

---

CS 188Introduction to  
Spring 2014Artificial IntelligenceWritten HW2

---

## INSTRUCTIONS

- **Due:** Monday, February 10th, 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:** Submit the answer sheet pdf containing your answers. Page 1 must be this page, with your name, SID, and pandagrader email filled in. Page 2 must contain your answers to question 1 part a. Page 3 must contain your answers to question 1 part b. Page 4 must contain your answers to question 1 part c. You should 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 (check edx/piazza for latex templating files and instructions). **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](http://www.pandagrader.com). Log in and click on the class CS188 Spring 2014. Click on the submission titled Written HW 2 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	Chen
First Name	Jianzhong
SID	23478230
Email	chenjianzhong@berkeley.edu
Collaborators	None

For staff use only

Q. 1 Total
/30

## 1. (30 points) CSPs

## (a) 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)  $|P - B| = 2$

- (i) [1 pt] **Unary constraints** On the grid below cross out the values from each domain that are eliminated by enforcing unary constraints.

P	<del>1</del>	2	<del>3</del>	4	5	6
B	<del>1</del>	2	<del>3</del>	4	<del>5</del>	6
C	1	2	3	4	5	6
K	1	2	3	4	5	6
I	<del>1</del>	2	3	4	5	<del>6</del>
M	<del>1</del>	2	<del>3</del>	<del>4</del>	5	6

- (ii) [1 pt] **MRV** According to the Minimum Remaining Value (MRV) heuristic, which variable should be assigned to first?

☐ P                      ☐ B                      ☐ C                      ☐ K                      ☐ I                      ☒ M

- (iii) [3 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	<del>1</del>	2	<del>3</del>	4	<del>5</del>	<del>6</del>
C	1	2	3	4	5	<del>6</del>
K	1	2	3	4	5	<del>6</del>
I	<del>1</del>	2	3	4	5	<del>6</del>
M	<del>1</del>	2	<del>3</del>	<del>4</del>	5	<del>6</del>

- (iv) [3 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 **in the next iteration**. 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				
K						
I		x	x			
M						

**(b) Variable ordering**

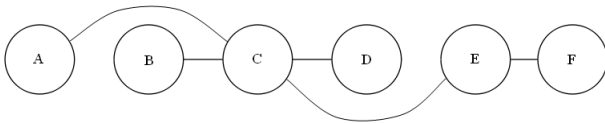
We say that a variable  $X$  is backtracked if, after a value has been assigned to  $X$ , the recursion returns at  $X$  without a solution, and a different value must be assigned to  $X$ .

For this problem, consider the following three algorithms:

- (a) Run backtracking search with no filtering
- (b) Initially enforce arc consistency, then run backtracking search with no filtering
- (c) Initially enforce arc consistency, then run backtracking search while enforcing arc consistency after each assignment

(i) [6 pts]

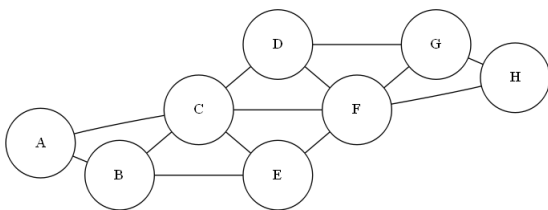
For each algorithm, circle all orderings of variable assignments that guarantee that no backtracking will be necessary when finding a solution to the CSP represented by the following constraint graph.



