6.034 Quiz 1
September 24, 2008

Name	
EMail	

Circle your TA and recitation time, if any, so that we can more easily enter your score in our records and return your quiz to you promptly.

TAs
Sam Glidden
Mike Klein
Rick Lestier
Mark Seifter

Thu	
Time	Instructor
1-2	Bob Berwick
2-3	Bob Berwick
3-4	Bob Berwick

Fri	
Time	Instructor
1-2	Howard Shrobe
2-3	Howard Shrobe
3-4	Howard Shrobe

Problem number	Maximum	Score	Grader
1	50		
2	50		
Total	100		

There are 13 pages in this quiz, including this one. In addition, tear-off sheets are provided at the end with duplicate drawings and data.

As always, open book, open notes, open just about everything.

Problem 1: Rule Systems (50 points)

Part A (20 points)

The Smithsonian Institute asks you to put your artificial intelligence programming skills to work identifying strange creatures. They give you a set of rules and assertions, listed below.

Rules:

P0:	IF('(?x) has sharp teeth',
	    THEN('(?x) is a carnivore'))

P1:	IF('(?x) has three horns',
	    THEN('(?x) is a Triceratops'))

P2:	IF(AND('(?x) is a carnivore',
	         '(?x) has big arms',)
	    THEN('(?x) is a Deinonychus'))

P3:	IF(AND('(?x) is a carnivore',
	         '(?x) hunts (?y)',)
	    THEN('(?x) is a TRex'))

P4:	IF(AND('(?x) has sharp teeth',
	         NOT('(?x) is a Deinonychus'),
	         '(?y) is a Triceratops',)
	    THEN('(?x) hunts (?y)'))

Assertions:

A0:	('Spike has three horns')
A1:	('Fido has sharp teeth')
A2:	('Fido has big arms')
A3:	('Rover has sharp teeth')
A4:	('Spike has flat teeth')

Note that when forward chaining, a NOT clause matches if and only if there is NO matching assertion in the list of assertions.

You may make the following assumptions about forward chaining:

- Assume rule-ordering conflict resolution
- New assertions are added to the bottom of the list of assertions.
- If a particular rule matches assertions in the list of assertions in more than one way, the matches are considered in the order corresponding to the top-to-bottom order of the matched assertions. Thus, if a particular rule has an antecedent that matches both A1 and A2, the match with A1 is considered first.

Part A1 (15 points)

Run forward chaining on the rules and assertions provided. For the first two iterations, fill out the first two rows in the table below, noting the rules whose antecedents match the data, the rule that fires, and the new assertions that are added by the rule. For the remainder, supply only the new assertions. You have more than enough room.

	Matched	Fired	New Assertions Added to List of Assertions
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			

Part A2 (5 points)

Suppose the rules were reordered, such that rule P4 was moved to the front (so that P4 is always tried first, before P0). **Briefly describe how the outcome of forward chaining would change.**

Part B: Backwards Chaining (30 points)

Make the following assumptions about backwards chaining:

- When working on a hypothesis, the backward chainer tries to find a matching assertion in the list of assertions. If no matching assertion is found, the backward chainer tries to find a rule with a matching consequent. In case none are found, then the backward chainer assumes the hypothesis is false.
- The backward chainer never alters the list of assertions, so it can derive the same result multiple times.
- Rules are tried in the order they appear.
- Antecedents are tried in the order they appear.
- When backward chaining, the NOT clause matches if and only if there is no matching assertion in the list of assertions and no rules that connect assertions in the list of assertions to the clause.

Part B1 (5 points)

Evaluate the hypothesis 'Fido is a Deinonychus' using backwards chaining. Draw a goal tree in the space below.

```
('Fido is a Deinonychus')
```

Part B2 (20 points)

Now, assume that an additional rule is added:

```
P5:    IF( OR( '(?x) has three horns',
               '(?x) is a Deinonychus' ),
           THEN ( '(?x) is dangerous' ) )
```

Evaluate the hypothesis 'Rover is dangerous' using backwards chaining. List the hypotheses the backwards chainer checks in order in the table **on the next page.** Drawing a goal tree on this page will help us assign partial credit.

('Rover is dangerous')

List, in order, the hypotheses checked by backwards chaining from 'Rover is dangerous'. You have more than enough room:

1.	`Rover is dangerous`
2.	
3.	
4.	
5.	
6.	
7.	
8.	
9.	
10.	

