CS 188 Fall 2013 | Introduction to Artificial Intelligence | Written HW4

- **Due:** Monday, September 29, 2014 11:59 PM
- **Policy:** Can be solved in groups (acknowledge collaborators) but must be written up individually. However, we strongly encourage you to first work alone for about 30 minutes total in order to simulate an exam environment. Late homework will not be accepted.
- **Format:** You must solve the questions on this handout (either through a pdf annotator, or by printing, then scanning; we recommend the latter to match exam setting). Alternatively, you can typeset a pdf on your own that has answers appearing in the same space. **Make sure that your answers (typed or handwritten) are within the dedicated regions for each question/part. If you do not follow this format, we may deduct points.**
- **How to submit:** Go to www.pandagrader.com. Log in and click on the class CS188 Fall 2014. Click on the submission titled Written HW 4 and upload your pdf containing your answers. If this is your first time using pandagrader, you will have to set your password before logging in the first time. To do so, click on "Forgot your password" on the login page, and enter your email address on file with the registrar's office (usually your @berkeley.edu email address). You will then receive an email with a link to reset your password.

Last Name	Lee
First Name	Charles
SID	23728976
Email	charleslee94@berkeley.edu
Collaborators	Daniel Kle

For staff use only:

Q1.	CSPs: Midterm 1 Staff Assignments	/25
Q2.	Pacman's new house	/35
Q3.	All Satisfying Assignments	/40
	Total	/100

Q1. [25 pts] CSPs: Midterm 1 Staff Assignments

The CS188 Midterm is coming up, and the CS188 staff has yet to write the test. There are a total of 6 questions on the exam and each question will cover a topic. Here is the format of the exam:

- Q1. Search
- Q2. Games
- Q3. CSPs
- Q4. True/False
- Q5. Agents
- Q6. Short Answer

There are 3 people on the course staff who will be writing questions: John, Kevin, and Anna. Each question will have only one TA assigned to writing it. However, the staff are pretty quirky and want the following constraints to be satisfied:

- Kevin will only work on Search, Games and CSPs.
- Anna is very odd, so she can only contribute to an odd-numbered question.
- The same TA must write both the True/False and the Short Answer question.
- No TA may work on two consecutive questions.

(a) Formulate this problem as a CSP problem in which there is one variable per class, stating the domains, and constraints. Constraints should be specified formally and precisely, but may be implicit rather than explicit.

(i) [7 pts] List the variables and domains after unary constraints are applied.

Put your answer to 1ai here:

1	2	3	4	5	6
J	J	J	J	J	J
K	k	k	~~K~~	~~K~~	~~K~~
A	~~A~~	A	~~A~~	A	~~A~~

1: JKA
2: JK
3: JKA
4: J
5: JA
6: J

(ii) [8 pts] List the constraints.

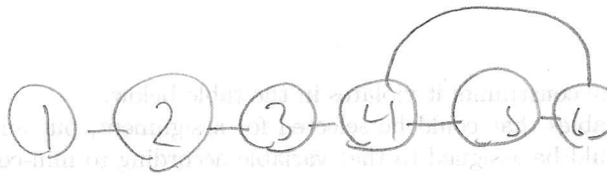
Put your answer to 1aii here:

$Q_4 = Q_6$

$Q_i \neq Q_{i+1}$ for i in range $(1-6)$

(b) [10 pts] Draw the constraint graph associated with your CSP.

Put your answer(s) to 1b here:



Q2. [35 pts] Pacman's new house

After years of struggling through mazes, Pacman has finally made peace with the ghosts, Blinky, Pinky, Inky, and Clyde, and invited them to live with him and Ms. Pacman. The move has forced Pacman to change the rooming assignments in his house, which has 6 rooms. He has decided to figure out the new assignments with a CSP in which the variables are Pacman (**P**), Ms. Pacman (**M**), Blinky (**B**), Pinky (**K**), Inky (**I**), and Clyde (**C**), the values are which room they will stay in, from 1-6, and the constraints are:

i) No two agents can stay in the same room
ii) **P** > 3 ✓
iii) **K** is less than **P**
iv) **M** is either 5 or 6 ✓
v) **P** $>$ **M**
vi) **B** is even ✓
vii) **I** is not 1 or 6 ✓
viii) $|$**I-C**$| = 1$
ix) $|$**P-B**$| = 2$

(a) [10 pts] **Unary constraints** On the grid below cross out the values from each domain that are eliminated by enforcing unary constraints.

