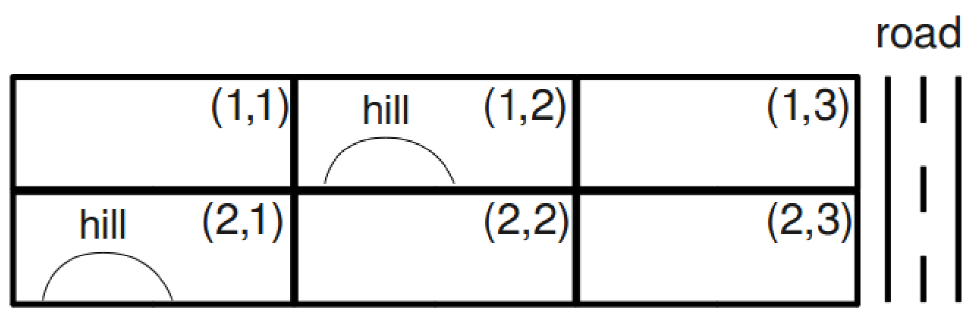
Q1 Campus Layout

44 Points

You are asked to determine the layout of a new, small college. The campus will have four structures: an administration structure (A), a bus stop (B), a classroom (C), and a dormitory (D). Each structure (including the bus stop) must be placed somewhere on the grid shown below.



The layout must satisfy the following constraints:

- i. The bus stop (B) must be adjacent to the road.
- ii. The administration structure (A) and the classroom (C) must both be adjacent to the bus stop (B).
- iii. The classroom (C) must be adjacent to the dormitory (D).
- iv. The administration structure (A) must not be adjacent to the dormitory (D).
- v. The administration structure (A) must not be on a hill.
- vi. The dormitory (D) must be on a hill or adjacent to the road.
- vii. All structures must be in different grid squares.

Here, adjacent means that the structures must share a grid edge, not just a corner.

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

Q1.1

4 Points

Which of the constraints above are unary constraints?

- [x] i
- [ ] ii
- [ ] iii
- [ ] iv
- [x] v
- [x] vi
- [ ] vii
- [ ] None of the above

Q1.2
4 Points

Select the domains of all variables after unary constraints have been applied.

- A

- [x] (1,1)
- [ ] (1,2)
- [x] (1,3)
- [ ] (2,1)
- [x] (2,2)
- [x] (2,3)

- B

- [ ] (1,1)
- [ ] (1,2)
- [x] (1,3)
- [ ] (2,1)
- [ ] (2,2)
- [x] (2,3)

- C

- [x] (1,1)
- [x] (1,2)
- [x] (1,3)
- [x] (2,1)
- [x] (2,2)
- [x] (2,3)

- D

- [ ] (1,1)
- [x] (1,2)
- [x] (1,3)
- [x] (2,1)
- [ ] (2,2)
- [x] (2,3)

Q1.3
4 Points

Let's start from the table above (the answer to Part 2) and enforce arc consistency. Initally, the queue contains all arcs (in alphabetical order).

Let's examine what happens when enforcing A $\rightarrow$ B. After enforcing unary constraints, the domains of A and B are:

A	B
(1,1)	
(1,3)	(1,3)
(2,2)	
(2,3)	(2,3)

Which of the following contains the correct domains after enforcing A $\rightarrow$ B?
Pay attention to which variable's domain changes and which

side of the arc
it's on.

A	B
(1,1)	
	(1,2)
(1,3)	(1,3)
(2,2)	
(2,3)	(2,3)

i

A	B
(1,3)	(1,3)
(2,2)	
(2,3)	(2,3)

ii

A	B
(1,1)	
(1,3)	
(2,2)	
(2,3)	(2,3)

iii

A	B
(1,1)	
(1,3)	(1,3)
(2,2)	
(2,3)	

iv

i

ii

iii

iv

Q1.4
4 Points

Starting from the answer to Part 2 (in which unary constraints are enforced), select the domains of all variables after A $\rightarrow$ B is enforced.