Part B3 (5 points)

Would the addition of rule P6, listed below, change the outcome of backwards chaining? Circle YES or NO.

```
P6:    IF( '(?x) has sharp teeth',
           THEN( '(?x) is a Deinonychus' ) )
```

YES or **NO**

Problem 2: Search(50 points)

Mark Vader is shopping for a new evil stronghold. Starting from his current stronghold, the Depth-First-Search Star, he can explore the available models by either subtracting or adding a single feature. Fortunately, Mark remembers how to perform the search techniques he learned in 6.034 from his mentor Emperor Patricktine.

Part A: Basic Search (25 points)

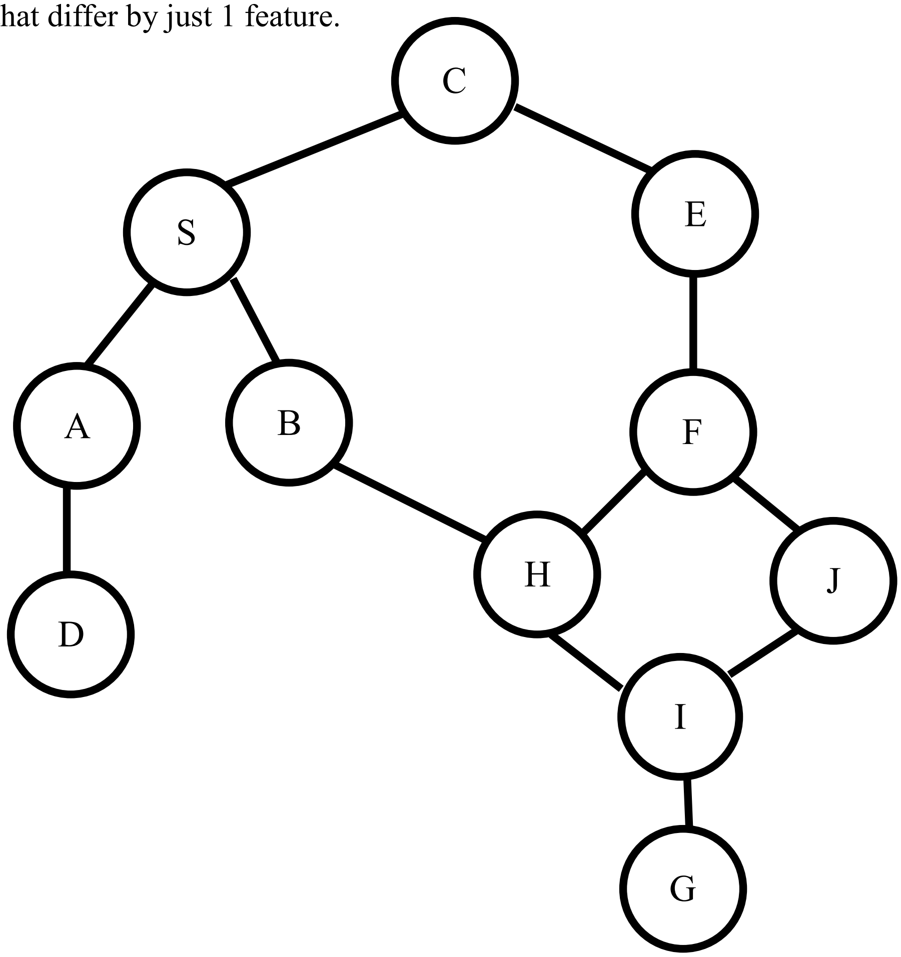
Mark is looking for a stronghold that has the following qualities:

		Exhaust Pipe Weakness	"That's no Moon"	Race of Enslaved Minions	Secret Escape Route	Sharks with Laser Beams
G	6.03Fortress	-	+	+	+	+

Here is a table of 11 possible strongholds, in tie-breaking order.

		Exhaust Pipe Weakness	"That's no Moon"	Race of Enslaved Minions	Secret Escape Route	Sharks with Laser Beams
S	DFS Star	+	+	-	-	-
A	Shayol Ghul	-	+	-	-	-
B	Dol Guldor	+	+	+	-	-
C	Moonraker	+	-	-	-	-
D	Zeal Underwater Palace	-	+	-	+	-
E	Core of Zeromus	+	-	+	-	-
F	Whalers of the Moon Ride	+	-	+	-	+
G	6.03Fortress	-	+	+	+	+
H	Atlantis	+	+	+	-	+
I	Willy Wonka's Factory	+	+	+	+	+
J	Dr. Evil Moon Base	+	-	+	+	+

Being a clever Overlord, Mark also produces this graph of exploration **choices** with edges joining the strongholds that differ by just 1 feature.



