This handout includes space for every question that requires a written response. Please feel free to use it to handwrite your solutions (legibly, please). If you choose to typeset your solutions, the README.md for this assignment includes instructions to regenerate this handout with your typeset LaTeX solutions.

0.a

(a) **[3 points (Written)]** Let's create a CSP. Suppose you have n light bulbs, where each light bulb $i = 1, \ldots, n$ is initially off. You also have m buttons which control the lights. For each button $j = 1, \ldots, m$, we know the subset $T_j \subseteq \{1, \ldots, n\}$ of light bulbs that it controls. When button j is pressed, it toggles the state of each light bulb in T_j (For example, if $3 \in T_j$ and light bulb 3 is off, then after the button is pressed, light bulb 3 will be on, and vice versa).

Your goal is to turn on all the light bulbs by pressing a subset of the buttons. Construct a CSP to solve this problem. Your CSP should have m variables and n constraints. *For this problem only*, you can use constraints with any arity. Describe your CSP precisely and concisely. You need to specify the variables with their domain, and the constraints with their scope and expression. Make sure to include T_j in your answer.

Variables:

There are m binary variables, one for each button

j (where 1 <= j <= m).

Each variable B_j represents the state of button j:

Domain of variables:

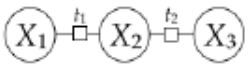
Each variable B_j will have a binary domain, that is, $B_J \in \{0,1\}$. Where $B_j = 1$ means that button j is pressed, and $B_j = 0$ means that it is not pressed.

Constraint:

$$\sum_{j \in 1,\ldots,m} B_j \cdot [bulb_i \in T_j]$$

0.b

[3 points (Written)] Let's consider a simple CSP with 3 variables and 2 binary factors:



where $X_1, X_2, X_3 \in \{0, 1\}$ and t_1, t_2 are XOR functions (that is $t_1(X) = x_1 \bigoplus x_2$ and $t_2(X) = x_2 \bigoplus x_3$).

i. How many consistent assignments are there for this CSP?

ii. To see why variable ordering is important, let's use backtracking search to solve the CSP *without using any heuristics (MCV, LCV, AC-3) or lookahead.* How many times will `backtrack()` be called to get all consistent assignments if we use the fixed ordering X_1, X_3, X_2? Draw the call stack for `backtrack()`.

(You should use the Backtrack algorithm from the slides. The initial arguments are $x = \emptyset$, $w = 1$, and the original Domain.) In the code, this will be `BacktrackingSearch.numOperations`.

iii. To see why lookahead can be useful, let's do it again with the ordering X_1, X_3, X_2 and AC-3. How many times will Backtrack be called to get all consistent assignments? Draw the call stack for `backtrack()`.

i) Since each variable can take two values (0 or 1), there are a total of 2^3 = 8 possible combinations of assignments.
Now, we need to check how many of these assignments satisfy both constraints.
Since T_1 it is an XOR function X_1 And X_2 there are only two combinations that satisfy.

X1 = 1; X2 = 0; X3 = 1

and

X1 = 0; X2 = 1; X3 = 0.

ii) backtrack(∅, 1, {X1, X3, X2})
backtrack({X1 = 0}, 1, {X3, X2})
backtrack({X1 = 0, X3 = 0}, 1, {X2})
backtrack({X1 = 0, X3 = 0, X2 = 0}, 0, ∅)
backtrack({X1 = 0, X3 = 0, X2 = 1}, 1, ∅)
backtrack({X1 = 0, X3 = 1}, 1, {X2})
backtrack({X1 = 1}, 1, {X3, X2})
backtrack({X1 = 1, X3 = 0}, 1, {X2})
backtrack({X1 = 1, X3 = 1}, 1, {X2})
backtrack({X1 = 1, X3 = 1, X2 = 0}, 0, ∅)

