6.034 Quiz 1

28 September 2016

Name	
Email	

Circle the TA whose recitations you attend **(for 1 extra credit point)**, so that we can more easily enter your score in our records and return your quiz to you promptly.

Jake Barnwell	**Michaela Ennis**	**Rebecca Kekelishvili**
Vinny Chakradhar	**Phil Ferguson**	**Nathan Landman**
Alex Charidis	**Stevie Fine**	**Samarth Mohan**
Brian Copeland	**Brittney Johnson**	**Jessica Noss**

Problem number	Maximum	Score	Grader
1 - Search	35		
2 - Rules	40		
3 - Games	25		
Total	100		

There are 12 pages in this quiz, including this one, but not including tear-off sheets. A tear-off sheet with duplicate drawings and data is located after the final page of the quiz.

As always, open book, open notes, open just about everything, including a calculator, but no computers.

Problem 1: Search (35 points)

Part A: Cory out of the House (10 points)

While walking to the White House with Cory Baxter, you start explaining search algorithms to him. In typical Cory fashion, he understands completely and decides to quiz you instead! He puts a twist on various search algorithms to transform them into different algorithms.

For each modified search algorithm below, circle the one best answer. Some answers may be used more than once, or not at all.

For this part, assume that:

- Non-optimal search algorithms do not use an extended set.
- Branch and bound (B&B) does not use an extended set or a heuristic, unless specified.

1. Branch and bound with a heuristic and extended set is:

   Depth first **Breadth first** **A*** **Hill climbing** **Beam**

2. If you modify breadth-first search so that new paths are added to the front of the agenda instead of the back, you get:

   Depth first **B&B** **A*** **Hill climbing** **Beam**

3. Depth-first search with the entire agenda sorted by path length (least to greatest) is:

   Breadth first **B&B** **A*** **Hill climbing** **Beam**

4. Breadth-first search with the entire agenda sorted by path length (least to greatest) is:

   Depth first **B&B** **A*** **Hill climbing** **Beam**

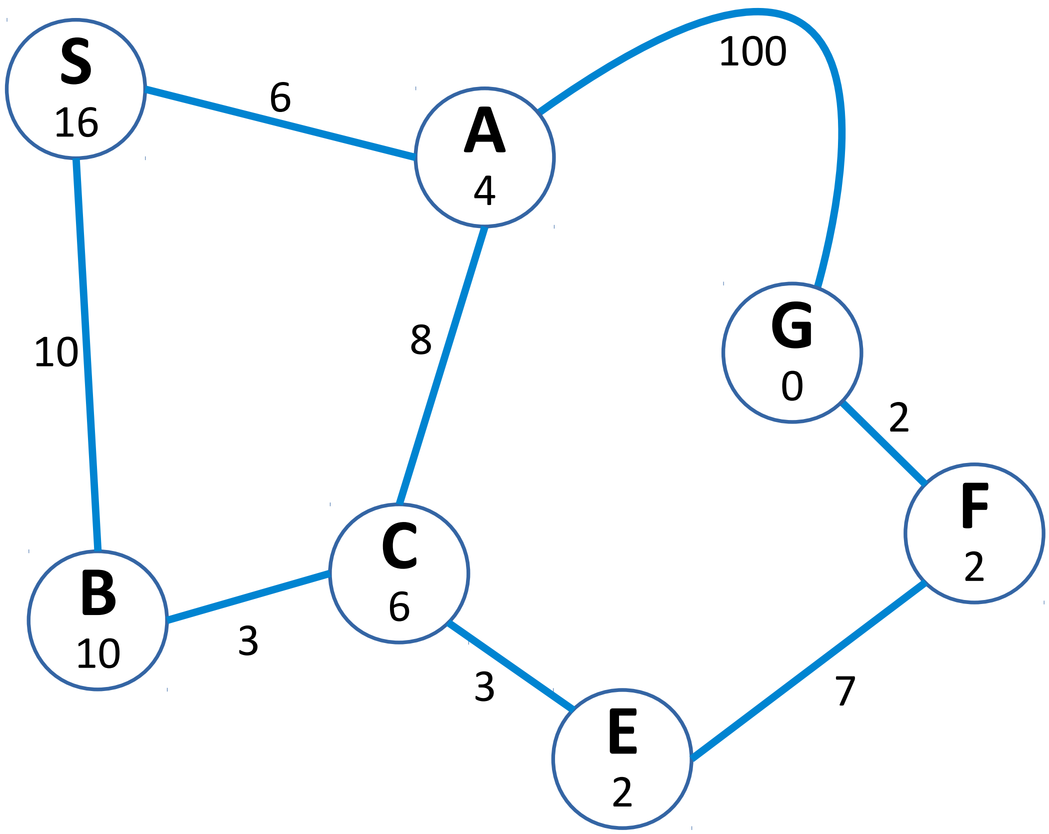
5. For this question, assume that **no** searches use backtracking.
   Beam search with a beam width of 1 is:

   Depth first **Breadth first** **B&B** **A*** **Hill climbing**

Part B: Cory back in the House (17 points)

Now that you and Cory are at the White House, you want to get to the kitchen, because Cory is the son of the head chef and you're hoping for a delicious free meal. Your goal is to find a path from the front door at "**S**" to the kitchen at "**G**" using Cory's map shown below. The map has hallway lengths marked, and numbers inside each node indicate heuristic estimates of the distance to the goal.

For your convenience, a copy of the graph is provided on a tear-off sheet after the last page of the quiz.



B1 (15 points) Use **branch and bound search** with NO heuristic and NO extended set to find a path from "**S**" to "**G**". Draw your search tree in the space provided **on the next page**.

- Draw the children of each node in alphabetical order. (A < B < C)
- Break any ties using lexicographic/dictionary order. (S-A-Z < S-B-Y) (If two paths S-A-Z and S-B-Y are tied, extend path S-A-Z before path S-B-Y because 'SAZ' would come before 'SBY' in the English dictionary.)
- **For partial credit, clearly indicate the order in which you extended nodes** by numbering the extended nodes in your search tree (①, ②, ③, ...).

For credit, draw the search tree for **branch and bound search** in the space below.

B2 (2 points) What is the path that you found using branch and bound search in part B1 above? (Write the path found, including nodes S and G, or write NONE if the search found no path.)

Part C: That's So Shavin' (8 points)

Raven, Cory's sister, has stolen Cory's extremely important razor. She wants to use a more efficient search method to escape from Cory, but first she needs to assess the reliability of the heuristic on Cory's map.

For this part, refer to the graph from Part B (also available on a tear-off sheet).

C1 (3 points): Is the heuristic to "G" **admissible**? If so, circle YES. If not, circle NO and give an example of a node in the graph (such as "A") for which the heuristic is not admissible.

YES **NO. For example:** ☐

C2 (5 points): Is the heuristic to "G" **consistent**? If so, circle YES. If not, circle NO and give an example of a pair of nodes in the graph for which the heuristic is not consistent.

YES **NO. For example:** ☐ **and** ☐

Problem 2: Rule-Based Systems (40 points)

While going on another one of their inter-dimensional adventures, Rick and Morty got stranded on planet Glibglob because their flying car's battery broke in a crash landing. We will use the following rules and assertions to determine whether they can get back to Earth.

For your convenience, a copy of these rules and assertions is provided on a tear-off sheet.

Rules:

P0	IF AND('**(?x)** is kind', '**(?y)** lost his blaster'), THEN '**(?x)** helps **(?y)** find his blaster'
P1	IF OR('**(?x)** has a spare quantum carburetor', AND('**(?x)** is a genius', NOT('**(?x)** is drunk'))), THEN '**(?x)** fixes his car'
P2	IF '**(?x)** helps **(?y)** find his blaster', THEN '**(?x)** gets 10 Flurbos'
P3	IF OR('**(?x)** buys a new car battery', '**(?x)** fixes his car'), THEN '**(?x)**'s car can fly'
P4	IF AND('**(?x)** gets 10 Flurbos', '**(?x)** wants to help **(?y)**'), THEN '**(?y)** buys a new car battery'

Assertions:

```
A0: Rick is a genius
A1: Rick is drunk
A2: Morty is kind
A3: Morty wants to help Rick
A4: Krombopulos lost his blaster
A5: Krombopulos has a spare quantum carburetor
```

Part A: Forward Chaining (20 points)

Using the rules and assertions provided, fill in the table below for the first seven iterations of forward chaining.