- A

- [ ] (1,1)
- [ ] (1,2)
- [x] (1,3)
- [ ] (2,1)
- [x] (2,2)
- [x] (2,3)

- B

- [ ] (1,1)
- [ ] (1,2)
- [x] (1,3)
- [ ] (2,1)
- [ ] (2,2)
- [x] (2,3)

- C

- [x] (1,1)
- [x] (1,2)
- [x] (1,3)
- [x] (2,1)
- [x] (2,2)
- [x] (2,3)

- D

- [ ] (1,1)
- [x] (1,2)
- [x] (1,3)
- [x] (2,1)
- [ ] (2,2)
- [x] (2,3)

Q1.5
4 Points

You should verify that enforcing consistency for A $\longrightarrow$ C, A $\longrightarrow$ D, B $\longrightarrow$ A, B $\longrightarrow$ C, B $\longrightarrow$ D, and C $\longrightarrow$ A do not change the domains of any variables. After enforcing these arcs, the next is C $\longrightarrow$ B.

Continuing from the previous parts, select the domains of all variables after C $\longrightarrow$ B is enforced.

- A

- [ ] (1,1)
- [ ] (1,2)
- [x] (1,3)
- [ ] (2,1)
- [x] (2,2)
- [x] (2,3)

- B

- [ ] (1,1)
- [ ] (1,2)
- [x] (1,3)
- [ ] (2,1)
- [ ] (2,2)
- [x] (2,3)

- C

- [ ] (1,1)
- [x] (1,2)
- [x] (1,3)
- [ ] (2,1)
- [x] (2,2)
- [x] (2,3)

- D

- [ ] (1,1)
- [x] (1,2)
- [x] (1,3)
- [x] (2,1)
- [ ] (2,2)
- [x] (2,3)

Q1.6
4 Points

What arcs got added to the queue while enforcing C $\longrightarrow$ B? Remember that the queue

contained C $\longrightarrow$ D, D $\longrightarrow$ A, D $\longrightarrow$ B, and D $\longrightarrow$ C prior to enforcing C $\longrightarrow$ B.

- [ ] A $\rightarrow$ B
- [x] A $\rightarrow$ C
- [ ] A $\rightarrow$ D
- [ ] B $\rightarrow$ A
- [x] B $\rightarrow$ C
- [ ] B $\rightarrow$ D
- [ ] C $\rightarrow$ A
- [ ] C $\rightarrow$ B
- [ ] C $\rightarrow$ D
- [ ] D $\rightarrow$ A
- [ ] D $\rightarrow$ B
- [ ] D $\rightarrow$ C
- [ ] None

Q1.7
4 Points

Continuing from the previous parts, select the domains of all variables after enforcing arc consistency until the queue is empty. Remember that the queue currently contains C $\rightarrow$ D, D $\rightarrow$ A, D $\rightarrow$ B, D $\rightarrow$ C, and any arcs that were added while enforcing C $\rightarrow$ B.

- A

- [ ] (1,1)
- [ ] (1,2)
- [x] (1,3)
- [ ] (2,1)
- [x] (2,2)
- [x] (2,3)

- B

- [ ] (1,1)
- [ ] (1,2)
- [x] (1,3)
- [ ] (2,1)
- [ ] (2,2)
- [x] (2,3)

- C

- [ ] (1,1)
- [x] (1,2)
- [x] (1,3)
- [ ] (2,1)
- [x] (2,2)
- [x] (2,3)

- D

- [ ] (1,1)
- [x] (1,2)
- [x] (1,3)
- [x] (2,1)
- [ ] (2,2)
- [ ] (2,3)

Q1.8
4 Points

If arc consistency had resulted in all domains having a single value left, we would have already found a solution. Similarly, if it had found that any domain had no values left, we would have already found that no solution exists. Unfortunately, this is not the case in our example (as you should have found in the previous part). To solve the problem, we need to start searching. Use the MRV (minimum remaining values) heuristic to choose which variable gets assigned next (breaking any ties alphabetically).

