Homework: 1 CS 205. Introduction to Artificial Intelligence

**STOP: If you have taken cs170 with me, go to Appendix A instead.**

**To begin**: Watch this video: Just watch the from 9:00 to 14:10 www.youtube.com/watch?v=5YBIrc-6G-0&t=539s

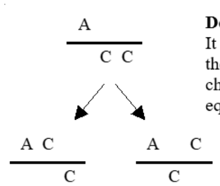
# Crossing the River

On the bank of a river are 1 adult, 2 children and a very small rowboat. We want to get all of the people across the river by rowing. The boat will hold:

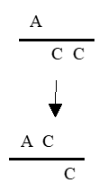
- 2 children or
- a single child or
- a single adult (it is too small to hold an adult and a child)

All of the people can row.

**I) Draw the search space**, you may use a computer drawing package (MS PowerPoint works reasonably well) or you may do it by hand (in which case I expect it to be extremely neat).
Note that you should take advantage of the symmetry to reduce the branching factor. There is really no difference between the two children and your tree should reflect this (to be clear look at the examples below and assume that the boat is with the children in the parent node).

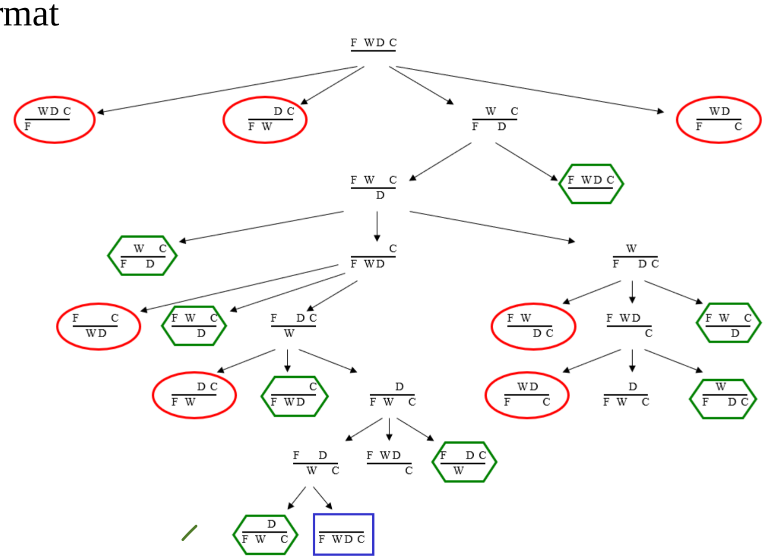



**Don't do this!**
It is wasteful, since the left and right children are logically equivalent.




**Do this.**

## Sample of desired format



**Extra credit:** Instead of doing "**1 adult, 2 children**" do the "**Missionaries and Cannibals**" problem instead.

## 2) Exhaustive search

16 777 216 → All combination.

16777 s = 4.6 hrs

Consider the 8-queens problem. Imagine that we are going to try to solve it using exhaustive search (we are simply going to enumerate every possible arrangement of the 8-queens on a standard chessboard and check each arrangement to see if it is a solution). Note that for this assignment we are going to be a little inefficient and generate all arrangements, including arrangements that are identical after rotation and reflection. For each question, be sure to show all work.

40,320 / 1000 = ~40 s    8! = 40,320 (all arrangements?)