Part A1 (10 points)

Mark uses **Depth-First Search with backtracking but no extended list.** He breaks ties according to the order in the table. List the strongholds that Mark **extends**, in order, starting with the Depth-First-Search Star. Use the letters provided. If he extends a single stronghold more than once, list it multiple times. Extra space is provided below in case you want to show your work.

Part A2 (5 points)

How many times did Mark hit a dead end?

Part A3 (5 points)

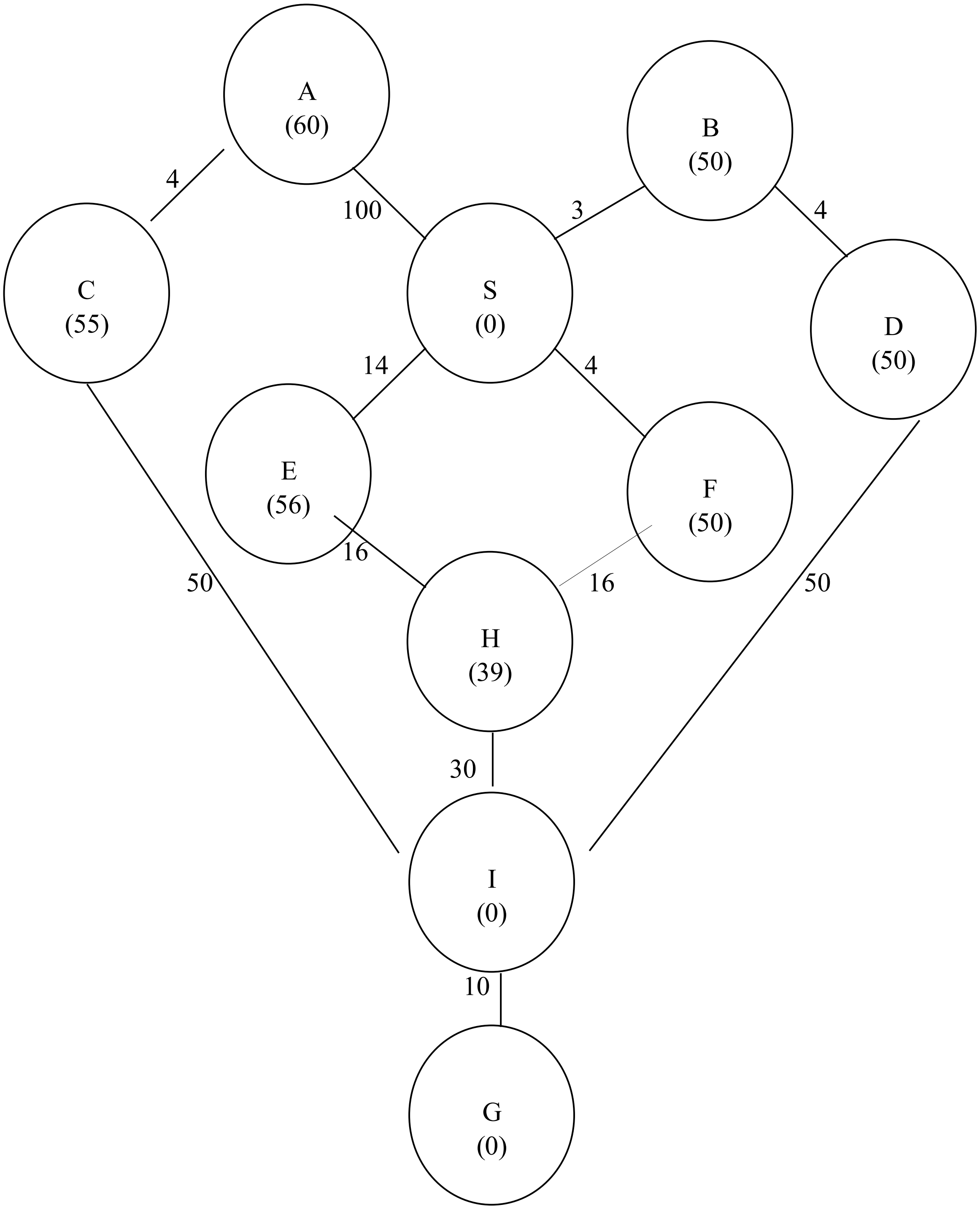
Mark repeats the process with a **Breadth-First Search with an extended list.** What path does he find?

Part A4 (5 points)

Mark considers using **Hill-Climbing with backtracking but no extended list** using the number of features that do not match his ideal stronghold as a heuristic. Would this substantially help Mark's search through the strongholds given in this problem? Explain briefly.