Which variable gets assigned next?

- A
- B
- C
- D

Q1.9
4 Points

The variable you selected should have two values left in its domain. We will use the least-constraining value (LCV) heuristic to decide which value to assign before continuing with the search. To choose which value is the least-constraining value, enforce arc consistency for each value (on a scratch piece of paper). For each value, count the total number of values remaining over all variables.

Which value has the largest number of values remaining (and therefore is the least constraining value)?

- (1,1)
- (1,2)
- (1,3)
- (2,1)
- (2,2)
- (2,3)

Q1.10
4 Points

After assigning a variable, backtracking search with arc consistency enforces arc consistency before proceeding to the next variable.

Select the domains of all variables after assignment of the least-constraining value to the variable you selected and enforcing arc consistency. Note that you already did this computation to determine which value was the LCV.

- A
  - [ ] (1,1)
  - [ ] (1,2)
  - [x] (1,3)
  - [ ] (2,1)
  - [ ] (2,2)
  - [ ] (2,3)

- B
  - [ ] (1,1)
  - [ ] (1,2)
  - [ ] (1,3)
  - [ ] (2,1)
  - [ ] (2,2)
  - [x] (2,3)

- C

- [ ] (1,1)
- [ ] (1,2)
- [ ] (1,3)
- [ ] (2,1)
- [x] (2,2)
- [ ] (2,3)

- D

- [ ] (1,1)
- [ ] (1,2)
- [ ] (1,3)
- [x] (2,1)
- [ ] (2,2)
- [ ] (2,3)

Q1.11
4 Points

Is the answer to the previous part a solution to the CSP?

Yes

No

Q2 CSP Properties
10 Points

Q2.1
5 Points

Select all of the following statements about CSPs that are true.

- [x] Even when using arc consistency, backtracking might be needed to solve a CSP.
- [x] Even when using forward checking, backtracking might be needed to solve a CSP.
- [ ] None of the above

Q2.2
5 Points

Select all of the following statements about CSPs that are true.

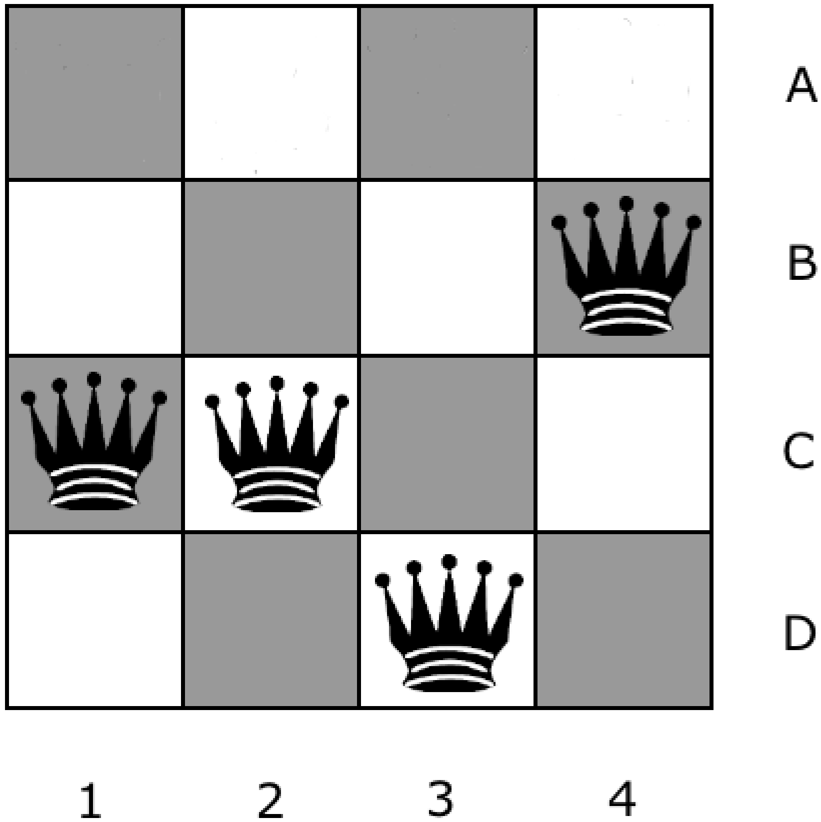
- [ ] When using backtracking search with the same rules to select unassigned variables and to order value assignments (in our case, usually Minimum Remaining Values and Least-Constraining Value, with alphabetical tiebreaking), arc consistency will always give the same solution as forward checking, if the CSP has a solution.
- [x] For a CSP with binary constraints that has no solution, some initial values may still pass arc consistency before any variable is assigned.
- [ ] None of the above

