edX

Q1: CSPs: Final Exam Staff Assignments
Problem 1: CSPs: Final Exam Staff Assignments

The CS188x Final is coming up, and the CS188x 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. MDPs
- q5. True/False
- q6. Short Answer

There are 7 people on the course staff: Brad, Donahue, Ferguson, Judy, Kyle, Michael, and Nick. Each of them is responsible to work with Prof. Abbeel on one question. (But a question could end up having more than one staff person, or potentially zero staff assigned to it.) However, the staff are pretty quirky and want the following constraints to be satisfied:

i. Donahue (D) will not work on a question together with Judy (J).

ii. Kyle (K) must work on either Search, Games or CSPs

iii. Michael (M) is very odd, so he can only contribute to an odd-numbered question.

iv. Nick (N) must work on a question that's before Michael (M)'s question.

v. Kyle (K) must work on a question that's before Donahue (D)'s question

vi. Brad (B) does not like grading exams, so he must work on True/False.

vii. Judy (J) must work on a question that's after Nick (N)'s question.

viii. If Brad (B) is to work with someone, it cannot be with Nick (N).

ix. Nick (N) cannot work on question 6.

x. Ferguson (F) cannot work on questions 4, 5, or 6

xi. Donahue (D) cannot work on question 5.

xii. Donahue (D) must work on a question before Ferguson (F)'s question.

<div align="center">

Keep Constraints Always Visible

</div>

Clarification on the word "before": q1 and q2 are before q3, but q3, q4, q5 and q6 are not before q3.

We recommend you work out the solutions to the following questions on a sheet of scratch paper, and then enter your results below.

Part 1

3/3 points (ungraded)
We will model this problem as a constraint satisfaction problem (CSP). Our variables correspond to each of the staff members, B, D, F, J, K, M, N, and the domains are the questions 1, 2, 3, 4, 5, 6. After applying the unary constraints, what are the resulting domains of each variable?

B	**D**	**F**	**J**
☐ q1	☑ q1	☑ q1	☑ q1
☐ q2	☑ q2	☑ q2	☑ q2

☐ q3	☑ q3	☑ q3	☑ q3
☐ q4	☑ q4	☐ q4	☑ q4
☑ q5	☐ q5	☐ q5	☑ q5
☐ q6	☑ q6	☐ q6	☑ q6
✔	✔	✔	✔

K   **M**   **N**

☑ q1	☑ q1	☑ q1
☑ q2	☐ q2	☑ q2
☑ q3	☑ q3	☑ q3
☐ q4	☐ q4	☑ q4
☐ q5	☑ q5	☑ q5
☐ q6	☐ q6	☐ q6
✔	✔	✔

Submit

✔ Correct (3/3 points)

Part 2

3/3 points (ungraded)

If we apply the Minimum Remaining Value (MRV) heuristic, which variable should be assigned first?

◉ Brad ✔

○ Donahue

○ Ferguson

○ Judy

○ Kyle

○ Michael

○ Nick

Submit

✔ Correct (3/3 points)

Part 3

4/4 points (ungraded)
Normally we would now proceed with the variable you found in (b), but to decouple this question from the previous one (and prevent potential errors from propagating), let's proceed with assigning Michael first. For value ordering we use the Least Constraining Value (LCV) heuristic, where we use **Forward Checking** to compute the number of remaining values in other variables domains. What ordering of values is prescribed by the LCV heuristic?

If during forward checking for a certain value a variable's domain becomes empty, then that value is particularly constraining (per resulting in small resulting domains) but doesn't need to be treated in a special way --- i.e., still simply count sizes of resulting domains even if some domains are empty.

You will want to work out on a piece of scratch paper what the resulting domains are after assigning each value

Note: There may be fewer than 6 values remaining for Michael, in which case, you should leave any remaining boxes blank.

When entering the values into the box, you can either enter the number or q followed by the number. For example, to enter q1. Search --- simply enter into the box "1" or "q1" (but *without* the quotes).

Michael's first value using the LCV heuristic is:

> 5 ✔

Michael's second value using the LCV heuristic is:

> 3 ✔

Michael's third value using the LCV heuristic is:

> 1 ✔

Michael's fourth value using the LCV heuristic is:

> ✔

Michael's fifth value using the LCV heuristic is:

> ✔

Michael's sixth value using the LCV heuristic is:

> ✔

Submit

✔ Correct (4/4 points)

Part 4

6/6 points (ungraded)
Realizing this is a tree-structured CSP, we decide not to run backtracking search, and instead use the efficient two-pass algorithm to solve tree-structured CSPs. We will run this two-pass algorithm **after** applying the unary constraints from part (a). Below is the linearized version of the tree-structured CSP graph for you to work with.

K → D → F → J → N → B → M
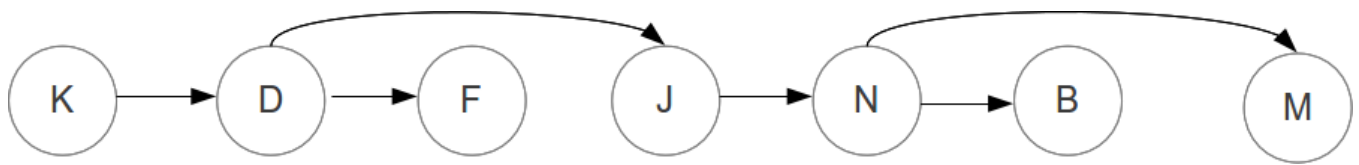
Pass from **right to left** to perform Domain Pruning. Select the values that remain in the domains below.

B
- ☐ q1
- ☐ q2
- ☐ q3
- ☐ q4
- ☑ q5
- ☐ q6

✔

D
- ☑ q1
- ☑ q2
- ☐ q3
- ☐ q4
- ☐ q5
- ☐ q6

✔

F
- ☑ q1
- ☑ q2
- ☑ q3
- ☐ q4
- ☐ q5
- ☐ q6

✔

J
- ☐ q1
- ☑ q2
- ☑ q3
- ☑ q4
- ☑ q5
- ☑ q6

✔

K
- ☑ q1
- ☐ q2
- ☐ q3
- ☐ q4
- ☐ q5
- ☐ q6

✔

M
- ☑ q1
- ☐ q2
- ☑ q3
- ☐ q4
- ☑ q5
- ☐ q6

✔

N
- ☑ q1
- ☑ q2
- ☑ q3
- ☑ q4
- ☐ q5
- ☐ q6

✔

Submit

✔ Correct (6/6 points)

Part 5

5/5 points (ungraded)
Pass from **left to right**, assigning values for the solution. If there is more than one possible assignment, choose the highest value, meaning choose q6 over q5, q5 over q4, q4 over q3, q3 over q2, and q2 over q1.

When entering the values into the box, you can either enter the number or q followed by the number. For example, to enter q1. Search --- simply enter into the box "1" or "q1" (but *without* the quotes).

Kyle's value is:

| 1 | ✔ |

Donahue's value is:

| 2 | ✔ |

Ferguson's value is:

| 3 | ✔ |

Judy's value is:

| 6 | ✔ |

Nick's value is:

| 4 | ✔ |

Brad's value is:

| 5 | ✔ |

Michael's value is:

| 5 | ✔ |

Submit

✔ Correct (5/5 points)