6.034 Quiz 1
September 30, 2009

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
Erica Cooper
Matthew Peairs
Keith Richards
Mark Seifter
Yuan Shen
Jeremy Smith
Olga Wichrowska

Thu	
Time	Instructor
11-12	Gregory Marton
12-1	Gregory Marton
1-2	Bob Berwick
2-3	Bob Berwick
3-4	Bob Berwick

Fri	
Time	Instructor
1-2	Randall Davis
2-3	Randall Davis
3-4	Randall Davis

Problem number	Maximum	Score	Grader
1	50		
2	50		
Total	100		

There are 11 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)

Due to constant pressure from the 6.034 staff, J. K. Rowling decides to write an 8th Harry Potter book. But, she's suffering from a bad case of writer's block and decides to use a rule-based system to help her with the plot for *Harry Potter and the Deadhorse Principle*. She's given you a set of rules and assertions and would like your help with developing key plot points.

Rules:

P0:	IF (AND('(?x) is ambitious', '(?x) is a squib') THEN '(?x) has a bad term')
P1:	IF ('(?x) lives in Gryffindor Tower') THEN ('(?x) is a protagonist')
P2:	IF (('(?x) lives in Slytherin dungeon') THEN ('(?x) is a villain'), '(?x) is ambitious))
P3:	IF (AND(OR('(?x) is a protagonist', '(?x) is a villain'), '(?x) is ambitious') THEN ('(?x) studies a lot'))
P4:	IF (AND('(?x) studies a lot', '(?x) is a protagonist') THEN ('(?x) becomes Hermione's friend'))
P5:	IF (AND('(?x) snogs (?y)', '(?x) lives in Gryffindor Tower', '(?y) lives in Slytherin dungeon') THEN ('(?x) has a bad term'))

Assertions:

A0:	(Millicent lives in Slytherin dungeon)
A1:	(Millicent is ambitious))
A2:	(Seamus lives in Gryffindor Tower)
A3:	(Seamus snogs Millicent)

Part A: Backward Chaining (15 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.

Part A1

JK knows she would like Millicent to become friends with Hermione. To help her figure out what other assertions must be satisfied, draw the goal tree for the hypothesis:

```
(Millicent becomes Hermione's friend)
```

Part A2

Now, determine the minimum number of additional assertions required for Millicent to become Hermione's friend and list those assertions. **Include no assertion that matches the consequent of a rule.**

Part A3

Your solution to Part A2 creates an uncommon situation. What is that uncommon situation and if J. K. considers the situation to be a problem, what should she do to the list of assertions to solve the problem?

Part B: More Backward Chaining (15 points)

Now, perform backward chaining for the hypothesis **(Millicent has a bad term),** which you may or may not be able to show is true. This time, on the next page, list, in order, the hypotheses checked by backwards chaining from the indicated hypothesis. Draw your tree here to help us award partial credit if you don't get it right.

We recommend that you be extremely careful when you determine how the variables are bound. You are to assume we may be trying to trick you, but we have made no mistakes in formulating the problem.

(Millicent has a bad term)

1.	Millicent has a bad term
2.	
3.	
4.	
5.	
6.	
7.	
8.	
9.	
10.	

Part C: Forward Chaining (20 points)

You may make the following assumptions about forward chaining:

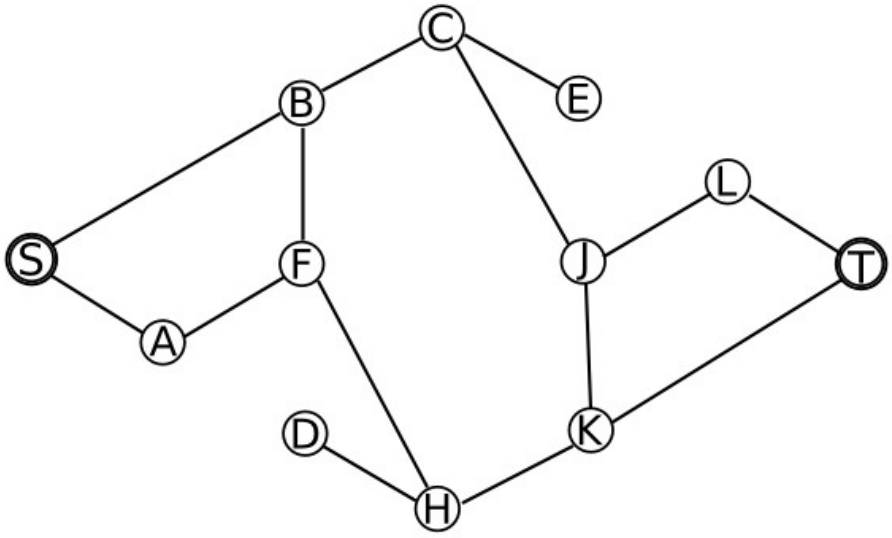
- Assume rule-ordering conflict resolution
- New assertions are added to the bottom of the list of assertions.
- If a particular rule matches assertions in the list of assertions in more than one way, the matches are considered in the order corresponding to the top-to-bottom order of the matched assertions.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

Run forward chaining on the rules and assertions provided on page 2. 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 fired rules and 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			

Problem 2: Search (50 Points)

You have just moved to a strange new city, and you are trying to learn your way around. Most importantly, you want to learn how to get from your home at S to the subway at T.



In all search problems,use alphabetical order to break ties when deciding the priority to use for extending nodes.

Part A (15 points)

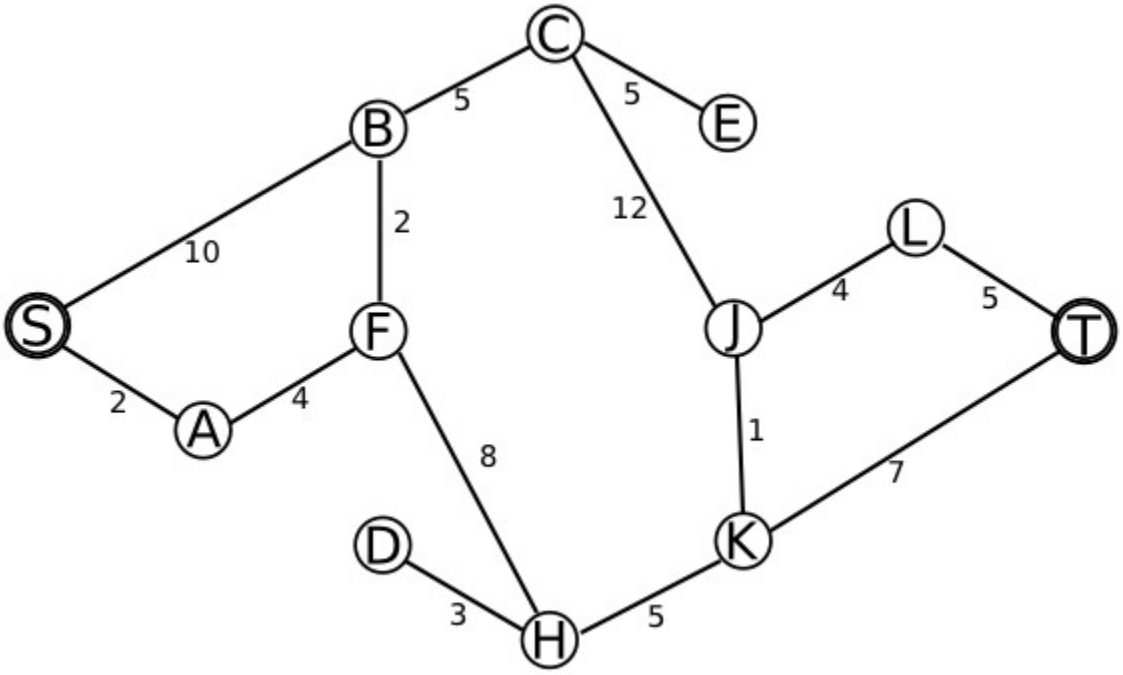
Using depth-first search with backtracking and **with** an extended list, draw that part of the search tree that is explored by the search.