Q3 4-Queens

14 Points

The min-conflicts algorithm attempts to solve CSPs iteratively. It starts
by assigning some value to each of the variables, ignoring the constraints
when doing so. Then, while at least one constraint is violated, it repeats
the following: (1) randomly choose a variable that is currenly violating a
constraint, (2) assign to it the value in its domain such that after the
assignment the total number of constraints violated is minimized (among all
possible selections of values in its domain).

In this question, you are asked to execute the min-conflicts algorithm on
a simple problem: the 4-queens problem in the figure shown below. Each
queen is dedicated to its own column (i.e. we have variables Q_1, Q_2, Q_3,
and Q_4 and the domain for each one of them is $\{A, B, C, D\}$). In the
configuration shown below, we have $Q_1 = C$, $Q_2 = C$, $Q_3 = D$,
$Q_4 = B$. Two queens are in conflict if they share the same row,
diagonal, or column (though in this setting, they can never share the same
column).



You will execute min-conflicts for this problem three times, starting with the state shown in the figure above. When selecting a variable to reassign, min-conflicts chooses a conflicted variable at random. For this problem, assume that your random number generator always chooses the leftmost conflicted queen. When moving a queen, move it to the square in its column that leads to the fewest conflicts with other queens. If there are ties, choose the topmost square among them.

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

Q3.1
4 Points

Starting with the queens in the configuration shown in the above figure, which queen will be moved, and where will it be moved to?

Queen

1

2

3

4

Position

A

B

C

D

Q3.2
5 Points

Continuing off of Part 1, which queen will be moved, and where will it be moved to?

Queen

1

2

3

4

Position

A

B

C

D

Q3.3
5 Points

Continuing off of Part 2, which queen will be moved, and where will it be moved to?

Queen

1

2

3

4

Position

A

B

C

D

Q4 Arc Consistency

14 Points

Consider the problem of arranging the schedule for an event. There
are three time slots: 1, 2, and 3. There are three presenters: A
,
B, and C. The variables for the CSP will then be A, B,
and C, each with domain $\{1, 2, 3\}$. The following constraints need to
be satisfied:

- A, B, and C all need to take on different values
- $A < C$

Q4.1

4 Points

Enforce consistency for the arc $A \to C$, and then select which values
remain for each variable.

- [x] $A : 1$
- [x] $A : 2$
- [ ] $A : 3$
- [x] $B : 1$
- [x] $B : 2$
- [x] $B : 3$
- [x] $C : 1$
- [x] $C : 2$
- [x] $C : 3$

Q4.2
5 Points

Starting from the result of the previous step, enforce consistency for the
arc $B \to A$, and then select which values remain for each variable.

- [x] $A : 1$
- [x] $A : 2$
- [ ] $A : 3$
- [x] $B : 1$
- [x] $B : 2$
- [x] $B : 3$
- [x] $C : 1$
- [x] $C : 2$
- [x] $C : 3$

Q4.3
5 Points

Starting from the result of the previous step, enforce consistency for the
arc $C \rightarrow A$, and then select which values remain for each variable.

- [x] $A : 1$
- [x] $A : 2$
- [ ] $A : 3$
- [x] $B : 1$
- [x] $B : 2$
- [x] $B : 3$
- [ ] $C : 1$
- [x] $C : 2$
- [x] $C : 3$

Q5 Arc Consistency Properties

6 Points

Assume you are given a CSP and you enforce arc consistency. Which of the following are true?