Part B: Optimal Search (25 points)

Now that Mark has his new stronghold, he wants to invade parallel universes. So Mark programs his evil supercomputer to find the shortest path of jumps from his starting universe S to his goal universe G.



Part B1 (10 points)

First, Mark programs a simple **branch-and-bound search with an extended list**. As usual, he breaks ties of equal length in lexicographic order. List the nodes Mark's computer adds to the extended list, in order. Distances are shown next to edges. Ignore the numbers in parentheses for this part of the problem. Extra space is provided below in case you want to show your work.

What path does Mark's computer find?

Part B2 (10 points)

Frustrated by branch-and-bound's speed, Mark reprograms his computer to use **A***. Mark counts the number of subspace anomalies between each universe and the goal and uses this count as the **heuristic** for A* (**these are the numbers in parentheses**). List the nodes Mark's computer adds to the extended list, in order. Extra space is provided below in case you want to show your work.

What path does Mark's computer find now?

Part B3 (5 points)

Mark is confused. Give a **brief** but **specific** explanation of what happened and why.

This page intentionally blank. Use it for scratch work.

You can tear this sheet off of the quiz. You need not hand it in.

From Problem 1

Rules:

P0:	IF('(?x) has sharp teeth',
	    THEN('(?x) is a carnivore'))

P1:	IF('(?x) has three horns',
	    THEN('(?x) is a Triceratops'))

P2:	IF(AND('(?x) is a carnivore',
	         '(?x) has big arms',)
	    THEN('(?x) is a Deinonychus'))

P3:	IF(AND('(?x) is a carnivore',
	         '(?x) hunts (?y)',)
	    THEN('(?x) is a TRex'))

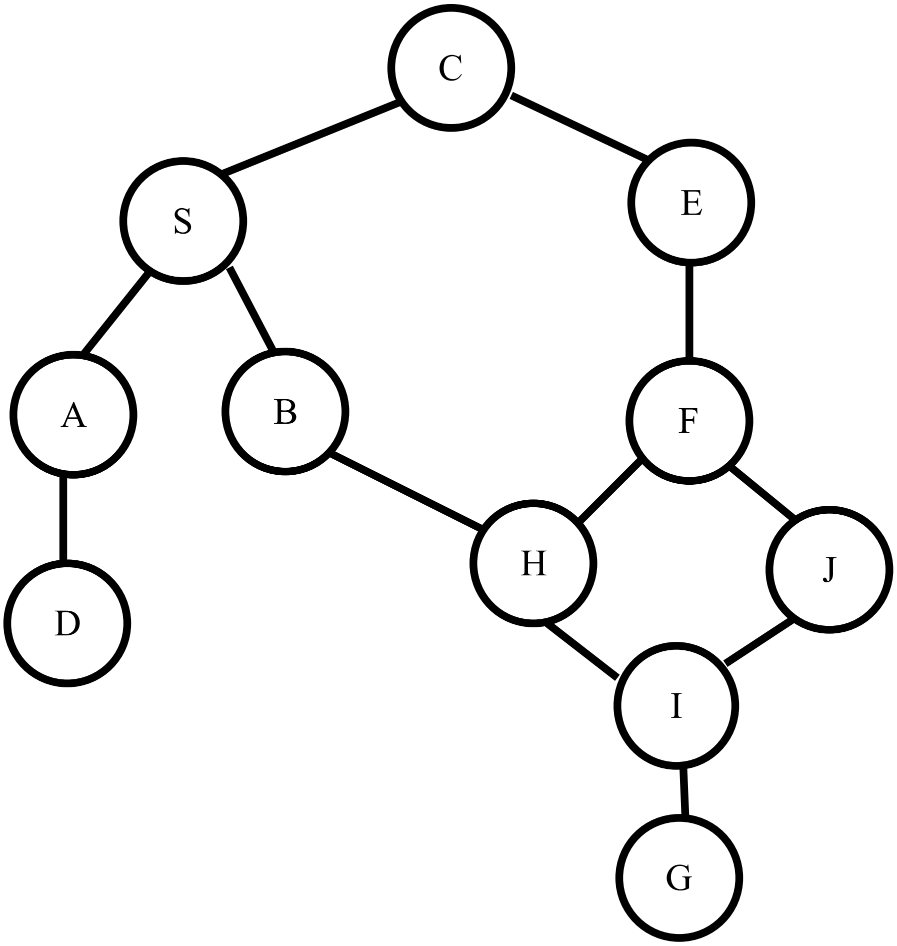
P4:	IF(AND('(?x) has sharp teeth',
	         NOT('(?x) is a Deinonychus'),
	         '(?y) is a Triceratops',)
	    THEN('(?x) hunts (?y)'))