P	~~1~~	~~2~~	~~3~~	4	5	6
B	~~1~~	2	~~3~~	4	~~5~~	6
C	1	2	3	4	5	6
K	1	2	3	4	5	6
I	~~1~~	2	3	4	5	~~6~~
M	~~1~~	~~2~~	~~3~~	~~4~~	5	6

(b) [3 pts] **MRV** According to the Minimum Remaining Value (MRV) heuristic, which variable should be assigned first?

○ P ○ B ○ C ○ K ○ I [✓ M]

(c) [10 pts] **Forward Checking** For the purposes of decoupling this problem from your solution to the previous problem, assume we choose to assign P first, and assign it the value 6. What are the resulting domains after enforcing unary constraints (from part i) and running forward checking for this assignment?

P						6
B	~~1~~	2	~~3~~	4	~~5~~	~~6~~
C	1	2	3	4	5	~~6~~
K	1	2	3	4	5	~~6~~
I	~~1~~	2	3	4	5	~~6~~
M	~~1~~	~~2~~	~~3~~	~~4~~	~~5~~	6

(d) [12 pts] **Iterative Improvement** Instead of running backtracking search, you decide to start over and run iterative improvement with the min-conflicts heuristic for value selection. Starting with the following assignment:

P:6, B:4, C:3, K:2, I:1, M:5

First, for each variable write down how many constraints it violates in the table below.
Then, in the table on the right, for all variables that could be selected for assignment, put an x in any box that corresponds to a possible value that could be assigned to that variable according to min-conflicts. When marking next values a variable could take on, only mark values different from the current one.

Variable	# violated
P	0
B	0
C	1
K	0
I	2
M	0

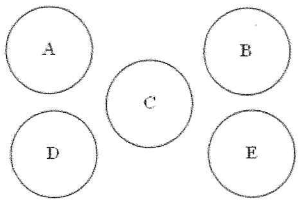
	1	2	3	4	5	6
P						
B						
C	X	X		X		
K						
I		X	X	X	X	
M						

Q3. [40 pts] All Satisfying Assignments

Consider a modified CSP in which we wish to find every possible satisfying assignment, rather than just one such assignment as in normal CSPs. In order to solve this new problem, consider a new algorithm which is the same as the normal backtracking search algorithm, except that when it sees a solution, instead of returning it, the solution gets added to a list, and the algorithm backtracks. Once there are no variables remaining to backtrack on, the algorithm returns the list of solutions it has found.

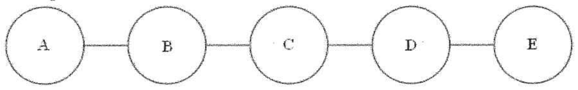
For each graph below, select whether or not using the MRV and/or LCV heuristics could affect the number of nodes expanded in the search tree in this new situation.

(a) [8 pts]



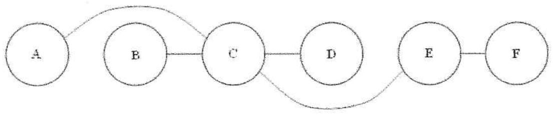
- [✓ Neither MRV nor LCV can have an effect.]
- ○ Only MRV can have an effect.
- ○ Only LCV can have an effect .
- ○ Both MRV and LCV can have an effect.

(b) [8 pts]



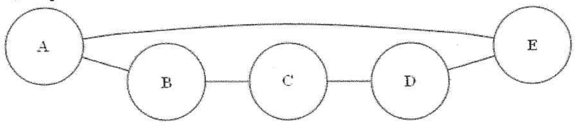
- ○ Neither MRV nor LCV can have an effect.
- ○ Only MRV can have an effect.
- [✓ Only LCV can have an effect .]
- ○ Both MRV and LCV can have an effect.

(c) [8 pts]



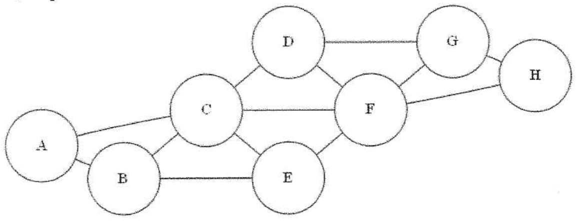
- ○ Neither MRV nor LCV can have an effect.
- ○ Only MRV can have an effect.
- ○ Only LCV can have an effect .
- [✓ Both MRV and LCV can have an effect.]

(d) [8 pts]



- ○ Neither MRV nor LCV can have an effect.
- [✓ Only MRV can have an effect.]
- ○ Only LCV can have an effect .
- ○ Both MRV and LCV can have an effect.

(e) [8 pts]



- ○ Neither MRV nor LCV can have an effect.
- ○ Only MRV can have an effect.
- ○ Only LCV can have an effect .
- [✓ Both MRV and LCV can have an effect.]