Algorithm 1	Algorithm 2	Algorithm 3
A-B-C-D-E-F	A-B-C-D-E-F	<span style="border: 1px solid red; padding: 2px;">A-B-C-D-E-F</span>
F-E-D-C-B-A	F-E-D-C-B-A	<span style="border: 1px solid red; padding: 2px;">F-E-D-C-B-A</span>
C-A-B-D-E-F	<span style="border: 1px solid red; padding: 2px;">C-A-B-D-E-F</span>	<span style="border: 1px solid red; padding: 2px;">C-A-B-D-E-F</span>
B-D-A-F-E-C	B-D-A-F-E-C	<span style="border: 1px solid red; padding: 2px;">B-D-A-F-E-C</span>
D-E-F-C-B-A	D-E-F-C-B-A	<span style="border: 1px solid red; padding: 2px;">D-E-F-C-B-A</span>
B-C-D-A-E-F	<span style="border: 1px solid red; padding: 2px;">B-C-D-A-E-F</span>	<span style="border: 1px solid red; padding: 2px;">B-C-D-A-E-F</span>

(ii) [6 pts]

For each algorithm, circle all orderings of variable assignments that guarantee that no more than two variables will be backtracked when finding a solution to the CSP represented by the following constraint graph.

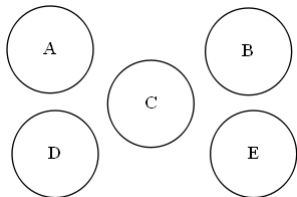


Algorithm 1	Algorithm 2	Algorithm 3
C-F-A-B-E-D-G-H	C-F-A-B-E-D-G-H	<span style="border: 1px solid red; padding: 2px;">C-F-A-B-E-D-G-H</span>
F-C-A-H-E-B-D-G	F-C-A-H-E-B-D-G	<span style="border: 1px solid red; padding: 2px;">F-C-A-H-E-B-D-G</span>
A-B-C-E-D-F-G-H	A-B-C-E-D-F-G-H	A-B-C-E-D-F-G-H
G-C-H-F-B-D-E-A	G-C-H-F-B-D-E-A	<span style="border: 1px solid red; padding: 2px;">G-C-H-F-B-D-E-A</span>
A-B-E-D-G-H-C-F	A-B-E-D-G-H-C-F	A-B-E-D-G-H-C-F
A-D-B-G-E-H-C-F	A-D-B-G-E-H-C-F	A-D-B-G-E-H-C-F

- (c) **All Satisfying Assignments** Now 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.

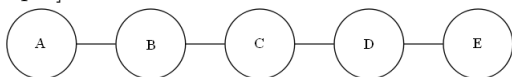
For each graph below, select whether or not using the MRV and/or LCV heuristics could affect the number of leaf nodes in the search tree in this new situation.

(i) [2 pts]



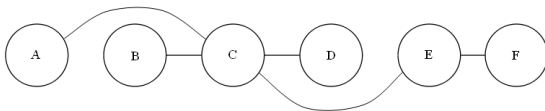
- ☒ Neither MRV nor LCV can have an effect.
- ☐ Only MRV can have an effect.
- ☐ Only LCV can have an effect .
- ☐ Both MRV and LCV can have an effect.

(ii) [2 pts]



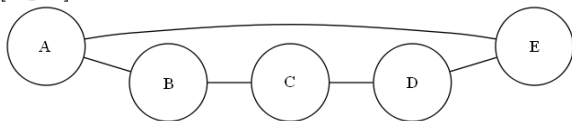
- ☐ Neither MRV nor LCV can have an effect.
- ☒ Only MRV can have an effect.
- ☐ Only LCV can have an effect .
- ☐ Both MRV and LCV can have an effect.

(iii) [2 pts]



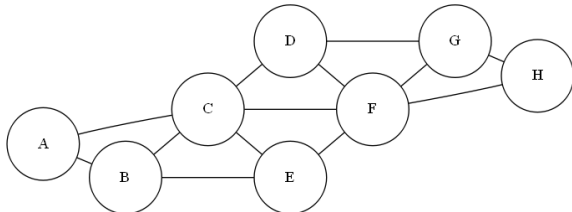
- ☐ Neither MRV nor LCV can have an effect.
- ☒ Only MRV can have an effect.
- ☐ Only LCV can have an effect .
- ☐ Both MRV and LCV can have an effect.

(iv) [2 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.

(v) [?? 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.