Assertions:

A0:	('Spike has three horns')
A1:	('Fido has sharp teeth')
A2:	('Fido has big arms')
A3:	('Rover has sharp teeth')
A4:	('Spike has flat teeth')

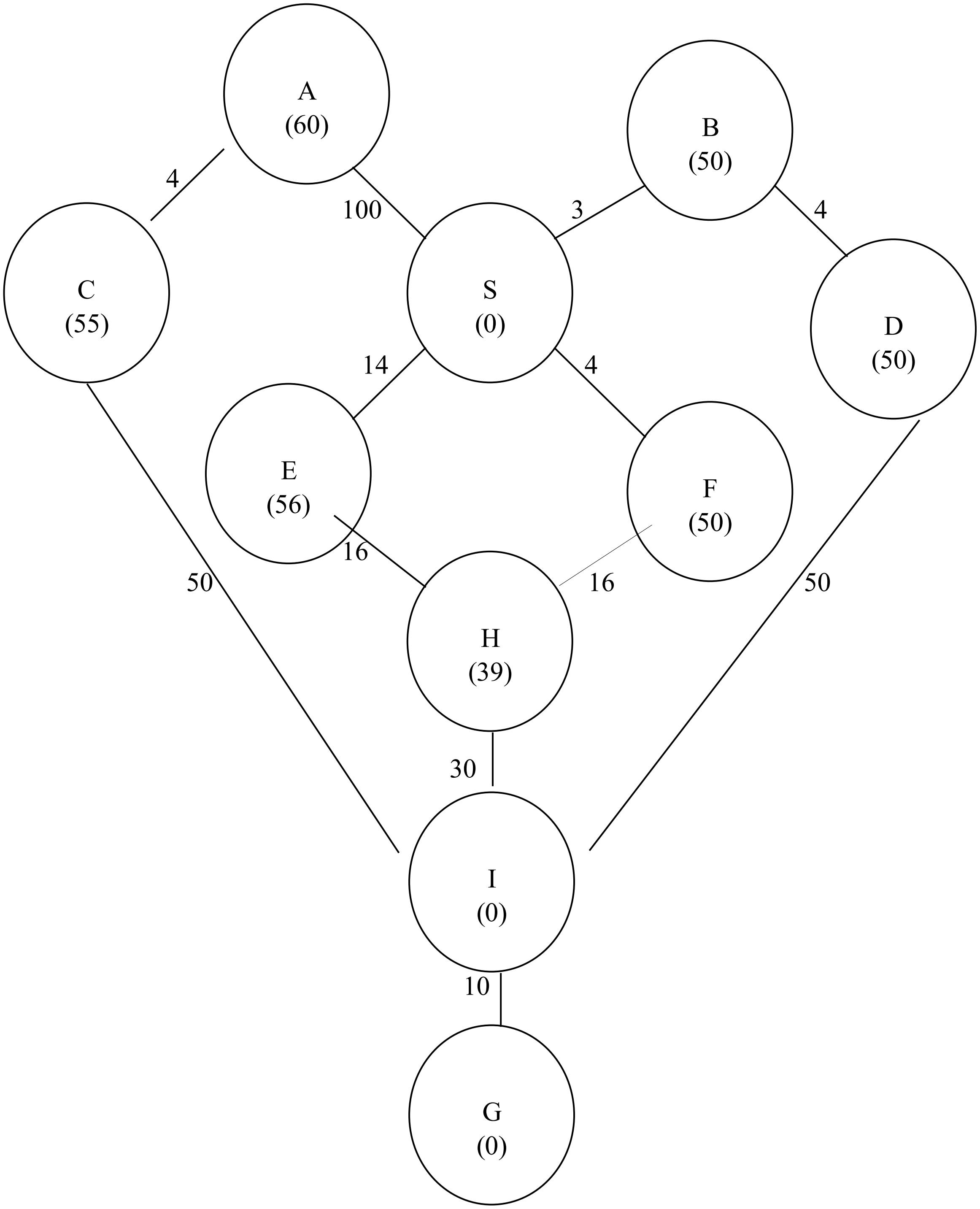
You can tear this sheet off of the quiz. You need not hand it in.

From Problem 2, Part A



You can tear this sheet off of the quiz. You need not hand it in.

From Problem 2, Part B



6.034 Artificial Intelligence
Fall 2010