- For each iteration, list the rules whose antecedents match the assertions, the rule that fires, and any new assertions that are added. (Some boxes have been filled in for you.)
- If no rules match or fire, or no new assertions are generated, write NONE in the corresponding box.

Make the following assumptions about forward chaining:

- When multiple rules match, rule-ordering determines which rule fires.
- New assertions are added to the bottom of the list of assertions.
- If a particular rule matches in more than one way, the matches are considered in 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.

	Matched	Fired	New assertions added
1		P0	A6: Morty helps Krombopulos find his blaster
2			
3			
4			
5	P0, P1, P2, P3, P4		
6	P0, P1, P2, P3, P4		
7	P0, P1, P2, P3, P4		

Part B: Backward Chaining (20 points)

Make the following assumptions about backward chaining:

- 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 no matching consequents are found, the backward chainer concludes that the hypothesis is false.
- The backward chainer never alters the list of assertions.
- Rules are tried in the order they appear.
- Antecedents are tried in the order they appear.
- Lazy evaluation/short circuiting is in effect.

Using the rules and assertions provided (NOT including any assertions you found in Part A), perform backward chaining starting from the hypothesis:

`'Rick's car can fly'`

- In the table below, write all the hypotheses that the backward chainer checks, in the order they are checked. (The first line has been filled in for you, and the table has more lines than you should need.)
- You can show your work for partial credit: Use the space on the next page to draw the goal tree that would be created by backward chaining from this hypothesis.

1	Rick's car can fly
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	

Space provided to draw goal tree (for partial credit).

Rick's car can fly

Problem 3: Games (25 points)

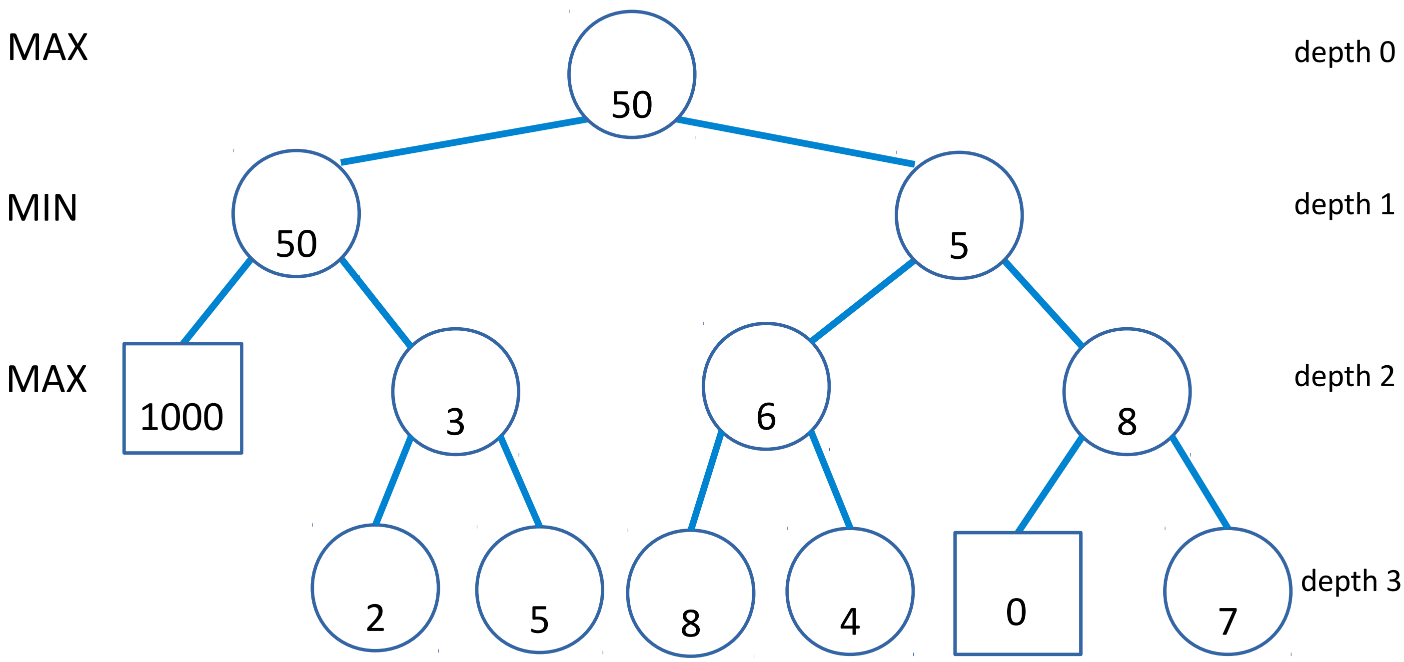
Part A: Max and the mailman (Minimax) (9 points)