What is the final path found from the start (S) to the goal (T)?

List the nodes at which you have to backtrack:

Part B (15 points)

Some streets have more traffic on them than others. Your friend who has lived in this city for a long time provides you with information about the traffic on each street - the streets are labeled with costs, in the form of how many minutes it will take you to traverse each one.



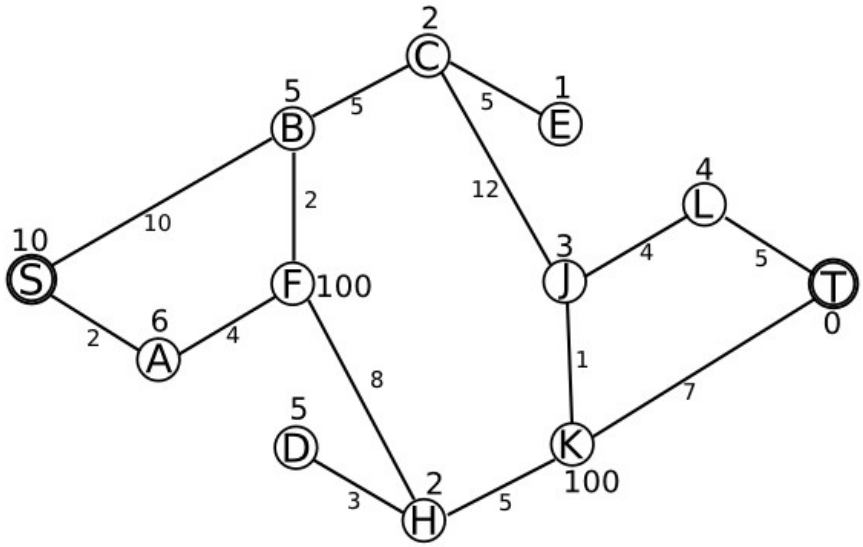
Using these given path costs, you are to find the lowest-cost path from S to T using branch-and-bound **with an extended list but with no distance heuristic.** First draw the search tree. **Number each node as it is expanded, from 1 to n.**

Now identify the shortest path:

After you have found a path to T, which nodes must you still expand before you can be certain that you have found the shortest path to T?

Part C: (20 points)

Now you are to use A* search, expecting to do less work as you find the lowest-cost path from S to T. That is, you are to **use both an extended list and a distance heuristic.** The distance metric is not straight line distance; instead use the numbers provided by an oracle and written immediately above or below each node. For example, the oracle tells you that the estimated distance from node C to the goal, node T, is 2.



First draw the search tree. **Number each node as it is expanded, from 1 to n.**

Now, show the path you have found:

If the path you found using A* is the same as the path you found in Part B, explain in detail why it must be the same; if the path is not the same, explain why your answers are different.

Other than this note, this page is blank. Use if for scratch work if you like.

This is a tear off sheet, with copies of drawings. You need not hand this page in.

Part 1 Rules and assertions

Rules:

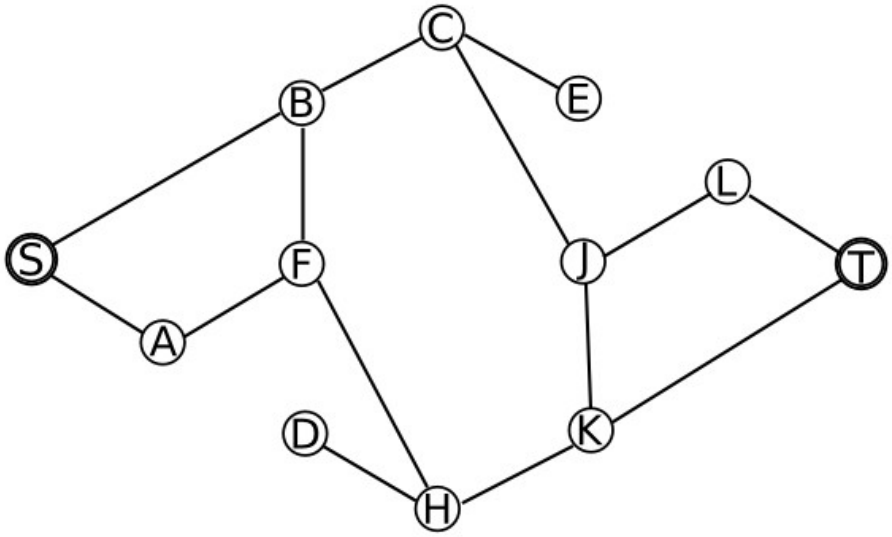
P0:	IF (AND('(?x) is ambitious', '(?x) is a squib') THEN '(?x) has a bad term')
P1:	IF ('(?x) lives in Gryffindor Tower') THEN ('(?x) is a protagonist')
P2:	IF (('(?x) lives in Slytherin dungeon') THEN ('(?x) is a villain'), '(?x) is ambitious))
P3:	IF (AND(OR('(?x) is a protagonist', '(?x) is a villain), '(?x) is ambitious') THEN ('(?x) studies a lot'))
P4:	IF (AND(('(?x) studies a lot'), '(?x) is a protagonist') THEN ('(?x) becomes Hermione's friend'))
P5:	IF (AND('(?x) snogs (?y)'. '(?x) lives in Gryffindor Tower' '(?y) lives in Slytherin dungeon') THEN ('(?x) has a bad term'))

Assertions:

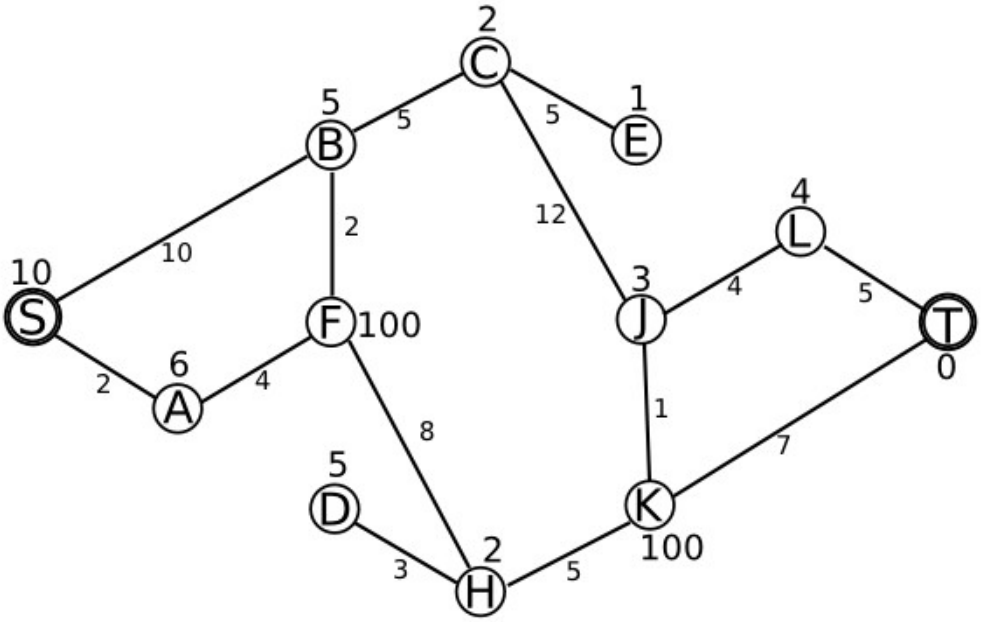
A0:	(Millicent lives in Slytherin dungeon)
A1:	(Millicent is ambitious))
A2:	(Seamus lives in Gryffindor Tower)
A3:	(Seamus snogs Millicent)

This is a tear off sheet, with copies of drawings. You need not hand this page in.

Graph for Part 2A



Graph for Parts 2B and 2C. Part C uses the heuristic distance estimates above and below the nodes. Part B does not.



6.034 Artificial Intelligence

Fall 2010