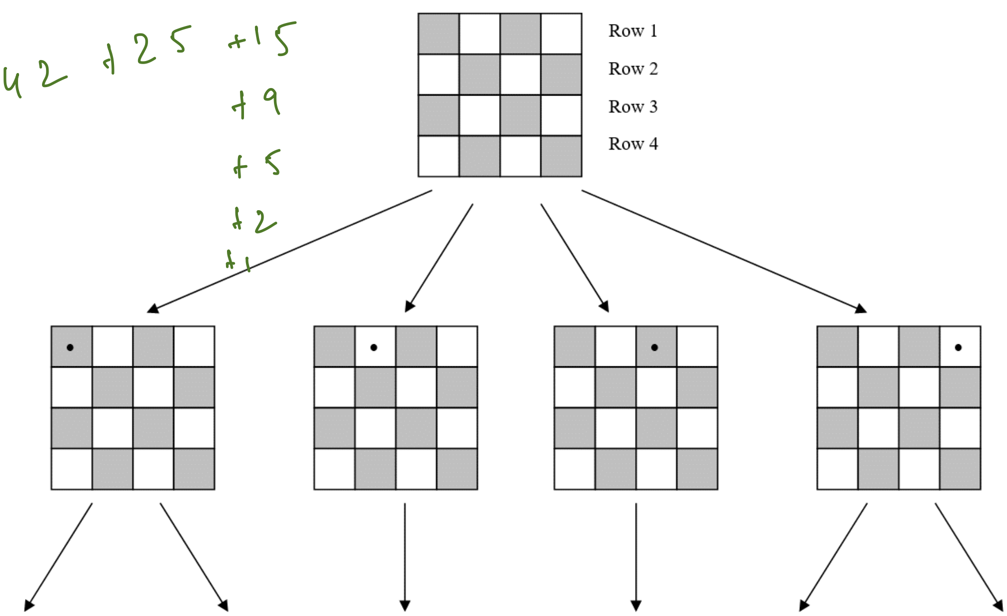
**A)** Assume that you can generate and check 1,000 arrangements per second. Approximately how long will it take to test all arrangements (give the answer in meaningful units, i.e. don't write "*Approximately 1805 seconds*", write "*Approximately 30 minutes*"). Clearly state any assumptions you are making. Assumption :- not in same row & col.
- can be same diagonal.

**B)** Assume you just want to find *any* answer (not all of them). I happen to know that there are 12 possible solutions, and assume that they are essentially randomly distributed in the search space . Approximately how long will we have to wait for a solution?

**C)** Consider the following problem. We want to put **9** queens on a *standard* chessboard such that none are attacking each other. We can quickly see that this impossible, but even if we could not, we could do exhaustive search, and as soon as the search ended without finding an answer, we could be convinced that no such solution existed.
Can you come up with a problem for which exhaustive search cannot show that a solution does not exist? (Your answer does not have to involve games, but it may) Hint, I can think of several answers to this question that could be written on a single line. If you are spending more than 15 minutes on this question, come see me for hints (but do not email me for hints on *just* this question).

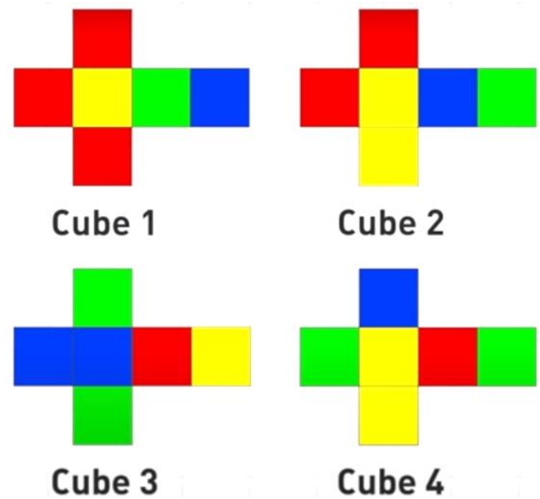
**D)** The more general version of the 8-queens problem is called the *n*-queens problem, where we try to find an arrangement of *n*-queens on an *n* by *n* board such that none are attacking each other. Draw a complete search tree for *n* = 4 and find all the solutions (if any). Draw only the legal states to save space (although you should realize that any algorithm (including your mind!) has to actually generate the illegal states in order to prune them).
To reduce the size of the tree we can use a little common sense. Once we place a queen on a row, we can never place another queen on the same row. So we can systemically start with an empty board, and expand to every possible placement of a queen on row one, then expand those nodes to every legal possible placement of a queen on row two etc.
I have done a little to get you started…