Max, the adorable Golden Retriever, is playing a game with the mailman.

Max wants to maximize the score, while the mailman wants to minimize it. It is Max's turn.

Max is not very patient, so he uses minimax **looking only 2 moves ahead**.

The game tree below shows the static values (endgame scores or heuristic scores) up to 3 moves ahead. The static values in square nodes are endgame scores, while the ones in circular nodes are heuristic scores. Keep in mind that you might not need to use all of the static values.



Complete the game tree above using minimax <u>to depth 2</u> with NO pruning:

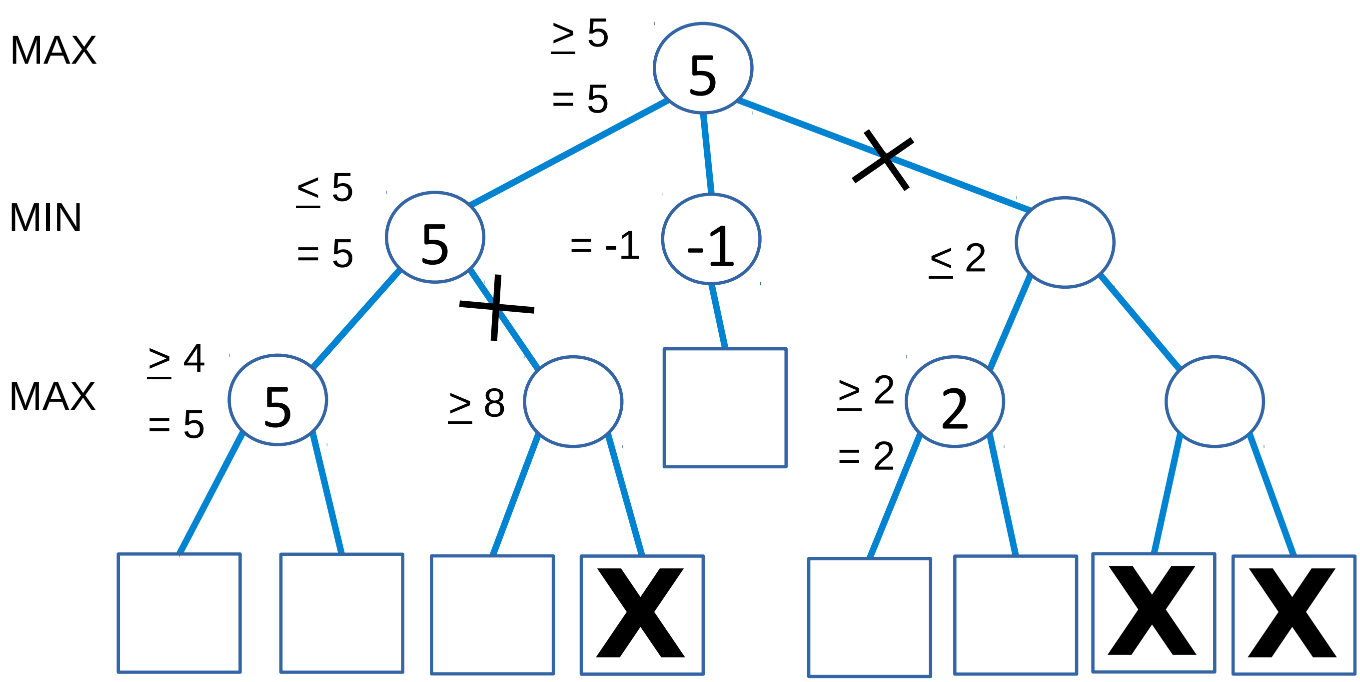
1. In each node that Max will consider, **<u>write the minimax score</u>** just above the static value.
2. **<u>Circle the minimax path corresponding to Max's search</u>** (the optimal sequence of moves that Max and the mailman will play, based on Max's search).

