CPSC 322 **Artificial Intelligence I** **Fall 2020**

Assignment Three: CSPs

Due: 11:59pm, Monday 5 October 2020.

Solving following problems requires using the Python code at `http://aipython.org/aipython_322_as3.zip` and/or the AISpace 2 or AISpace 1 applets at `http://AIspace.org`. Note that this Python code extends the previous code to include code for CSPs. You will need to download the AIspace applets to use them.

This can be done in groups of size 1, 2 or 3. Working alone is not recommended. All members of the group need to be able to explain the group's answer.

Submit your answers using Canvas. Use proper sentences (not note form) in your answer.

Ask questions on Canvas discussion board. Feel free to answer them too.

Question One

CSP techniques are useful in solving complex configuration and allocation problems. You are given the task of allocating four developments in a new site in Whistler. You have to place a housing complex, a big hotel, a recreational area and a garbage dump. The area for development can be represented as 3x3 grid (three rows 0,1,2 and three columns 0,1,2) and you need to place each development in their own cell of the grid. Unfortunately there are some practical constraints on the problem that you need to take into account. In the following, A is close to B if A is in a cell that shares an edge or a corner with B.

- There is a cemetery in cell 0,0.
- There is a lake in cell 1,2.
- The garbage dump should not be close to the lake.
- The housing complex and the big hotel should not be close to the cemetery.
- The recreational area should be close to the lake.
- The housing complex and the big hotel should be close to the recreational area.
- The housing complex and the big hotel should not be close to the garbage dump.

Represent this problem as a CSP. Be as precise as you can in specifying the constraints. Also there may be some basic constraints that are in the description (and not just in the points). You need to give a human-level description of the variables, domains and constraints as well as either an AIspace or AIPython CSP representation and demonstrate that it works.

How many solutions are there? If there are solutions, what are they? If there are no solutions, explain why. The solutions and the explanations you give, must be in straightforward non-techical language that a member of the public could understand.

Question Two

Consider a scheduling problem, where there are eight variables A, B, C, D, E, F, G, H each with domain $\{1, 2, 3, 4\}$. Suppose the constraints are: $A > G$, $A \leq H$, $|F - B| = 1$, $G < H$, $|G - C| = 1$, $H - C$ is even, $H \neq D$, $D > G$, $D \neq C$, $E \neq C$, $E < D - 1$, $E \neq H - 2$, $G \neq F$, $H \neq F$, $C \neq F$, $D \neq F$, $|E - F|$ is odd.

(a) Show how search can be used to solve this problem, using the variable ordering A, B, C, D, E, F, G, H. To do this, write a program to print out all answers and count the number of nodes expanded and the number failing consistency checks.

You can use whatever programming language you like. We don't want a general program; just one for this example. Python code that produces the answer for variables X, Y and Z each with domains $\{t, f\}$, and constraints $X \neq Y$, $Y \neq Z$ is at `http://www.cs.ubc.ca/~poole/cs322/2020/as3/streeCount.py`

(b) Is there a smaller tree? Give a node ordering that results in as small a tree as you can find. Show both how many failing consistency checks there are, and how many nodes are expanded to find all solutions. Explain how you found this ordering and why you would expect the tree resulting from this ordering to be good. (A good explanation as to why your ordering is expected to be good is more important than the perfect ordering.)

(c) How many failures would there be for generate-and-test? Hint: it can be computed in a straightforward manner if you know the number of variables, the size of the domains, and the number of solutions; you do not need to write a program like the second part of `streeCount.py`.

The file `as3csp.py` in the zip file contains an AIPython representation of this domain. Note that AIPython does not tell us how many failing consistency checks there are.

Question Three

For each question, specify how long you spend on it, and what you learned. How was the work in the team allocated? Was the question reasonable? (This questions is worth marks, so please do it!)