64 + 42 + 25 + 15 + 9 + 5 + 2 + 1



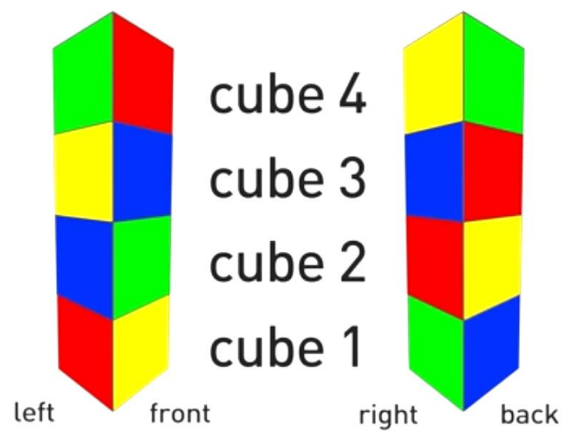
Consider this puzzle:

I give you four cubes; they are colored as shown in this image.




Note that the coloring on these cubes is very important. In other words: we’re only considering THESE four particular cubes. So, for example, Cube 1 has exactly three red faces, and one yellow, green, and blue face, moreover, those colors appear on the particular faces as indicated. Similarly for the other cubes.

Here is the puzzle: Can you stack these cubes so that each color appears exactly once on each of the four sides of the stack?




As it happens, the answer is “yes”, in fact I show the solution to the right.

It is possible to solve these problems using some graph theory, but we can also solve it with a brute force search.

**Question:** Suppose you decide to solve this problem with a brute force search.

- How large is the search space? Show your work, I am expecting about 4 to 12 English sentences explaining how you got your answer.
- Pick one cube of the four: Can you change one side’s color such that the problem has no solution? I am expecting about 3 to 6 English sentences what your change is, and why it leads to having no solution (or why this cannot be done).

**Appendix A**

If you have taken cs170 with me, you have already done a similar homework.

So, I am going to give you a different assignment.

1) Write two new questions, that I could ask in this homework (that test the same concepts), and write a key. For example you might say…

The Seven Bridges of Königsberg is a historically notable problem in mathematics. Its negative resolution by Leonhard Euler in 1736. However, I want you to imagine Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.
Given those assumptions:

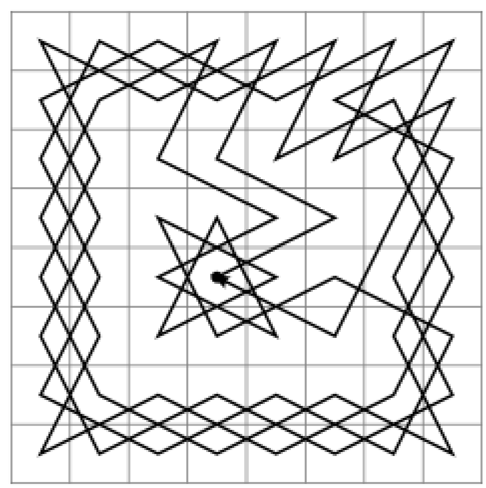
- Show that Ut enim ad minim veniam, quis nostrud.
- How many unique minim veniam, quis nostrud?

Or

A knight's tour is a sequence of moves of a knight on a chessboard such that the knight visits every square exactly once. If the knight ends on a square that is one knight's move from the beginning square (so that it could tour the board again immediately, following the same path), the tour is closed (or re-entrant); otherwise, it is open.

Suppose you have a 4 by 4 board, instead of a 8 by 8 board:

1) How many ut labore et dolore magna aliqua? (and upper and lower bound is sufficient).
2) If you allow mirror images to count, how et dolore magna aliqua?



The knight's tour. This particular solution is closed (circular), and can thus be completed from any point on the board

Or

<something interesting!>