Part B: Reverse Alpha-Beta (8 points)

Max is studying for a 6.034 quiz and remembers a particularly helpful problem from recitation. Unfortunately, he took incomplete notes, so he only has the partially completed alpha-beta game tree pictured below. Notably, Max forgot to record any static values for leaf nodes (square nodes below).

Max's tree follows the same conventions from lecture and recitation:

- Expressions written next to nodes (e.g. "**≥ 5**") indicate best guaranteed scores.
- "**X**" on a leaf node indicates that the node was pruned and not statically evaluated.
- "**X**" on a branch indicates that the player can stop considering that branch.

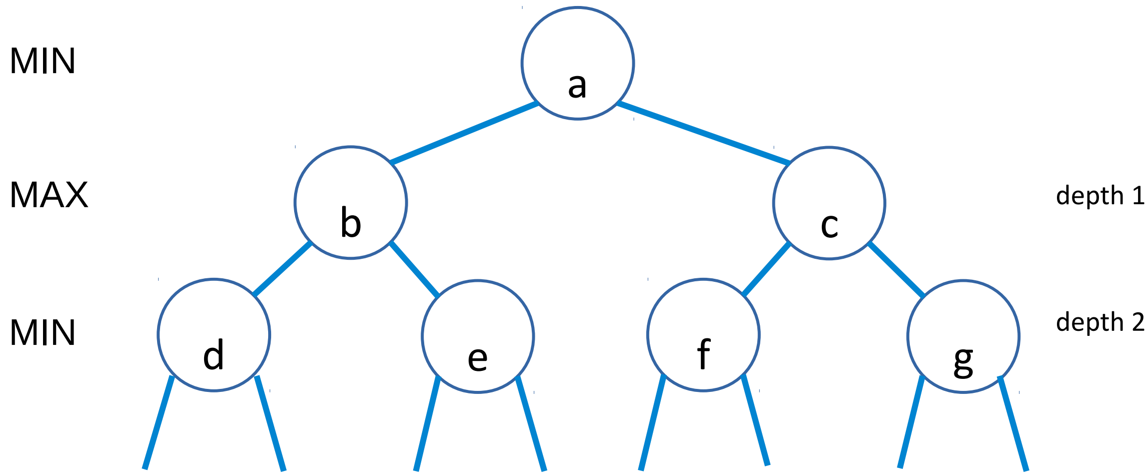


On the tree above, help Max fill in the missing leaf values. **For each non-crossed-out leaf node, write the original value of the leaf in the square.** If it is impossible to determine the original value, write an expression describing the range of possible values (such as "**> 3**").

Part C: Progressive Deepening & Reordering (8 points)

In a new game, the mailman (playing as MIN) is doing progressive deepening on a game tree. He performs minimax with alpha-beta pruning to depth 2 and finds that no nodes can be pruned! **Before proceeding to depth 3, the mailman reorders the nodes** in the tree to maximize the chances of pruning in the next round of progressive deepening.

After reordering the nodes at depths 1 and 2, but before searching to depth 3, the mailman's tree looks like this:



The mailman forgot to keep track of the results from performing minimax to depth 2, so all he has are the static values (heuristic scores) at each node, represented by lowercase letters.