iii) backtrack(∅, 1, {X1, X3, X2})
backtrack({X1 = 0}, 1, {X3, X2})
backtrack({X1 = 0, X3 = 0}, 1, {X2})
backtrack({X1 = 0, X3 = 0, X2 = 0}, 0, ∅)
backtrack({X1 = 0, X3 = 0, X2 = 1}, 1, ∅)

backtrack({X1 = 0, X3 = 1}, 1, {X2})
backtrack({X1 = 1}, 1, {X3, X2})
backtrack({X1 = 1, X3 = 0}, 1, {X2})
backtrack({X1 = 1, X3 = 1}, 1, {X2})
backtrack({X1 = 1, X3 = 1, X2 = 0}, 0, Ø)

2.d

(d) **[2 points (Written)]** Now try to use the course scheduler for the winter and spring quarters (and next year if applicable). Create your own `profile.txt` and then run the course scheduler:

```
python run_p2.py profile.txt
```

You might want to turn on the appropriate heuristic flags to speed up the computation. Does it produce a reasonable course schedule? Please include your `profile.txt` and the best schedule in your writeup; we're curious how it worked out for you! Please include your schedule and the profile in the PDF, otherwise you will not receive credit.

My profile.txt

Unit limit per quarter. You can ignore this for the first

few questions in problem 2.

minUnits 1

maxUnits 10

These are the quarters that I need to fill out.

It is assumed that the quarters are sorted in chronological order.

register Win2024

register Spr2024

Courses I've already taken

taken CS107

taken CS103

taken CS221

taken CS106X

taken MATH51

taken MATH21

taken CS229

taken MATH42

taken CS109

taken CS140

taken CS145

taken CS124

taken CS106B

Courses that I'm requesting

request CS246

request CS205A

request CS341

request CS228

request CS231B after CS228

Here's the best schedule:

Quarter	Units	Course
Win2024	**4**	**CS228**
Spr2024	**3**	**CS205A**

3.a

3. **Residency Hours Scheduling**

(a) **[3 points (Written)]**

Many uses of constraint satisfaction in real-world scenarios involve assignment of resources to entities, like assigning packages to different trucks to optimize delivery. However, when the agents are people, the issue of fair division arises. In this question, you will consider the ethics of what constraints to remove in a CSP when the CSP is unsatisfiable.

Medical residents are often scheduled to work long shifts with insufficient rest, leading to exhaustion and burnout. This can negatively affect the residents and potentially lead to mistakes that also negatively affect the patients in their care [1]. A hospital could use a constraint-satisfaction approach to try to create a work schedule that respects the "on-call night, day-off, rest period, and total work-hour regulations mandated by the Accreditation Council for Graduate Medical Education, as well as the number of residents needed each hour given the demand (aka number of patients and procedures scheduled) [2]. The constraints are:

1 One day off every 7 days

2 Minimum 8 hour rest period between shifts

3 No more than 80 hours of work per week averaged over a 4 week period

4 At least 14 hours free of clinical work and education after 24 hours of in-house call

5 Number of residents needed each hour

Let's assume for a given hospital that the constraints listed above were collectively **unsatisfiable** given the number of residents assigned to that hospital. However, its formulation as an unsatisfiable CSP depends on other factors remaining fixed, such as

A The number of total residents

B The work to be performed by residents as opposed to other staff

C The budget available to hire residents or other staff

In this case, would you remove one of the numbered constraints 1-5 or advocate that the hospital administration change one of A-C to make the problem solveable? If so, explain which one and give a reason why.

What we expect: In 2-4 sentences, you must explicitly state which numbered constraint or lettered factor you would change and justify your choice with a reason that explains why you chose that one.

I propose to divide the work performed between residents and other employees (element B) to reduce the burden on residents while respecting the limits of working hours. This guarantees public health and patient safety. It is important to comply with restrictions 1 through 5 to protect public health and the quality of patient care. Changing the B factor reduces workload and allows regulatory requirements to be met without compromising the quality of life of residents or patients.