- [ ] If the CSP has no solution, it is guaranteed that enforcement of arc consistency resulted in at least one domain being empty.
- [ ] If the CSP has a solution, then after enforcing arc consistency, you can directly read off the solution from resulting domains.
- [x] In general, to determine whether the CSP has a solution, enforcing arc consistency alone is not sufficient; backtracking may be required.
- [ ] None of the above.

Q6 Backtracking Arc Consistency
12 Points

We are given a CSP with only binary constraints. Assume we run backtracking search with arc consistency as follows. Initially, when presented with the CSP, one round of arc consistency is enforced. This first round of arc consistency will typically result in variables having pruned domains. Then we start a backtracking search using the pruned domains. In this backtracking search we use filtering through enforcing arc consistency after every assignment in the search.

Q6.1
4 Points

Which of the following are true about this algorithm?

- [ ] If after a run of arc consistency during the backtracking search we end up with the filtered domains of **all** of the not yet assigned variables being empty, this means the CSP has no solution.
- [ ] If after a run of arc consistency during the backtracking search we end up with the filtered domain of **one** of the not yet assigned variables being empty, this means the CSP has no solution.
- [x] None of the above.

Q6.2
4 Points

Which of the following are true about this algorithm?

- [x] If after a run of arc consistency during the backtracking search we end up with the filtered domains of **all** of the not yet assigned variables being empty, this means the search should backtrack because this particular branch in the search tree has no solution.
- [x] If after a run of arc consistency during the backtracking search we end up with the filtered domain of **one** of the not yet assigned variables being empty, this means the search should backtrack because this particular branch in the search tree has no solution.
- [ ] None of the above.

Q6.3
4 Points

Which of the following are true about this algorithm?

- [x] If after a run of arc consistency during the backtracking search we end up with the filtered domains of all of the not yet assigned variables each having exactly one value left, this means we have found a solution.

- [ ] If after a run of arc consistency during the backtracking search we end up with the filtered domains of all of the not yet assigned variables each having more than one value left, this means we have found a whole space of solutions and we can just pick any combination of values still left in the domains and that will be a solution.

- [x] If after a run of arc consistency during the backtracking search we end up with the filtered domains of all of the not yet assigned variables each having more than one value left, this means we can't know yet whether there is a solution somewhere further down this branch of the tree, and search has to continue down this branch to determine this.

HW 2 (Electronic Component)

Graded

Student

ریحانه شاهرخیان

Total Points

100 / 100 pts

Question 1

Campus Layout **44** / 44 pts

1.1 (no title) **4** / 4 pts

1.2	(no title)	**4** / 4 pts
1.3	(no title)	**4** / 4 pts
1.4	(no title)	**4** / 4 pts
1.5	(no title)	**4** / 4 pts
1.6	(no title)	**4** / 4 pts
1.7	(no title)	**4** / 4 pts
1.8	(no title)	**4** / 4 pts
1.9	(no title)	**4** / 4 pts
1.10	(no title)	**4** / 4 pts
1.11	(no title)	**4** / 4 pts

Question 2

CSP Properties **10** / 10 pts

2.1	(no title)	**5** / 5 pts
2.2	(no title)	**5** / 5 pts

Question 3

4-Queens **14** / 14 pts

3.1	(no title)	**4** / 4 pts
3.2	(no title)	**5** / 5 pts
3.3	(no title)	**5** / 5 pts

Question 4

Arc Consistency **14** / 14 pts

4.1	(no title)	**4** / 4 pts
4.2	(no title)	**5** / 5 pts
4.3	(no title)	**5** / 5 pts

Question 5

Arc Consistency Properties **6** / 6 pts

Question 6

Backtracking Arc Consistency **12** / 12 pts

6.1	(no title)	**4** / 4 pts
6.2	(no title)	**4** / 4 pts

6.3 (no title) **4** / 4 pts