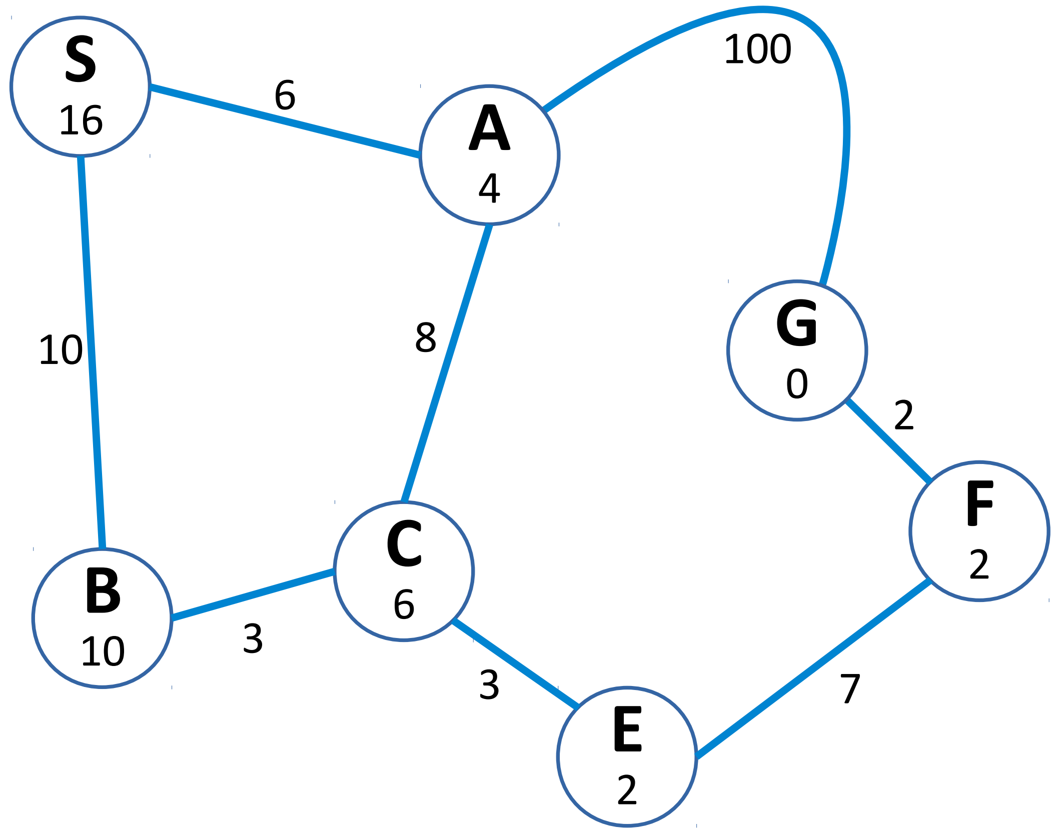
Note that the lowercase letters represent the static values at each node, **NOT minimax scores that were propagated up the tree**.

Consider the mailman's reordered tree. Given that he reordered the tree to maximize the chances of pruning, which of the following conditions on the <u>heuristic scores</u> **must** hold? (Circle all that apply)

d ≥ e	**e ≥ f**	**f ≥ g**	**b ≥ a ≥ c**	**b ≥ c**	**min(d, e) ≥ min(f, g)**
d ≤ e	**e ≤ f**	**f ≤ g**	**b ≤ a ≤ c**	**b ≤ c**	**max(d, e) ≤ max(f, g)**

(Tear-off sheet)

Graph for Problem 1, Parts B & C



(see reverse for Problem 2 rules)

(Tear-off sheet)

Rules for Problem 2, Parts A & B

Rules:

P0	IF AND('**(?x)** is kind', '**(?y)** lost his blaster'), THEN '**(?x)** helps **(?y)** find his blaster'
P1	IF OR('**(?x)** has a spare quantum carburetor', AND('**(?x)** is a genius', NOT('**(?x)** is drunk'))), THEN '**(?x)** fixes his car'
P2	IF '**(?x)** helps **(?y)** find his blaster', THEN '**(?x)** gets 10 Flurbos'
P3	IF OR('**(?x)** buys a new car battery', '**(?x)** fixes his car'), THEN '**(?x)**'s car can fly'
P4	IF AND('**(?x)** gets 10 Flurbos', '**(?x)** wants to help **(?y)**'), THEN '**(?y)** buys a new car battery'

Assertions:

A0: Rick is a genius
A1: Rick is drunk
A2: Morty is kind
A3: Morty wants to help Rick
A4: Krombopulos lost his blaster
A5: Krombopulos has a spare quantum carburetor

(see reverse for Problem 1 graph)