1.58	0.53
3/6	0.5	-1.0	0.5
4/6	0.67	-0.58	0.39
5/6	0.83	-0.26	0.22
6/6	1.0	0.0	0.0

x	*decimal* x	$\log_2(x)$	$-x\log_2(x)$
1/5	0.2	-2.32	0.46
2/5	0.4	-1.32	0.53
3/5	0.6	-0.74	0.44
4/5	0.8	-0.32	0.26
5/5	1.0	0.0	0.0

x	*decimal* x	$\log_2(x)$	$-x\log_2(x)$
1/4	0.25	-2.0	0.5
2/4	0.5	-1.0	0.5
3/4	0.75	-0.42	0.31
4/4	1.0	0.0	0.0